

Exploring Problem Solving with C++ Across All Engineering Disciplines (In Spanish)

Abstract

Freshman engineering students globally often embark on their programs with varying degrees of computer programming experience, encountering a diverse array of languages such as Fortran, Visual Basic, C, JAVA, Python, and MATLAB in their initial year. Each language brings its own unique advantages and limitations. This paper investigates the introduction of a C++ course, emphasizing how it can elevate the critical thinking skills of engineering students across diverse educational and cultural settings. By drawing on case studies and examples from various international academic environments, this paper aims to showcase how C++ enhances problem-solving and adaptability to global engineering challenges. Unlike the higher-level nature of Python and MATLAB, C++ necessitates a more contemplative approach to code writing, cultivating a deeper understanding. The additional cognitive effort invested in learning C++ proves beneficial when students later engage with other programming languages. Furthermore, this paper illustrates how the object-oriented features of C++ can systematically address engineering problems in mechanical, civil, chemical, and electrical engineering disciplines on a global scale. Acknowledging the evolving educational landscape, we propose a specially tailored C++ course designed for engineering disciplines. This course is uniquely structured to incorporate global engineering scenarios, fostering intercultural learning and collaboration. Additionally, the curriculum bridges engineering and computer science, providing a valuable pathway for students contemplating a shift to computer science. The paper encompasses a detailed course description, sample problems, and comparative results, showcasing the course's ability to enhance critical thinking and address international educational goals.

Keywords: computer programming, C++, object-oriented, education, curriculum, first-year engineering, international engineering education

Explorando la resolución de problemas con C++ en todas las disciplinas de ingeniería

Resumen

Los estudiantes de primer año de ingeniería de todo el mundo suelen iniciar sus programas con distintos grados de experiencia en programación informática, y durante su primer año se enfrentan a una amplia gama de lenguajes como Fortran, Visual Basic, C, JAVA, Python y MATLAB. Cada lenguaje presenta sus propias ventajas y limitaciones. Este artículo investiga la introducción de un curso de C++, haciendo hincapié en cómo puede mejorar las habilidades de pensamiento crítico de los estudiantes de ingeniería en diversos entornos educativos y culturales. A partir de estudios de caso y ejemplos de diversos entornos académicos internacionales, este artículo pretende mostrar cómo C++ mejora la resolución de problemas y la adaptabilidad a los desafíos globales de la ingeniería. A diferencia de la naturaleza de alto nivel de Python y MATLAB, C++ requiere un enfoque más contemplativo para la escritura de código, lo que fomenta una comprensión más profunda. El esfuerzo cognitivo adicional invertido en el aprendizaje de C++ resulta beneficioso cuando los estudiantes se involucran posteriormente con otros lenguajes de programación. Además, este artículo ilustra cómo las características orientadas a objetos de C++ pueden abordar sistemáticamente problemas de ingeniería en las disciplinas de ingeniería mecánica, civil, química y eléctrica a escala global. Reconociendo la evolución del panorama educativo, proponemos un curso de C++ especialmente diseñado para las disciplinas de ingeniería. Este curso tiene una estructura única que incorpora escenarios globales de ingeniería, fomentando el aprendizaje intercultural y la colaboración. Además, el plan de estudios conecta la ingeniería con la informática, ofreciendo una valiosa vía de acceso para los estudiantes que contemplan una transición hacia la informática. El documento incluye una descripción detallada del curso, ejemplos de problemas y resultados comparativos, lo que demuestra la capacidad del curso para fomentar el pensamiento crítico y abordar objetivos educativos internacionales.

Palabras clave: programación informática, C++, orientado a objetos, educación, plan de estudios, ingeniería de primer año, formación internacional en ingeniería.

Introducción

MATLAB y Python se encuentran entre los lenguajes de programación más utilizados en la formación en ingeniería. Sin embargo, los estudiantes suelen interactuar con estos lenguajes de dos maneras distintas: algunos los utilizan, principalmente, para análisis numérico y álgebra lineal, mientras que otros los aprovechan como lenguajes de programación completos [1]. Los primeros tienden a escribir scripts para la resolución de problemas específicos, mientras que los estudiantes con experiencia previa en programación desarrollan funciones y soluciones computacionales más amplias.

Los estudiantes con dominio de la programación reconocen rápidamente las ventajas de la resolución estructurada de problemas, lo que les permite abordar desafíos complejos de ingeniería con mayor eficacia que quienes ven MATLAB únicamente como una herramienta matemática. La programación se basa en la organización lógica del pensamiento y la descripción determinista de procesos, habilidades que van más allá de los lenguajes específicos.

A medida que los estudiantes avanzan en sus carreras, las limitaciones de MATLAB, de manera especial, en términos de velocidad de ejecución y capacidades de E/S de datos, se hacen evidentes. Tareas como la construcción de imágenes de alta resolución a partir de datos sin procesar o la gestión de operaciones complejas

con archivos binarios a menudo requieren un lenguaje de bajo nivel y alto rendimiento como C++. Si bien C++ no siempre es un requisito previo, sirve como una caja de herramientas de ingeniería versátil y ofrece capacidades que van más allá de las aplicaciones especializadas de MATLAB.

La elección de C++ en este estudio se basa en sus ventajas únicas en aplicaciones de ingeniería. Si bien Python y MATLAB son esenciales para el análisis de datos y las simulaciones, C++ proporciona una comprensión más profunda de la gestión de memoria, la programación orientada a objetos y las operaciones de sistemas de bajo nivel, habilidades cruciales para las disciplinas de ingeniería. Además, muchas herramientas estándar de la industria, como el software de análisis de elementos finitos (FEA) y los sistemas embebidos, se crean con C++, lo que lo convierte en un lenguaje esencial para la resolución de problemas de ingeniería en aplicaciones del mundo real. Este curso está diseñado para dotar a los estudiantes de habilidades de pensamiento computacional directamente aplicables a los desafíos prácticos de la ingeniería.

Mejorando el Contexto Global en la Educación en Ingeniería

Para demostrar la adaptabilidad de este curso de C++ en diversos entornos educativos, examinamos su implementación en diversos contextos internacionales. En Alemania, C++ es un componente fundamental de los cursos de mecatrónica y sistemas de control, donde los estudiantes desarrollan software para automatización y robótica [2]. En Japón, C++ desempeña un papel importante en las aplicaciones de manufactura y automoción, y las universidades integran proyectos industriales en sus cursos [3]. En India, donde muchos estudiantes de ingeniería se especializan en desarrollo de software, C++ se enseña junto con estructuras de datos y algoritmos, lo que ayuda a los estudiantes a desarrollar sólidas habilidades de resolución de problemas computacionales

aplicables a simulaciones de ingeniería [4]. Estos ejemplos resaltan la continua relevancia de C++ en las disciplinas de ingeniería globales y cómo este curso puede adaptarse a las necesidades de la industria regional.

En un entorno de ingeniería globalizado, el pensamiento crítico y las estrategias adaptables de resolución de problemas son esenciales. La integración de C++ en los planes de estudio de ingeniería proporciona a los estudiantes herramientas de aplicación universal que les permiten afrontar retos del mundo real en diversos contextos culturales y geográficos. Esto se alinea con los objetivos educativos internacionales, formando a los estudiantes para colaboraciones en multinacionales de ingeniería. Al resolver problemas basados en C++ adaptados a escenarios globales, los estudiantes desarrollan la capacidad de pensar más allá de los contextos locales y considerar las implicaciones más amplias de sus soluciones.

Una diferencia fundamental entre C++ y lenguajes de alto nivel como MATLAB reside en el manejo de variables y la definición de tipos. En MATLAB, las operaciones de división consideran automáticamente los tipos de datos; por lo tanto, la operación x/y , donde $x=6$ y $y=4$, produce directamente 1,5. En cambio, C++ requiere una definición de tipo explícita o una conversión de tipos para obtener el mismo resultado. Si ambas variables se declaran como enteros, C++ realiza una división entera, devolviendo 1 en lugar de 1,5. Esta estricta aplicación de los tipos de datos mejora la comprensión de los estudiantes sobre la lógica de programación y la precisión computacional, habilidades valiosas al aprender otros lenguajes de programación.

Una encuesta de enero de 2023 que clasifica los lenguajes de programación por popularidad en las consultas de los motores de búsqueda ofrece más información sobre la relevancia de C++ [5,6]:

Lenguaje	Porcentaje
----------	------------

Python	16.36%
C	16.26%
C++	12.91%
Java	12.21%
C#	5.73%
Visual Basic	4.64%
JavaScript	2.87%
SQL	2.50%
Assembly language	1.60%
PHP	1.39%
Others	23.53%

Cabe destacar que el grupo C/C++/C# representa en conjunto el 35 % de todas las búsquedas relacionadas con la programación, lo que refuerza la continua demanda de C++ tanto en el ámbito académico como en el industrial.

Programación Orientada a Objetos para Desafíos Globales

Una de las características que definen a C++ es su paradigma de programación orientada a objetos (POO). Los programas informáticos generalmente siguen un enfoque orientado a procedimientos u objetos. En la programación orientada a procedimientos, las tareas se estructuran en funciones, con el foco puesto en los procesos y las operaciones [6]. En cambio, la programación orientada a objetos se centra en los objetos, que encapsulan tanto datos como comportamiento.

Un objeto puede representar cualquier cosa tangible (una entidad física o un modelo

conceptual) que pueda interactuar con otros objetos en un programa. Este enfoque mejora la modularidad y la reutilización, permitiendo a los programadores definir clases como planos a partir de los cuales se pueden crear múltiples objetos con mínimas modificaciones. Estas ventajas hacen que la POO se adopte ampliamente en aplicaciones de ingeniería, reduciendo el tiempo de desarrollo y mejorando la eficiencia del código.

Cada objeto tiene atributos (propiedades descriptivas) y comportamientos (acciones que puede realizar o a las que puede responder) [7]. Por ejemplo, considere una clase Rectángulo en una simulación de ingeniería:

- Una empresa de paisajismo podría usar el objeto para estimar los costos de instalación de césped. Una empresa de cercas podría usar el mismo objeto para calcular los gastos de cercado.

Al definir una sola clase, los programadores pueden aplicar el mismo objeto en múltiples escenarios de resolución de problemas, lo que refuerza la reutilización y la eficiencia del código.

La siguiente sección explora aplicaciones reales de ingeniería de C++ y POO, demostrando cómo el diseño basado en clases mejora las habilidades de resolución de problemas y el pensamiento computacional en las disciplinas de ingeniería.

Colaboración Internacional y Sensibilidad Cultural en la Formación en Ingeniería

La integración del aprendizaje y la colaboración interculturales en los planes de estudio de ingeniería mejora tanto la competencia técnica como la adaptabilidad cultural, preparando a los estudiantes para un mercado laboral globalizado. Estos enfoques son especialmente relevantes en la enseñanza de C++, un lenguaje que requiere un sólido dominio de los principios fundamentales de la programación y la resolución estructurada de problemas.

Una distinción clave de C++ reside en su énfasis en los tipos de datos y la gestión de memoria,

habilidades fundamentales esenciales para las aplicaciones de ingeniería del mundo real. Dominar estos conceptos permite a los estudiantes abordar desafíos computacionales complejos en diversas disciplinas de la ingeniería.

La siguiente sección explora escenarios prácticos de ingeniería donde C++ y la programación orientada a objetos desempeñan un papel crucial en la resolución eficiente de problemas.

1. Descripción del curso

Este estudio se llevó a cabo durante dos semestres con 27 estudiantes de primer año de ingeniería. Como asignatura obligatoria en el currículo de ingeniería de primer año, participaron todos los estudiantes matriculados, representando una mezcla diversa de disciplinas, incluyendo ingeniería mecánica, civil, eléctrica y química. Esta diversidad garantizó que el currículo rediseñado se evaluara en diversos contextos de resolución de problemas de ingeniería.

Cada clase tenía capacidad para entre 15 y 20 estudiantes en un laboratorio de informática equipado con Visual Studio. El programa de estudios abarcaba conceptos fundamentales de programación, como variables, flujo de control, bucles, procesamiento de cadenas, matrices, programación orientada a objetos (POO) y gestión de archivos.

Al inicio del curso, se introdujo a los estudiantes al proceso de resolución de problemas de ingeniería (Figura 1) [8], que enfatiza la identificación, el análisis, el desarrollo y la presentación de soluciones, una metodología que se refuerza a lo largo del currículo de ingeniería.

Steps in the Engineering Problem Solving Method	
1.	Reconocimiento y comprensión del problema.
2.	Accumulate data and verify accuracy.
3.	Select the appropriate theory or principle.
4.	Make necessary assumptions.
5.	Solve the problem.

Figura 1. Proceso de resolución de problemas de ingeniería.

En los dos últimos semestres, el profesor utilizó zyBooks, un libro de texto interactivo diseñado para estudiantes de informática. Sin embargo, sus ejercicios genéricos —que abarcaban escenarios financieros, patrones numéricos, conteo de dígitos, formato de fechas y calificaciones— carecían de un enfoque específico de ingeniería [9].

En el primer semestre, el curso se basó únicamente en los ejemplos y ejercicios del libro de texto. En el segundo semestre, el currículo se rediseñó para enfatizar la resolución de problemas de ingeniería y el análisis de sistemas. Las actividades revisadas incorporaron desafíos específicos de ingeniería de diversos campos, como la economía de la ingeniería, la termodinámica, la estática, el balance de materiales y energía, y el análisis de circuitos. Estos cambios buscaban fortalecer el pensamiento crítico y las habilidades analíticas de los estudiantes en contextos reales de ingeniería.

Los criterios de calificación del curso son los siguientes:

Actividad de Participación (PA)	15%
Actividades de Laboratorio	20%
Actividad de Reto (CA)	15%
Proyectos (tres) - ponderación igual	50%
Total	100%

- **Actividades de Participación (AP):** Ejercicios interactivos con animación que se realizan antes de clase para desarrollar conocimientos fundamentales.
- **Actividades de Laboratorio:** Se realizan en parejas y se centran en la resolución colaborativa de problemas en aplicaciones de ingeniería del mundo real.

- Actividades de Desafío (AC): Versiones más avanzadas de las AP, que requieren mayor exploración y complejidad.
- Proyectos: En sustitución de los exámenes tradicionales, los estudiantes completan tres proyectos principales aplicando C++ a problemas de ingeniería interdisciplinarios. Este enfoque refuerza las aplicaciones prácticas y el aprendizaje interdisciplinario.

2. Metodología

El estudio evalúa la eficacia de integrar C++ en un currículo de ingeniería, comparando el rendimiento de los estudiantes en un curso de programación tradicional con un curso rediseñado con un enfoque en C++ centrado en la ingeniería. El estudio se llevó a cabo durante dos semestres, con la participación de todos los estudiantes matriculados en el curso.

2.1 Muestreo y asignación de grupos

Los estudiantes no fueron asignados aleatoriamente al curso tradicional o al rediseñado, ya que cada semestre solo se ofreció una sección. Sin embargo, ambas cohortes tenían características demográficas similares, incluyendo formación académica y experiencia previa en programación. El instructor del curso se mantuvo para minimizar la variabilidad en la enseñanza. Estudios futuros buscarán implementar un diseño aleatorio o cuasiexperimental para mejorar la robustez de los datos.

2.2 Recopilación y evaluación de datos

El rendimiento de los estudiantes se evaluó mediante una combinación de:

- Actividades de participación (PA): ejercicios interactivos realizados antes de la clase.
- Actividades de laboratorio: problemas de ingeniería reales resueltos de forma colaborativa.

- Actividades de desafío (CA): versiones avanzadas de las PA que requerían habilidades más profundas de resolución de problemas.
- Evaluación basada en proyectos: tres proyectos de programación interdisciplinarios.

En lugar de basarnos únicamente en las calificaciones finales, examinamos las evaluaciones basadas en proyectos como un indicador del desarrollo del pensamiento crítico. Si bien no se realizaron evaluaciones directas del pensamiento crítico, las reflexiones de los estudiantes y las observaciones del profesorado indicaron una mayor participación en la resolución de problemas. Las futuras iteraciones incorporarán herramientas de evaluación validadas, como pruebas previas y posteriores, o rúbricas estructuradas para la evaluación del pensamiento crítico.

3. Ejercicios de ejemplo del curso

Se imparten varios ejercicios en clase para demostrar y destacar la importancia de la resolución de problemas de ingeniería integrada con las funciones orientadas a objetos de C++. Algunos de ellos son:

1. Determinación del momento de inercia para áreas simples y compuestas
2. Localización del centroide de cuerpos simples y compuestos
3. Formulación vectorial cartesiana del momento de una fuerza respecto a un punto y respecto a un eje
4. Realización de operaciones con números complejos
5. Determinación del calor específico de gases ideales
6. Cálculo de la humedad relativa

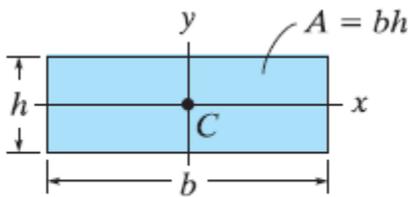
3.1. Momento de inercia de áreas simples y compuestas - Estática

El momento de inercia es una propiedad geométrica crucial que se utiliza para evaluar la resistencia de un elemento estructural o determinar la ubicación de una fuerza de presión resultante en placas sumergidas. Se define como el segundo momento de un área respecto a un eje, calculado elevando al cuadrado la distancia desde el eje hasta cada elemento del área. Los valores estándar de momento de inercia para formas básicas (triángulos, rectángulos, círculos, etc.) se suelen introducir en los cursos de Estática.

Para automatizar estos cálculos, los estudiantes pueden desarrollar una clase *MomentOfInertia* en C++, que define atributos y métodos clave. La clase realiza tres funciones principales:

1. Inicializar miembros de datos privados (p. ej., longitud y anchura).
2. Asignar valores a partir de la entrada del usuario o de un programa.
3. Calcular y devolver momentos de inercia a lo largo de los ejes x e y mediante fórmulas estándar.

$$I_x = \frac{1}{12} b \times h^3, \quad I_y = \frac{1}{12} h \times b^3$$



Rectangular area

Al ejecutar estas tareas, el objeto *MomentOfInertia* se convierte en una herramienta versátil para calcular momentos de inercia de diversas formas, lo que contribuye a la eficiencia y la modularidad del programa.

A continuación, se describe la implementación parcial de la clase *MomentOfInertia*. Esta clase encapsula los atributos y métodos necesarios para

facilitar el cálculo de momentos de inercia de área para diferentes formas. Esta clase parcial incluye métodos para definir dimensiones y calcular momentos de inercia para los ejes x e y. Puede integrar esta clase en un programa inicializando una instancia de *MomentOfInertia*, definiendo las dimensiones y utilizando los valores calculados según sea necesario.

```
#include <cmath>

//declaration section
class MomentOfInertia
{
public:
    MomentOfInertia();
    void setDimensions(double, double);
    double calcIx();
    double calcIy();

private:
    double length;
    double width;
};

//Implementation

MomentOfInertia::MomentOfInertia() :
length (0.0), width(0.0) {}
double MomentOfInertia::calcIx() { return
(1.0 / 12) * length * pow(width,3); }

double MomentOfInertia::calcIy() { return
(1.0 / 12) * width * pow(length, 3); }
```

Este enfoque orientado a objetos mejora la modularidad y la eficiencia, permitiendo a los estudiantes ampliar la clase para calcular momentos de inercia de áreas compuestas. Además, la clase puede ampliarse para determinar momentos de inercia respecto a ejes inclinados (Figura 2) mediante transformaciones trigonométricas:

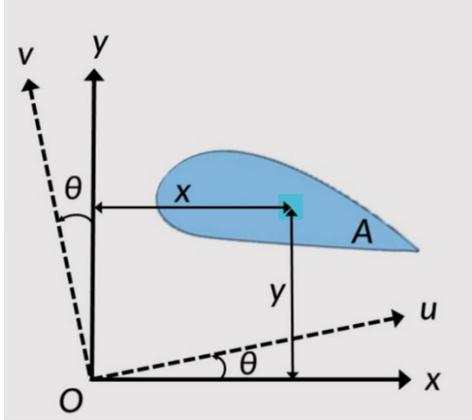


Figura 2. Momento de inercia para un área alrededor de ejes inclinados

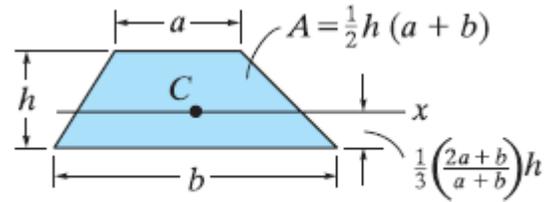
$$I_u = \frac{I_x + I_y}{2} + \frac{I_x - I_y}{2} \cos 2\theta - I_{xy} \sin 2\theta \quad \text{Eq. 1}$$

$$I_v = \frac{I_x + I_y}{2} + \frac{I_x - I_y}{2} \cos 2\theta + I_{xy} \sin 2\theta \quad \text{Eq. 2}$$

3.2. Centroide de cuerpos simples y compuestos - Estática

El centroide sirve como centro geométrico de un cuerpo, coincidiendo con el centro de masa o gravedad solo cuando el material que lo compone es uniforme u homogéneo [10]. Las fórmulas para calcular el centroide de áreas regulares (como segmentos de arco circular, cuartos, arcos de semicírculo, trapecios, semiparábolas y áreas parabólicas) se encuentran al final de los libros de texto de Estática.

Para simplificar estos cálculos, se puede crear una clase Centroide que solicite (a) la forma del área y (b) la(s) dimensión(es) requerida(s) para calcular el centroide. Por ejemplo, para un área trapezoidal, el centroide se puede determinar proporcionando las magnitudes de las dos bases y la altitud.



En ingeniería mecánica, existen casos en los que es necesario determinar el centroide de un cuerpo compuesto, compuesto por una serie de cuerpos con formas más simples y conectados. Tras dividir el cuerpo en partes compuestas más simples, el centroide del cuerpo compuesto puede calcularse mediante las siguientes fórmulas:

$$\bar{x} = \frac{\sum \tilde{x}W}{\sum W}, \quad \bar{y} = \frac{\sum \tilde{y}W}{\sum W}, \quad \bar{z} = \frac{\sum \tilde{z}W}{\sum W} \quad \text{Eq. 3}$$

Aquí \tilde{x} , \tilde{y} y \tilde{z} representan las coordenadas del centroide de cada parte compuesta del cuerpo, que se pueden calcular utilizando la clase Centroid.

3.3. Momento de una fuerza respecto a un punto y un eje: Formulación vectorial - Estática

El momento de una fuerza F respecto a un punto O (Figura 3) se determina mediante el producto vectorial:

$$\mathbf{M}_O = \mathbf{r} \times \mathbf{F} \quad \text{Eq. 4}$$

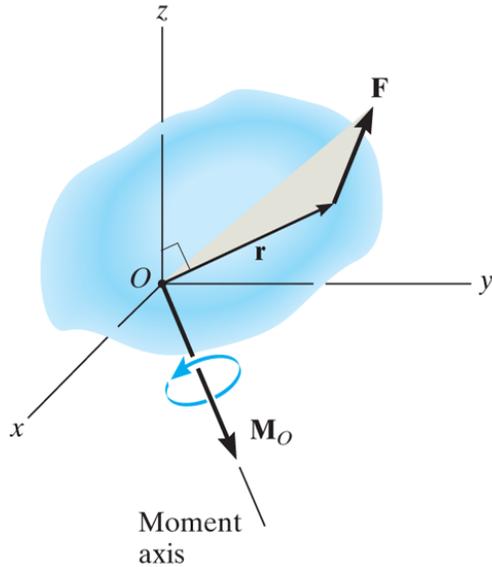


Figura 3. Momento de una fuerza respecto a un punto-Formulación vectorial

Aquí \mathbf{r} es el vector de posición desde O hasta cualquier punto de la línea de acción de \mathbf{F} [10]. En forma cartesiana:

$$\mathbf{M}_O = \mathbf{r} \times \mathbf{F} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ r_x & r_y & r_z \\ F_x & F_y & F_z \end{vmatrix} \quad \text{Eq.5}$$

Los componentes resultantes (M_{Ox}, M_{Oy}, M_{Oz}) definen el momento en el espacio 3D y su magnitud está dada por:

$$|\mathbf{M}_O| = \sqrt{M_{O,x}^2 + M_{O,y}^2 + M_{O,z}^2} \quad \text{Eq. 6}$$

Los ángulos de dirección relativos a los ejes de coordenadas son:

$$\begin{aligned} \alpha &= \cos^{-1} \left(\frac{M_{O,x}}{|\mathbf{M}_O|} \right), & \text{Eq. 7} \\ \beta &= \cos^{-1} \left(\frac{M_{O,y}}{|\mathbf{M}_O|} \right), & \gamma \\ &= \cos^{-1} \left(\frac{M_{O,z}}{|\mathbf{M}_O|} \right) \end{aligned}$$

Para momentos alrededor de un eje específico a (Figura 4), la forma determinante es:

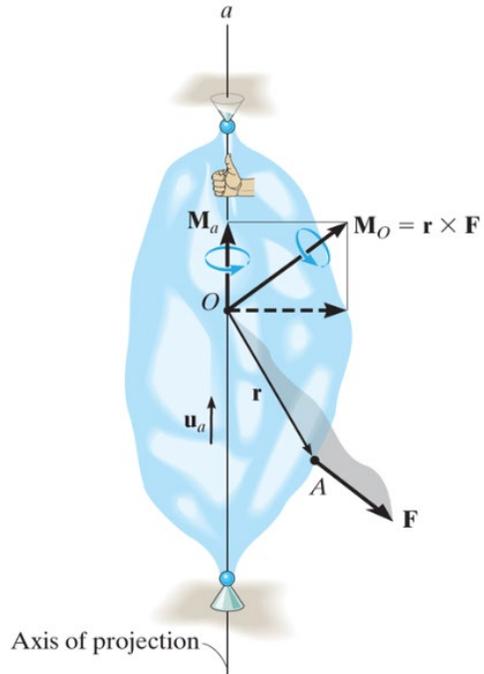


Figura 4. Momento de una fuerza respecto a un eje

$$\begin{aligned} \mathbf{M}_a &= \mathbf{u}_a \cdot (\mathbf{r} \times \mathbf{F}) & \text{Eq. 8} \\ &= \begin{vmatrix} u_{ax} & u_{ay} & u_{az} \\ r_x & r_y & r_z \\ F_x & F_y & F_z \end{vmatrix} \end{aligned}$$

Donde \mathbf{u}_a es el vector unitario a lo largo del eje a . Estos cálculos se pueden implementar en una clase *MomentVector* para determinar momentos respecto a un punto o un eje.

3.4. Análisis de circuitos de CA con números complejos

En el campo de la Ingeniería Eléctrica, los números complejos desempeñan un papel crucial en el curso de Análisis de Circuitos de CA. Los estudiantes los utilizan ampliamente para representar fasores, analizar la impedancia y realizar cálculos relacionados con circuitos de corriente alterna. La representación fasorial simplifica el análisis de señales sinusoidales, lo que la convierte en una herramienta poderosa para comprender las relaciones entre voltaje y corriente, así como para los cálculos de potencia en circuitos sujetos a frecuencias variables.

Un número complejo en matemáticas se define como $a+bi$, donde a define la parte real del número y b es la parte imaginaria [11]. La letra i representa la raíz cuadrada de -1 (lo que significa que i^2 es -1). Para realizar operaciones básicas con números complejos, se puede crear una clase "Complejo".

1. Addition:

$$\begin{aligned} (a_1 + b_1i) + (a_2 + b_2i) & \text{ Eq. 9} \\ &= (a_1 + a_2) + i(b_1 + b_2) \end{aligned}$$

2. Subtraction:

$$\begin{aligned} (a_1 + b_1i) - (a_2 + b_2i) & \text{ Eq. 10} \\ &= (a_1 - a_2) + i(b_1 - b_2) \end{aligned}$$

3. Multiplication:

$$\begin{aligned} (a_1 + b_1i) * (a_2 + b_2i) & \text{ Eq. 11} \\ &= (a_1a_2 - b_1b_2) + i(a_1b_2 + a_2b_1) \end{aligned}$$

4. Division:

$$\begin{aligned} \frac{(a_1 + b_1i)/(a_2 + b_2i)}{(a_1a_2 + b_1b_2) + i(-a_1b_2 + a_2b_1)} & \text{ Eq. 12} \\ &= \frac{a_2^2 + b_2^2}{a_2^2 + b_2^2} \end{aligned}$$

The *Complex* class will prompt for:

- The operation to be performed.

- The real and imaginary parts of the host object.
- The real and imaginary parts of the new complex number.

La clase *Complejo* solicitará:

- La operación a realizar.
- Las partes real e imaginaria del objeto anfitrión.
- Las partes real e imaginaria del nuevo número complejo.

3.5. Calores específicos de gases ideales - Termodinámica

El cambio en la energía interna o entalpía de un gas ideal durante un proceso del estado 1 al estado 2 se determina mediante las siguientes integraciones:

$$\Delta u = u_2 - u_1 = \int_1^2 c_v(T) dT \left(\frac{kJ}{kg} \right) \text{ Eq. 13}$$

$$\Delta h = h_2 - h_1 = \int_1^2 c_p(T) dT \left(\frac{kJ}{kg} \right) \text{ Eq. 14}$$

Para realizar estas integraciones, se requieren relaciones para c_v y c_p en función de la temperatura [12]. A bajas presiones, todos los gases reales se comportan de forma similar a la de un gas ideal, y sus calores específicos dependen únicamente de la temperatura. Los calores específicos de los gases reales a bajas presiones se conocen como calores específicos de gas ideal o calores a presión cero, a menudo denotados como c_{p_0} y c_{v_0} . Existen expresiones analíticas precisas para los calores específicos de un gas ideal, basadas en mediciones directas o cálculos a partir del comportamiento estadístico de las moléculas, como polinomios de tercer grado:

$$\bar{c}_p = a + bT + cT^2 + dT^3 \text{ Eq. 15}$$

where T is in Kelvin, c_p in kJ/kmol.K, and values for a , b , and c for each substance are tabulated.

1.1.1. Specific Heat Class:

One can create a *SpecificHeat* class that will:

1. Prompt for the substance and the temperature range for specific heat calculation.
2. Check if the temperature range is within the valid range from the specific heat tables; otherwise, display an error.
3. Set up the polynomial with obtained constants and perform integration to obtain a fourth-order polynomial as the antiderivative.
4. Apply the Fundamental Theorem of Calculus to calculate the change in enthalpy of the substance for the given temperature range.
5. Calculate \bar{c}_v from \bar{c}_p using $\bar{c}_v = \bar{c}_p - R$ ($\frac{kJ}{kg.K}$) (where R is the universal gas constant).
6. Allow the user to specify whether to find the change in internal energy or enthalpy, and whether these changes are to be found on a mass or molar basis.

This class provides a comprehensive tool for thermodynamic calculations, automating the process and offering flexibility in choosing the basis for energy changes.

1.2. Relative Humidity-Material and Energy Balance

Relative humidity (RH) is a vital consideration in the Materials and Energy Balance course, particularly in processes involving drying, evaporation, and reactions influenced by water content. A profound understanding and accurate accounting for relative humidity are crucial in material balance calculations, where moisture content significantly affects mass transfer and energy considerations in various engineering

applications. Furthermore, in contexts such as heat exchangers and environmental control systems, engineers must factor in the impact of relative humidity on heat transfer properties and overall energy balances within a system.

1.2.1. Calculation of Relative Humidity

The relative humidity (RH) at sea level can be calculated using the dry-bulb temperature (T_{db}) and the wet-bulb temperature (T_{wb}), with temperatures in degrees Celsius [13].

The formula is given by:

$$RH = \frac{VP}{SVP} \times 100 \quad \text{Eq. 16}$$

where VP is the vapor pressure and SVP is the saturated vapor pressure. Vapor pressure (VP) is determined by the formula:

$$VP = e^{\frac{16.78T_{wb}-116.9}{T_{wb}+237.3}} - 0.066858(1 + 0.00115T_{wb})(T_{db} - T_{wb}) \quad \text{Eq. 17}$$

Y la presión de vapor saturado (SVP) se calcula utilizando:

$$SVP = e^{\frac{16.78T_{db}-116.9}{T_{db}+237.3}} \quad \text{Eq. 18}$$

Para facilitar la experiencia del usuario, se puede diseñar una clase de Humedad Relativa para determinar la HR al proporcionar las temperaturas de bulbo seco y bulbo húmedo. La clase incluye una sección de conversión de unidades de temperatura de Celsius a Fahrenheit. Esto no solo ahorra tiempo al usuario, sino que también permite un análisis de sensibilidad rápido para comprender cómo los cambios en T_{db} o T_{wb} afectan la humedad relativa.

4. Resultados y Discusión

4.1. Evaluación de Impacto

Los datos para este estudio se recopilaron de proyectos estudiantiles asignados a lo largo del

semestre. Cada proyecto se diseñó para evaluar la capacidad de los estudiantes para aplicar conceptos de programación en C++ y resolver problemas relacionados con la ingeniería. La rúbrica de calificación enfatizó la descomposición lógica de problemas, la eficiencia del código, la capacidad de depuración y la correcta implementación de los principios orientados a objetos. Para garantizar la coherencia, todos los proyectos se evaluaron utilizando una rúbrica estandarizada y los estudiantes recibieron retroalimentación detallada sobre su enfoque y estrategias de resolución de problemas.

Los estudiantes tuvieron que traducir desafíos reales de ingeniería en modelos computacionales utilizando C++. Por ejemplo, implementaron algoritmos para calcular el centroide de cuerpos compuestos, analizaron circuitos de CA utilizando números complejos y modelaron procesos termodinámicos. Estas tareas los desafiaron a descomponer problemas complejos, aplicar principios orientados a objetos y refinar sus soluciones mediante la depuración iterativa. La profundidad de sus soluciones y su capacidad para resolver errores fueron indicadores clave del desarrollo de su pensamiento crítico. El estudio se llevó a cabo durante dos semestres con un total de 27 estudiantes matriculados en el curso de programación en C++, que representan diversas disciplinas de ingeniería, como la mecánica, la civil y la eléctrica. Si bien el tamaño de la muestra es limitado, el enfoque estructurado basado en proyectos proporciona una evaluación significativa del desarrollo de habilidades de programación. Estudios futuros ampliarán el conjunto de datos para incluir iteraciones adicionales del curso e instituciones, con el fin de mejorar la generalización de los hallazgos.

El impacto del nuevo estilo de enseñanza se evaluó mediante dos medidas principales. En primer lugar, se realizó un análisis comparativo de las calificaciones promedio generales entre el primer y el segundo semestre mediante una prueba t para dos muestras con varianzas desiguales (véase la Tabla 1).

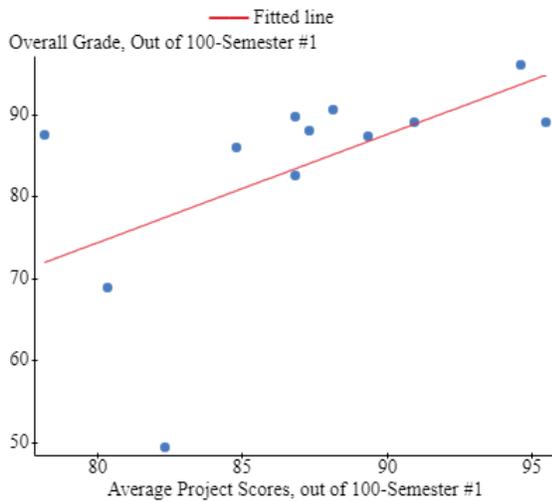
Tabla 1. Comparación por semestre

	Semestre #1	Semestre #2
Número de estudiantes	12	15
Calificación general promedio	83.8	87.4
Desviación estándar	12.6	8.9
Valor P (de dos colas)	0.414	

El análisis estadístico arrojó un valor p de 0,414, lo que indica que no hay diferencias estadísticamente significativas entre los cursos tradicionales y los rediseñados. Si bien el curso rediseñado mostró un aumento de 3,6 puntos en las calificaciones promedio, esta diferencia podría atribuirse a la variabilidad más que al fuerte efecto de la nueva metodología de enseñanza. Además, el pequeño tamaño de la muestra (n=27) limita la potencia estadística de esta comparación, lo que dificulta la detección de diferencias pequeñas pero significativas. Cabe destacar que el currículo rediseñado también buscaba mejorar las habilidades de resolución de problemas de los estudiantes, como lo demuestran las mejoras cualitativas en la complejidad del proyecto, las habilidades de depuración y la eficiencia algorítmica. Las investigaciones futuras incorporarán tamaños de muestra mayores y métricas de rendimiento adicionales para evaluar con mayor precisión el impacto del currículo.

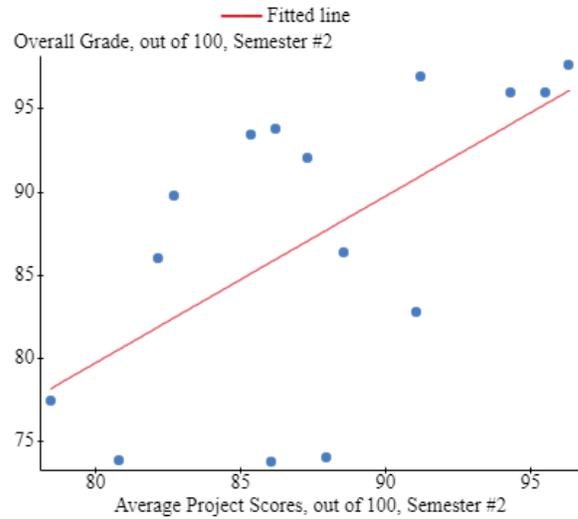
En segundo lugar, la evaluación examinó si el nuevo enfoque proporciona una medida significativa de las habilidades de resolución de problemas de los estudiantes (un aspecto no estudiado previamente) al comparar las calificaciones promedio del proyecto con las calificaciones del examen final. Como se ilustra en las Figuras 5 y 6, existe una correlación positiva entre las calificaciones del proyecto y las calificaciones finales; los estudiantes con calificaciones más altas en el proyecto tienden a obtener calificaciones finales más altas. Aunque los coeficientes de correlación de 0,55 y 0,60 indican una relación relativamente débil, afirman el valor del nuevo enfoque de enseñanza como un

posible indicador principal del rendimiento general de los estudiantes.



Sample size: 12
 R (correlation coefficient) = 0.55003957
 R-sq = 0.30254353
 Estimate of error standard deviation: 11.009782

Figura 5. Correlación de la puntuación media del proyecto con la calificación general del semestre 1



Sample size: 15
 R (correlation coefficient) = 0.60232108
 R-sq = 0.36279068
 Estimate of error standard deviation: 7.3861918

Figura 6. Correlaciones entre la puntuación promedio del proyecto y la calificación general del semestre 2

Este análisis de correlación proporciona información valiosa sobre la eficacia del nuevo enfoque de enseñanza para evaluar y mejorar las capacidades de resolución de problemas de los estudiantes, destacando su potencial valor predictivo para el desempeño académico general.

4.2. Trabajos futuros

Como parte de nuestro compromiso continuo con la mejora de la experiencia de aprendizaje, planeamos ampliar y profundizar nuestro análisis. Para lograrlo, los estudios futuros emplearán muestras más grandes y herramientas de evaluación más robustas. También exploraremos estudios longitudinales que monitoreen la competencia en programación de los estudiantes a lo largo de varios semestres para evaluar los beneficios a largo plazo. Además, se realizarán evaluaciones de codificación antes y después del curso y entrevistas estructuradas a los estudiantes para medir directamente las mejoras en el pensamiento crítico.

Se planean las siguientes iniciativas:

1. Análisis de cohorte extendido: Tenemos la intención de replicar nuestro análisis con cohortes sucesivas, ampliando nuestro conjunto de datos. Este enfoque proporcionará una base más sólida para nuestras conclusiones, permitiéndonos discernir tendencias y patrones a lo largo del tiempo.

2. Encuestas longitudinales: Para comprender el impacto duradero del nuevo enfoque de enseñanza, realizaremos encuestas de seguimiento anuales. Estas encuestas examinarán las tasas de retención, el rendimiento académico sostenido, la participación en actividades relacionadas con la ingeniería y la eventual inserción laboral. Esta perspectiva longitudinal ofrecerá una visión integral de los efectos duraderos de nuestras metodologías educativas. Estas iniciativas son parte integral de nuestro compromiso con la mejora continua y para garantizar la eficacia sostenida de nuestras estrategias de enseñanza.

4.3 *Discusión y Resultados del Aprendizaje*

El curso de C++ rediseñado tuvo como objetivo mejorar la capacidad de resolución de problemas de los estudiantes de ingeniería mediante la integración del pensamiento computacional con aplicaciones del mundo real. Si bien los resultados mostraron una modesta mejora en las calificaciones promedio de los estudiantes (+3.6 puntos), el análisis estadístico no indicó una diferencia significativa. Sin embargo, los indicadores cualitativos sugirieron mejoras significativas en la participación estudiantil y el pensamiento analítico.

Rendimiento Estudiantil y Aprendizaje Basado en Proyectos

El análisis de correlación entre las calificaciones del proyecto y las calificaciones finales ($r = 0.55$, 0.60 para dos semestres) sugiere que un buen desempeño en proyectos de programación basados en ingeniería es un predictor positivo del éxito general del curso. Los estudiantes que se

involucraron más a fondo en actividades de resolución de problemas tendieron a tener un mejor desempeño, lo que refuerza el valor de las evaluaciones basadas en proyectos.

Comentarios y Observaciones Cualitativas

Los estudiantes reportaron mayor confianza en la depuración y la estructuración lógica de su código.

Se observó una mayor participación en aplicaciones interdisciplinarias, particularmente en contextos de ingeniería mecánica y eléctrica.

Las observaciones del instructor destacaron una mejor retención de los conceptos de programación en comparación con versiones anteriores del curso. Si bien la falta de una evaluación formal del pensamiento crítico es una limitación, los estudios futuros integrarán encuestas cualitativas y evaluaciones basadas en rúbricas para medir el desarrollo de habilidades cognitivas.

En conclusión, este artículo presenta una exploración exhaustiva de un enfoque docente renovado en la formación en ingeniería, que integra a la perfección ejercicios prácticos de resolución de problemas con conceptos de programación orientada a objetos en C++. La incorporación de aplicaciones prácticas, como el cálculo del momento de inercia del área, la localización del centroide, la forma vectorial del momento, las operaciones con números complejos y el cálculo del calor específico de gases ideales y la humedad relativa, ha demostrado eficazmente la versatilidad y la practicidad de la pedagogía propuesta para conectar el conocimiento teórico con escenarios reales de ingeniería.

La evaluación del impacto de este innovador estilo de enseñanza ha revelado perspectivas interesantes. Si bien los análisis estadísticos mostraron una calificación general promedio ligeramente superior en el curso rediseñado en comparación con la sección tradicional, la diferencia, aunque no estadísticamente significativa, sugiere la promesa del nuevo enfoque. Además, el análisis de las correlaciones

entre las puntuaciones de los proyectos y las calificaciones de los exámenes finales indicó una asociación lógica, lo que destaca el nuevo enfoque docente como un valioso predictor de la capacidad de resolución de problemas de los estudiantes a escala global.

De cara al futuro, las iniciativas futuras ampliarán este análisis para abarcar cohortes internacionales más amplias, lo que garantizará una comprensión más completa del impacto sostenido de estas innovadoras metodologías de enseñanza. Las encuestas longitudinales proporcionarán información valiosa sobre la trayectoria académica de los estudiantes, su participación en actividades globales de ingeniería y su inserción profesional, lo que contribuirá al perfeccionamiento y la optimización continuos de nuestras estrategias educativas. Este compromiso con una perspectiva global subraya nuestra dedicación a la formación de ingenieros integrales capaces de abordar desafíos a escala mundial.

Referencias

[1] “Python for Engineers”: The First Course of Computing for a General Engineering Curriculum, Edris Ebrahimzadeh, Nick Safai, ASEE, 2019

[2]https://www.mec.ed.tum.de/en/ais/study-and-teaching/fundamental-and-specialized-courses/industrial-software-development-of-mechatronic-systems-and-implementation-in-c/?utm_source=chatgpt.com

[3]https://japan-dev.com/jobs/mujin/mujin-software-engineering-for-robotics-c-python-lktsvj?utm_source=chatgpt.com

[4]
https://www.cse.iitm.ac.in/course_details.php?arg=ODk%3D&utm_source=chatgpt.com

[5] <https://www.tiobe.com/tiobe-index/2023>

[6] Introduction to C++, zyBooks, 2023

[7] An Introduction to Programming with C++, Diane Zak, Cengage Learning, 8th Edition, 2016

[8] Engineering Fundamentals and Problem Solving, Arvid Eide, Steven Mickelson, Cheryl Eide, Roland Jenison, Larry Northup, 8th Edition, McGraw Hill, 2023.

[9] Animations for Learning: Design Philosophy and Student Usage in Interactive Textbooks, Nikitha Sambamurthy, Alex Daniel Edgcomb, Frank Vahid, ASEE, 2019.

[10] Engineering Mechanics, R.C. Hibbeler, 5th Edition, Pearson, 2022

[11] C Programming: An Object-Oriented Approach, Behrouz A. Forouzan, Richard F. Gilberg, McGraw Hill, 2020

[12] Thermodynamics: An Engineering Approach, Yunus Cengel, Michael Boles, Mehmet Kanoglu, 10th Edition, McGraw Hill, 2024.

[13] MATLAB: An Introduction with Applications, Amos Gilat, 6th Edition, John Wiley & Sons, Inc., 2017.