

# Simulación y Análisis de Operaciones Vectoriales en 2D mediante Python\*

Alan Miranda, Marlon Caviedes, Cristian Orozco

## Abstract

En este trabajo se presenta una implementación de un entorno de simulación 2D utilizando Python, en el que se realizan diversas operaciones vectoriales sobre un punto definido sobre una circunferencia. Las operaciones implementadas incluyen desplazamiento, simulación del movimiento, escalado, producto escalar, cálculo de la norma y rotación. Todas las operaciones se efectúan de manera vectorizada usando NumPy, lo que garantiza eficiencia y claridad en el desarrollo. Los resultados se muestran tanto de forma gráfica como en texto, permitiendo un análisis completo de los métodos aplicados. **Palabras clave:** operaciones vectoriales, simulación en Python, NumPy, transformación 2D, animación.

## 1 Introducción

Las operaciones vectoriales son fundamentales en áreas como la robótica, gráficos por computadora y análisis de datos. En este estudio se implementa un entorno 2D en el que un punto, definido sobre una circunferencia, se somete a varias transformaciones. El objetivo es analizar y demostrar operaciones como el desplazamiento, simulación del movimiento, escalado, producto escalar, cálculo de la norma y rotación, utilizando Python y aprovechando operaciones vectorizadas.

## 2 Metodología

La implementación se ha desarrollado en Python mediante la clase `Escena2D`. Las funcionalidades principales de la clase son:

- **Creación de la escena:** Se genera una circunferencia con un radio y color especificados, y se inicializa un punto en una posición determinada.
- **Desplazamiento y simulación:** El punto se mueve en las direcciones indicadas (derecha, izquierda, arriba, abajo) mediante operaciones vectorizadas, y se produce una animación del movimiento usando `FuncAnimation` de Matplotlib.

---

\*Esta investigación fue apoyada por la subvención No. 1234.

- **Escalado:** Se multiplica la posición del punto por un factor de escalado.
- **Producto escalar y norma:** Se calcula el producto escalar entre el punto y un vector de prueba, y se determina la norma (magnitud) del punto.
- **Rotación:** Se rota el punto alrededor del origen (centro de la circunferencia) en un ángulo especificado.

El desarrollo se realizó en celdas independientes de un Jupyter Notebook, permitiendo probar y visualizar cada operación por separado.

### 3 Resultados Experimentales

El análisis se realizó siguiendo estos pasos:

1. **Inicialización de la escena:** Se creó una escena con una circunferencia de radio 5 y un punto inicial en  $(2, 3)$ .
2. **Desplazamiento:** Se desplazó el punto 1 unidad a la derecha. Las nuevas coordenadas se mostraron en la consola y en la gráfica.
3. **Simulación del movimiento:** Se generó una animación en la que el punto se desplazaba 0.5 unidades hacia arriba durante 20 frames, con un intervalo de 200 ms.
4. **Escalado:** Se multiplicaron las coordenadas del punto por 2, mostrando el resultado tanto en texto como gráficamente.
5. **Producto escalar y norma:** Se calculó el producto escalar con el vector  $(1, 1)$  y la norma del punto, cuyos valores se imprimieron en la salida.
6. **Rotación:** Se rotó el punto 45 grados y se presentaron las nuevas coordenadas en la gráfica.

La Figura ?? muestra un ejemplo representativo de la salida del entorno tras aplicar las operaciones vectoriales.

### 4 Conclusiones

La simulación en Python demuestra de forma efectiva cómo se pueden implementar y visualizar operaciones vectoriales en un entorno 2D. Utilizando operaciones vectorizadas con NumPy y animaciones con Matplotlib, se lograron transformar las coordenadas de un punto mediante desplazamiento, escalado, producto escalar, cálculo de norma y rotación. Los resultados obtenidos, tanto gráficos como numéricos, evidencian la eficiencia y claridad de la metodología aplicada, resultando adecuada para aplicaciones en robótica, gráficos computacionales y análisis de datos.