

Implementación de simulación de tráfico con Python usando el Modelo de Gas de Tráfico como herramienta didáctica para fomentar el interés en matemáticas y estadística en estudiantes de cuarto año de la Unidad Educativa Chimborazo.

Implementation of traffic simulation with Python using the Traffic Gas Model as a teaching tool to promote interest in mathematics and statistics in fourth-year students of the Chimborazo Educational Unit.

Juan Carlos Rodríguez León¹[0000-0002-5183-8650], Manuel Antonio Meneses Freire²[0000-0001-8182-3153]

¹Universidad Nacional de Chimborazo – Maestrante de la Dirección de Posgrado - Ecuador, carlos.rodriguez@unach.edu.ec,

²Universidad Nacional de Chimborazo – Facultad de Ingeniería – Carrera de Ingeniería en Telecomunicaciones - Ecuador, ameneses@unach.edu.ec.

CITA EN APA:

Juan Carlos , J. C., & Meneses Freire, M. A. (2025). Implementación de simulación de tráfico con Python usando el Modelo de Gas de Tráfico como herramienta didáctica para fomentar el interés en matemáticas y estadística en estudiantes de cuarto año de la Unidad Educativa Chimborazo. Technology Rain Journal, 4(1). <https://doi.org/10.55204/trj.v4i1.e67>

Recibido: 21 de febrero-2025

Aceptado: 25 de marzo-2025

Publicado: 7 de abril-2025

Technology Rain Journal
ISSN: 2953-464X



Los contenidos de este artículo están bajo una licencia de Creative Commons Attribution 4.0 International (CC BY 4.0) Los autores conservan los derechos morales y patrimoniales de sus obras.

Resumen. La matemática y la estadística en las instituciones educativas se perciben como materias abstractas y difíciles, reduciendo la motivación de los estudiantes para profundizar en estos conocimientos esenciales. En este contexto, la utilización de simulaciones ayuda a los estudiantes a visualizar conceptos complejos en aplicaciones prácticas fomentando un aprendizaje significativo. El objetivo general fue implementar la simulación de tráfico mediante código Python usando el Modelo Gas de Tráfico como una herramienta didáctica para aumentar el interés y motivación en estudiantes de cuarto año de la Unidad Educativa PCEI Chimborazo. La metodología fue cuantitativa, con una muestra de 60 alumnos. Como resultados se pudo observar que el 95% de los estudiantes desarrolló interés en la temática y el 93,3% se sintió motivado tras la intervención. Los resultados ponen en evidencia que el uso de simulaciones convierte a la matemática y estadística en ciencias más inspiradoras, accesibles y relevantes para los estudiantes.

Palabras Clave: Aprendizaje significativo, Python, Motivación, Enseñanza, Simulación.

Abstract: Mathematics and statistics are perceived in educational institutions as abstract and difficult subjects, reducing students' motivation to delve deeper into these essential knowledges. In this context, the use of simulations helps students visualize complex concepts in practical applications, fostering meaningful learning. The overall objective was to implement a traffic simulation using Python code using the Gas Traffic Model as a teaching tool to increase interest and motivation among fourth-year students at the PCEI Chimborazo Educational Unit. The methodology was quantitative, with a sample of 60 students. The results showed that 95% of students developed interest in the subject and 93.3% felt motivated after the intervention. The results demonstrate that the use of simulations makes mathematics and statistics more inspiring, accessible, and relevant for students.

Keywords: Meaningful learning, Python, Motivation, Teaching, Simulation.

1. INTRODUCCIÓN

En las instituciones educativas, la matemática y la estadística se perciben como materias abstractas y difíciles, lo que reduce significativamente la motivación de los estudiantes para profundizar en estos conocimientos fundamentales (Mayorga et al., 2025). Esta desmotivación está vinculada a métodos de enseñanza tradicionales que enfatizan la memorización y el uso de procedimientos repetitivos sin demostrar la aplicabilidad de los conceptos en situaciones reales (Ureta et al., 2024). Como consecuencia, muchos alumnos experimentan ansiedad matemática, lo que se traduce en un desinterés progresivo que afecta su rendimiento académico y su disposición para explorar disciplinas clave como la ciencia, la tecnología, la ingeniería, la estadística y las matemáticas (Flores et al., 2024; Pino et al., 2022).

El desinterés en estas áreas se intensifica por la falta de experiencias prácticas y dinámicas que fomenten la curiosidad y el pensamiento crítico. La enseñanza centrada exclusivamente en la teoría limita la capacidad de los estudiantes para establecer conexiones significativas entre los contenidos y su entorno cotidiano, lo que refuerza la percepción de que las matemáticas y la estadística carecen de relevancia en su vida diaria (Reyes et al., 2021; Sombría et al., 2024).

En este contexto, el uso de simulaciones computacionales emerge como una estrategia eficaz para transformar la enseñanza de las matemáticas y la estadística. Las simulaciones permiten modelar fenómenos complejos de manera visual e intuitiva, facilitando la comprensión de conceptos abstractos a través de representaciones dinámicas y manipulables. Al integrar simulaciones en el aula, los estudiantes interactúan con modelos matemáticos, exploran distintas variables y observan cómo los cambios afectan los resultados, promoviendo un aprendizaje activo y significativo. De esta manera, se logra reducir la ansiedad matemática, incrementar la motivación y mejorar la percepción sobre la utilidad de estas disciplinas en la resolución de problemas del mundo real (Arróliga, 2024; Duarte, 2024; Vieira, 2021).

La simulación de tráfico utilizando Google Colab se presenta como una alternativa didáctica efectiva para fortalecer la enseñanza de matemáticas y estadística (Esnaola, 2022; Sanz et al., 2023). La modelización de sistemas dinámicos, como el tráfico vehicular, permite ilustrar principios fundamentales de probabilidad, estadísticas descriptivas, teoría de colas y sistemas complejos, en donde los estudiantes observan cómo pequeños cambios en los parámetros iniciales influyen en la evolución de un sistema en el tiempo, lo que fomenta el pensamiento crítico y analítico. Además, la implementación de estos modelos en entornos de programación como Python no solo mejora la comprensión matemática, sino que también

introduce a los estudiantes en el pensamiento computacional, una habilidad indispensable en la era digital (Pliego, 2023; Riera, 2023; Salazar et al., 2024).

El objetivo general de esta investigación es implementar la simulación de tráfico mediante código en Python, utilizando el Modelo de Gas de Tráfico como una herramienta didáctica para incrementar el interés y la motivación en estudiantes de cuarto año de la Unidad Educativa PCEI Chimborazo en noviembre de 2024. Este modelo, basado en la analogía entre el comportamiento de los vehículos en una carretera y el movimiento de partículas en un gas, permite explorar conceptos como densidad, velocidad promedio, probabilidad de congestión y propagación de ondas de choque en el tráfico. A través de la experimentación con estos modelos, los estudiantes desarrollarán una comprensión más profunda de los fenómenos estocásticos y las ecuaciones diferenciales que rigen el tráfico vehicular.

Para alcanzar este objetivo, la investigación adopta una metodología cuantitativa con un diseño descriptivo transversal, aplicando encuestas a los estudiantes antes y después de la intervención para medir los niveles de interés y motivación en relación con el aprendizaje de matemáticas y estadística. Estos instrumentos permiten recopilar datos cuantitativos sobre la percepción de los alumnos respecto a la utilidad de la simulación como herramienta de aprendizaje y su impacto en la comprensión de los temas abordados.

Se diseñó un plan de clase para introducir a los estudiantes en el uso de simulaciones como herramienta para fortalecer el aprendizaje de matemáticas y estadística. La sesión incluyó una introducción teórica sobre modelos matemáticos aplicados al tráfico vehicular, explicando el Modelo de Gas de Tráfico y sus fundamentos. Se abordaron conceptos como la distribución de velocidades, la influencia de la densidad vehicular y la formación de patrones de congestión. A través de preguntas reflexivas y casos de estudio sobre movilidad urbana, se incentivó la participación activa. Luego, se presentaron simulaciones en Python que permitieron visualizar en tiempo real la evolución del tráfico, integrando reglas como aceleración, frenado e interacción con semáforos, lo que facilitó la comprensión mediante representaciones gráficas y animaciones en formato GIF.

Finalmente, se aplicaron encuestas antes y después de la actividad para evaluar los cambios en el interés y motivación de los estudiantes. Se realizó un resumen de los conceptos abordados y se incentivó a los participantes a continuar explorando el uso de simulaciones como una herramienta innovadora en el aprendizaje de ciencias exactas. Este enfoque busca no solo mejorar la motivación y el rendimiento en matemáticas y estadística, sino también promover el desarrollo de habilidades analíticas y computacionales esenciales en el ámbito académico y profesional.

2. METODOLOGÍA O MATERIALES Y MÉTODOS

La metodología empleada en este estudio fue de carácter cuantitativo, con un diseño descriptivo transversal, permitiendo observar y describir las características de una población específica en un momento determinado. Este enfoque se centra en medir el impacto de una intervención didáctica específica (la simulación de tráfico con Python) en el interés y motivación de los estudiantes en las áreas de matemáticas y estadística.

Para evaluar el impacto de la implementación de la simulación de tráfico como herramienta didáctica, se utilizaron encuestas aplicadas a los estudiantes antes y después de la intervención. Las encuestas previas (pre-intervención) permitieron obtener un diagnóstico inicial sobre el nivel de interés y motivación que los estudiantes tienen hacia las matemáticas y la estadística antes de la implementación de la herramienta. Posteriormente, se aplicó una encuesta posterior a la intervención (post-intervención) con el fin de registrar cualquier cambio en la motivación y percepción de los estudiantes hacia estas materias después de haber experimentado la simulación.

El uso de encuestas permitió recopilar datos cuantitativos para analizar y comparar los resultados antes y después de la intervención. Estos datos facilitaron el análisis estadístico de las variaciones en las respuestas de los estudiantes, para identificar si la herramienta utilizada (simulación de tráfico con Python) logra un efecto positivo en la percepción y compromiso de los estudiantes hacia el aprendizaje de matemáticas y estadística. Con base en los resultados obtenidos, se extraen conclusiones sobre la efectividad de la herramienta didáctica implementada, así como recomendaciones para futuras aplicaciones en contextos educativos similares.

Se empleó un muestreo aleatorio no probabilístico, la muestra estuvo compuesta por 60 estudiantes de cuarto año de la Unidad Educativa PCEI Chimborazo, quienes fueron seleccionados de manera aleatoria dentro de los grupos de cuarto año disponibles durante el mes de febrero de 2025.

Los instrumentos utilizados fueron cuestionarios estructurados para cuantificar el nivel de interés y motivación. La validación de los instrumentos se llevó a cabo mediante la revisión de expertos en el área educativa Anexo 2, asegurando la confiabilidad y pertinencia de las preguntas para captar adecuadamente las variables de interés y asegurar la validez del estudio.

3. RESULTADOS Y DISCUSIÓN

3.1. Desarrollo de la clase

Se diseñó un plan de clase detallado, adjunto como Anexo 1, con el propósito de introducir a los estudiantes de cuarto año de la Unidad Educativa Chimborazo en el uso de simulaciones como una herramienta para fortalecer el aprendizaje de matemáticas y estadística. La estructura de la clase incluyó una introducción teórica sobre la aplicación de modelos matemáticos en la simulación de

tráfico vehicular, seguida de la exposición del Modelo de Gas de Tráfico, donde se explicaron sus fundamentos matemáticos y estadísticos.

Para fortalecer la comprensión, se fomentó la participación activa mediante preguntas orientadas a la reflexión y la aplicación del modelo en escenarios reales. Posteriormente, se presentó simulaciones en Python que permitió la visualización en tiempo real del comportamiento del tráfico bajo distintas condiciones. Los códigos desarrollados implementaron reglas básicas como la aceleración y frenado de los autos, la simulación de incertidumbre en la conducción y la visualización de la evolución del tráfico en cada iteración mediante gráficos en escala de grises. Adicionalmente, se mostraron simulaciones avanzadas que incluían la interacción de los vehículos con semáforos, generando animaciones en formato GIF para una mejor comprensión del fenómeno.

Finalmente, se aplicaron encuestas antes y después de la actividad para evaluar cambios en el interés y motivación de los estudiantes. Se realizó un resumen de los conceptos abordados y se incentivó a los participantes a seguir explorando el uso de simulaciones en el aprendizaje de ciencias exactas.

3.2. Modelo de gas de tráfico

Densidad del tráfico (ρ):

La densidad representa el número de vehículos por unidad de longitud de carretera:

$$\rho = \frac{N}{L}$$

donde:

- N es el número total de vehículos en la carretera.
- L es la longitud de la carretera.

Velocidad media (\bar{v}):

Se define como el promedio de las velocidades de todos los vehículos en la carretera:

$$\bar{v} = \frac{1}{N} \sum_{i=1}^N v_i$$

dónde v_i es la velocidad del i -ésimo vehículo.

Flujo de tráfico (J):

El flujo de vehículos (o caudal) es la cantidad de autos que pasan por un punto por unidad de tiempo:

$$J = \rho \bar{v}$$

Este concepto muestra cómo la densidad y la velocidad afectan la fluidez del tránsito.

Ecuación de Continuidad del Tráfico:

Similar a la ecuación de continuidad en dinámica de fluidos, describe la conservación de la cantidad de vehículos:

$$\frac{\partial \rho}{\partial t} + \frac{\partial J}{\partial x} = 0$$

Esto indica que cualquier cambio en la densidad y en el tiempo se debe a un flujo neto de vehículos entrando o saliendo de un punto.

Relación Fundamental del Tráfico:

La velocidad promedio de los vehículos disminuye cuando la densidad del tráfico aumenta, lo que se modela con la función:

$$\bar{v}(\rho) = v_{max} \left(1 - \frac{\rho}{\rho_{max}} \right)$$

dónde:

v_{max} es la velocidad máxima posible de los autos en condiciones óptimas.

ρ_{max} es la densidad máxima, es decir, cuando los vehículos están completamente detenidos en congestión.

Este modelo se usa en la teoría del tráfico para analizar fenómenos como la formación de atascos, la dispersión de ondas de choque en el tráfico y la optimización del flujo vehicular en carreteras y ciudades.

En este caso, analizamos la relación entre la densidad del tráfico y el flujo de vehículos utilizando la ecuación fundamental:

$$J = \rho * v_{max} \left(1 - \frac{\rho}{\rho_{max}} \right)$$

Este código grafica la relación fundamental del tráfico, mostrando cómo el flujo cambia con la densidad.

```
import numpy as np
import matplotlib.pyplot as plt

# Parámetros del modelo
v_max = 30 # Velocidad máxima (km/h)
rho_max = 1 # Densidad máxima (vehículos/km)

# Definir rango de densidades
rho = np.linspace(0, rho_max, 100) # Densidad del tráfico (vehículos/km)

# Calcular el flujo de tráfico usando la ecuación fundamental
J = rho * v_max * (1 - rho / rho_max)
```

```

# Graficar la relación fundamental del tráfico
plt.figure(figsize=(8, 5))
plt.plot(rho, J, label="Flujo de tráfico", color="b", linewidth=2)
plt.axvline(rho_max / 2, linestyle="--", color="r", label="Densidad crítica")
plt.xlabel("Densidad del tráfico (vehículos/km)")
plt.ylabel("Flujo de tráfico (vehículos/h)")
plt.title("Relación Fundamental del Tráfico")
plt.legend()
plt.grid()
plt.show()

```

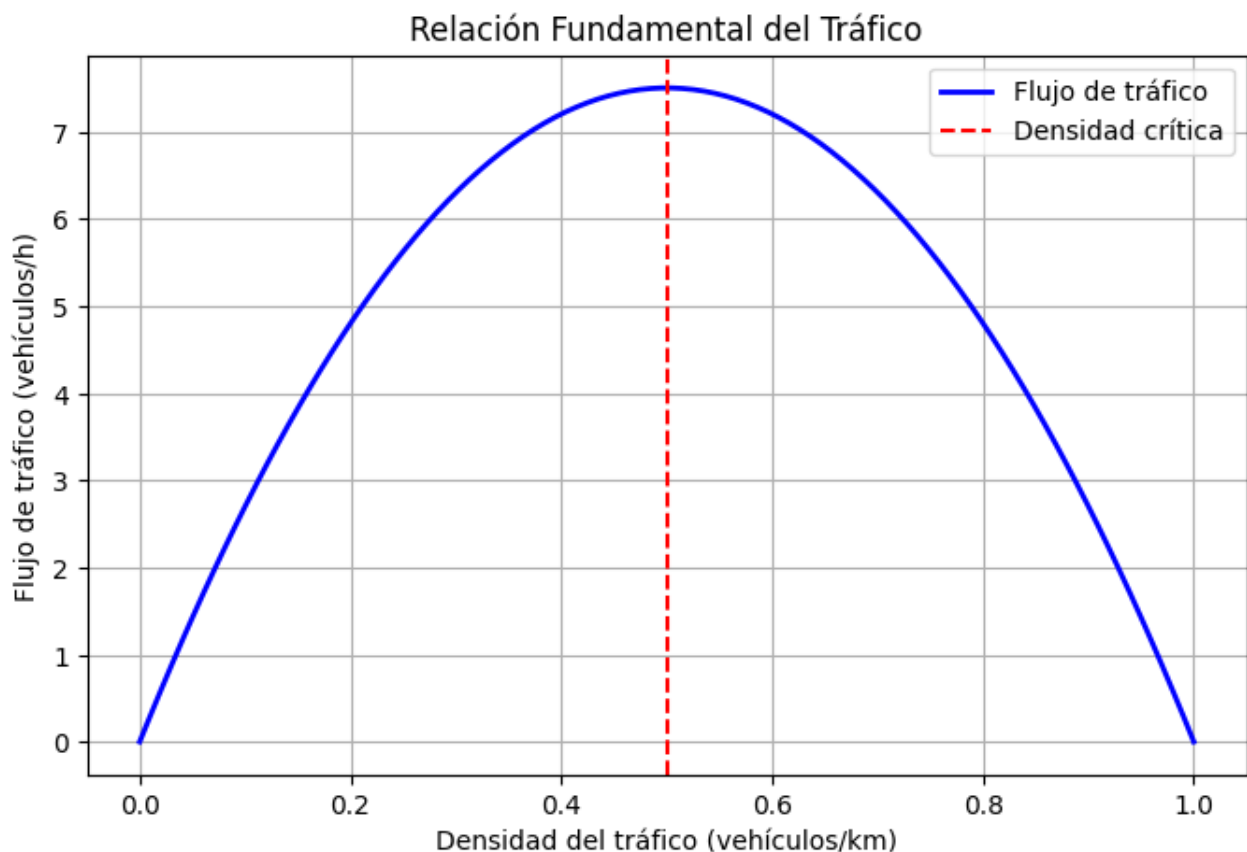
Definimos los parámetros:

Velocidad máxima de los autos ($v_{max}=30$ km/h).

Densidad máxima de la carretera ($\rho_{max}=1$ vehículo/km).

El flujo aumenta a medida que la densidad crece, pero después de cierto punto (densidad crítica), el flujo disminuye debido a la congestión.

Cuando la densidad sigue aumentando, el flujo disminuye porque los autos están más congestionados y se mueven más lento.



3.3. Aplicación de las simulaciones

Inicialización del escenario 1: Se creó una carretera con celdas vacías y autos ubicados aleatoriamente Anexo 1.

Reglas del modelo:

Los autos aceleran hasta una velocidad máxima.

Frenan si hay un auto adelante.

Pueden frenar aleatoriamente para simular comportamiento humano.

Evolución del tráfico: Cada iteración muestra cómo cambian las posiciones y velocidades de los autos.

Visualización: Matriz de color gris para mostrar la carretera en cada paso.

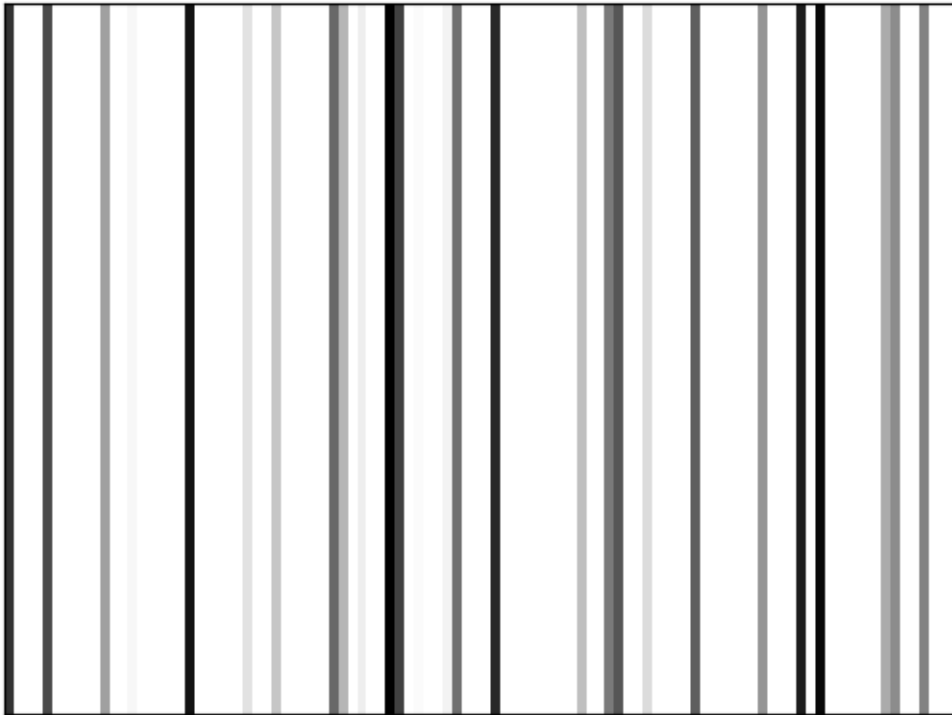


Fig. 1. Simulación de tráfico

Escenario 2:

Representa los autos como partículas rojas ('r0') en la carretera Simula el movimiento del tráfico en cada cuadro Guarda la animación como un GIF llamado "simulacion_trafico_particulas.gif" Se muestra directamente en Google Colab. Anexo 2.

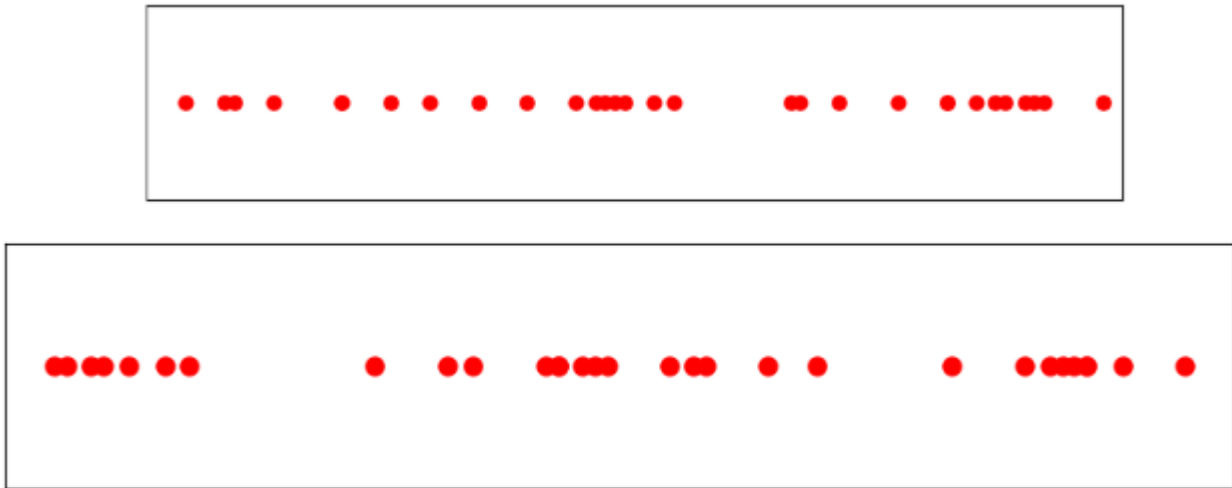


Fig. 2. Simulación de tráfico en partículas

Escenario 3:

Simula un sistema de tráfico con vehículos que se detienen en un semáforo rojo y avanzan cuando el semáforo es verde Anexo 3.

Cambia el estado del semáforo cada cierto número de pasos.

Genera un GIF que muestra la interacción entre los vehículos y el semáforo.

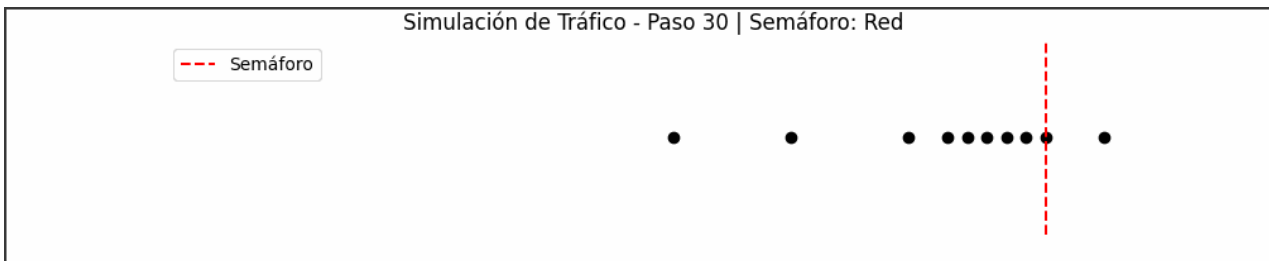


Fig. 3. Simulación de tráfico y semáforo

3.4. Resultados de la actividad

Tabla 1. Resultados antes y después de la actividad

Pre1		Nada interesado		Post1		No	
Interesado				Si			
Recuento	% de N totales de fila	Recuento	% de N totales de fila	Recuento	% de N totales de fila	Recuento	% de N totales de fila
8	13,30%	52	86,70%	57	95,00%	3	5,00%
Pre2		Nada Motivado		Post2		No	
Motivado				Si			
Recuento	% de N totales de fila	Recuento	% de N totales de fila	Recuento	% de N totales de fila	Recuento	% de N totales de fila
8	13,30%	52	86,70%	56	93,30%	4	6,70%

Los resultados obtenidos a partir de la aplicación de encuestas antes y después de la actividad demostraron un incremento significativo en el interés y la motivación de los estudiantes hacia las matemáticas y la estadística. En la encuesta inicial, solo el 13,3% de los estudiantes manifestó estar

interesado en el tema, mientras que el 86,7% indicó no tener interés. Tras la experiencia con la simulación en Python, el 95% de los estudiantes expresó haber desarrollado interés en la temática, reflejando un aumento considerable en la predisposición hacia el aprendizaje.

En términos de motivación, antes de la actividad el 13,3% de los participantes se consideraba motivado, mientras que el 86,7% no lo estaba. Luego de la implementación de la simulación, el 93,3% de los estudiantes se sintió motivado, evidenciando un impacto positivo en su percepción del aprendizaje de matemáticas y estadística. Estos resultados validan la efectividad del uso de simulaciones interactivas como estrategia pedagógica para mejorar el interés y la motivación en el aula.

Los resultados de este estudio respaldan la eficacia de la simulación de tráfico implementada en Google Colab como una herramienta didáctica en el aula, favoreciendo el incremento del interés y la motivación de los estudiantes en matemáticas y estadística.

Estos hallazgos coinciden con investigaciones previas que han demostrado cómo las tecnologías basadas en simulaciones mejoran la comprensión de conceptos complejos en entornos educativos.

Estudios como los de (Asqui, 2024; Castellanos et al., 2024; J. López, 2023) destacan que las simulaciones interactivas no solo refuerzan el aprendizaje teórico, sino que también promueven un enfoque activo al involucrar a los estudiantes en la resolución de problemas aplicados.

De manera similar, (Diez et al., 2023; Enriquez, 2024; García, 2024; Jara, 2024) resaltan que plataformas como Google Colab permiten integrar programación y visualización de datos en la enseñanza, haciendo los contenidos más accesibles y atractivos para los alumnos.

En línea con estos estudios, la presente investigación demuestra que el uso de simulaciones en Google Colab facilita la conexión entre los conceptos matemáticos y estadísticos y su aplicación en situaciones reales, como el tráfico vehicular, fomentando la participación activa y el compromiso de los estudiantes en el aprendizaje. Además, su implementación fortalece el desarrollo de habilidades tecnológicas, aspecto indispensable en el entorno educativo actual.

La incorporación de herramientas innovadoras como Python en el aula no solo es viable, sino que representa una estrategia efectiva para transformar la enseñanza de áreas tradicionalmente percibidas como desafiantes y con bajos niveles de motivación estudiantil (M. Flores, 2024; Huilca, 2024; F. López, 2024; Pinargote et al., 2024).

4. CONCLUSIONES

Los resultados de esta investigación evidencian que el uso de simulaciones interactivas en la enseñanza de matemáticas y estadística genera un impacto positivo en la motivación y el interés de los estudiantes. A partir de la evaluación inicial, se identificó un nivel bajo de motivación hacia

estas materias, confirmando la necesidad de innovar en las estrategias pedagógicas para mejorar la percepción estudiantil.

La simulación de tráfico basada en el Modelo de Gas de Tráfico, diseñada e implementada en Python, permitió a los estudiantes visualizar de manera dinámica y aplicada los conceptos matemáticos y estadísticos. Su integración didáctica facilitó la conexión entre la teoría y su aplicabilidad en contextos reales, promoviendo un aprendizaje más significativo y participativo.

La comparación entre la encuesta pre y post-intervención mostró un aumento en el nivel de interés y motivación de los estudiantes, lo que demuestra que la incorporación de herramientas tecnológicas en el aula puede ser una estrategia efectiva para reducir la percepción de dificultad y desinterés en matemáticas y estadística. Estos hallazgos resaltan la importancia de continuar explorando metodologías innovadoras que refuercen el aprendizaje mediante experiencias interactivas y contextualizadas.

AGRADECIMIENTOS

Agradecimiento a la Universidad Nacional de Chimborazo y Unidad Educativa PCEI Chimborazo, a la Red de Investigación en Ingeniería e Informática Ri3.

FINANCIACIÓN

Esta investigación no recibió financiamiento externo.

CONFLICTO DE INTERESES

El autor declara no tener ningún conflicto de interés.

CONTRIBUCIÓN DE AUTORÍA

En concordancia con la taxonomía establecida internacionalmente para la asignación de créditos a autores de artículos científicos (<https://credit.niso.org/>). Los autores declaran sus contribuciones en la siguiente matriz:

	Juan Carlos Rodríguez León	Manuel Antonio Meneses Freire	
Participar activamente en:			
Conceptualización	X	X	
Análisis formal	X	X	
Adquisición de fondos	X		
Investigación	X		
Metodología	X	X	
Administración del proyecto	X		
Recursos	X		
Redacción –borrador original	X		
Redacción –revisión y edición	X	X	
La discusión de los resultados	X	X	
Revisión y aprobación de la versión final del trabajo.	X	X	

REFERENCIAS

- Arróliga, J. (2024). Innovación en prácticas docentes y desarrollo de competencias digitales en la Universidad Jean Jacques Rousseau. *Revista científica de estudios sociales*, 3(5), Article 5. <https://doi.org/10.62407/rces.v3i5.138>
- Asqui, B. (2024). Recursos educativos digitales para mejorar el aprendizaje en matemáticas. *Esprint Investigación*, 3(1), 59-72.
- Castellanos, M., Zuña, C., Andrade, M., & Ruiz, A. (2024). Un Enfoque Innovador: Cooperación y Tecnología en el Aula. *Revista Científica UISRAEL*, 11(2), Article 2. <https://doi.org/10.35290/rcui.v11n2.2024.1136>
- Diez, M., Rojas, I., Rodríguez, V., & Coello, L. (2023). Herramientas interactivas para el trazado de las características mecánicas en accionamientos eléctricos. *Ingeniería Energética*, 44(3), 28-37.
- Duarte, N. (2024). *Tecnología educativa: Estrategias para el fortalecimiento del aprendizaje del inglés en niños a través de asesorías académicas*. <http://repository.unad.edu.co/handle/10596/65009>
- Enriquez, M. (2024). *La cultura digital de estudiantes de noveno grado en el desarrollo de objetivos específicos matemáticos del Programa de Bachillerato Internacional*. <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/27958>
- Eснаоla, L. (2022). Experiencia de uso de Google Colaboratory en Sistemas Inteligentes. *III WORKSHOP DE INNOVACIÓN Y TRANSFORMACIÓN EDUCATIVA*. <http://repositorio.unnoba.edu.ar/xmlui/handle/23601/652>
- Flores, A., Parrales, D., Mendoza, I., & Pisco, D. (2024). Uso de Padlet en la Motivación y el Aprendizaje Colaborativo en Ciencias Naturales: Un Estudio Descriptivo en la Educación Básica Superior. *Arandu UTIC*, 11(2), Article 2. <https://doi.org/10.69639/arandu.v11i2.378>
- Flores, M. (2024). Uso de Google Classroom en la enseñanza del inglés en docentes del nivel secundario de Huancabamba, 2024. *Repositorio Institucional - UCV*. <https://repositorio.ucv.edu.pe/handle/20.500.12692/151299>
- García, M. (2024). *Código Python para la simulación de detección de exoplanetas*. <https://hdl.handle.net/20.500.12371/21890>
- Huillca, A. (2024). *Análisis aplicando redes neuronales artificiales para incrementar la producción de acarreo de material estéril de las fases 5 y 3 del tajo Ferrobamba – Las Bambas – Apurímac*. <https://repositorio.unsaac.edu.pe/handle/20.500.12918/8579>
- Jara, F. (2024). *La inteligencia artificial generativa en el aula de física y química*. <http://dspace.umh.es/handle/11000/32995>
- López, F. (2024). *El aula prosocial digital. Uso de TIC para el desarrollo emocional y social en educación: Una propuesta educativa*. <http://dspace.umh.es/handle/11000/33149>
- López, J. (2023). *Últimos avances en redes de cápsulas orientados a la reducción de carga computacional y uso de memoria: Evaluación y aplicaciones*. <http://hdl.handle.net/10317/12839>
- Mayorga, L., Hernández, C., López, L., & Mayorga, M. (2025). Aporte de la asignatura matemáticas para carreras universitarias. *593 Digital Publisher CEIT*, 10(Extra 1), 231-247.
- Pinargote, J., Lino, V., & Vera, B. (2024). Python en la enseñanza de las Matemáticas para estudiantes de nivelación en Educación Superior. *MQRInvestigar*, 8(3), Article 3. <https://doi.org/10.56048/MQR20225.8.3.2024.3966-3989>

Pino, N., Melchor, N., & Mejía, C. (2022). La experiencia de la construcción de modelos matemáticos aplicado ala ecología:

Implementando simulaciones computacionales como punto de inicio. *Campus*, 27(33), 131-142.

<https://doi.org/10.24265/campus.2022.v27n33.09>

Pliego, J. (2023). *Evolución estadística de separaciones, nulidades y divorcios en los años 2013-2020*.

<http://dspace.umh.es/handle/11000/29754>

Reyes, M., Ambrocio, S., & Vélez, D. (2021). Inducción a las matemáticas empleando objetos de aprendizaje y diseño

instruccional. *XIKUA Boletín Científico de la Escuela Superior de Tlahuelilpan*, 9(18), Article 18.

<https://doi.org/10.29057/xikua.v9i18.7320>

Riera, L. (2023). *Desarrollo de una aplicación interactiva para el diseño de controladores PID basada en Python*.

<http://dspace.umh.es/handle/11000/29517>

Salazar, L., López, E., & Núñez, M. (2024). Estrategias para fomentar el aprendizaje activo en el aula de pedagogía Universitaria.

Polo del Conocimiento, 9(6), Article 6. <https://doi.org/10.23857/pc.v9i6.7331>

Sanz, A., García, E., Sánchez, M., Puente, H., Calderón, D., Cámara, N., García, S., Moreno, D., Blanco, J., Díaz, A., & Gutiérrez,

J. (2023). *Aula de innovación tecnológica para el uso de aplicaciones informáticas accesibles en las ciencias sociales*.

<https://hdl.handle.net/20.500.14352/87318>

Sombría, J., Díaz, J., & Dormido, S. (2024). Diseño de un curso de señales y sistemas basado en herramientas interactivas.

Jornadas de Automática, 45, Article 45. <https://doi.org/10.17979/ja-cea.2024.45.10828>

Ureta, M., Chicaiza, I., Zumba, J., & Llerena, L. (2024). El rol de la tecnología en la Educación Básica. *Ciencia y Educación*, 5(8),

Article 8. <https://doi.org/10.5281/zenodo.13112598>

Vieira, A. (2021). Matemática com Python no colab e Google planilhas. *Anais do ESEM - Encontro Sul-Mato-Grossense de*

Educação Matemática. <https://periodicos.ufms.br/index.php/aesmgemat/article/view/13895>

ANEXOS

Anexo 1. Simulación de tráfico

```
import numpy as np
import matplotlib.pyplot as plt
import time

# Parámetros de la simulación
num_celdas = 100 # Número de celdas en la carretera (longitud)
num_autos = 30 # Número de autos en la carretera
num_pasos = 50 # Número de iteraciones de la simulación
v_max = 5 # Velocidad máxima de los autos
p_freno = 0.2 # Probabilidad de frenado aleatorio (simula incertidumbre humana)

# Inicialización de la carretera con -1 indicando celdas vacías
carretera = np.full(num_celdas, -1)
```

```

# Ubicación inicial de los autos (espaciados aleatoriamente)
posiciones = np.random.choice(range(num_celdas), num_autos, replace=False)
velocidades = np.zeros(num_autos, dtype=int)

# Asignar autos a la carretera
for i, pos in enumerate(posiciones):
    carretera[pos] = i

# Función para actualizar el tráfico
def actualizar_trafico():
    global carretera, velocidades
    nueva_carretera = np.full(num_celdas, -1)
    nuevas_velocidades = np.copy(velocidades)

    for i in range(num_autos):
        pos_actual = posiciones[i]
        distancia = 1 # Contar espacio libre delante
        while (carretera[(pos_actual + distancia) % num_celdas] == -1) and distancia
<= v_max:
            distancia += 1
        distancia -= 1 # Ajustar distancia real

        # Reglas del modelo de Nagel-Schreckenberg
        if velocidades[i] < v_max:
            nuevas_velocidades[i] += 1 # Acelerar si no está en velocidad máxima
        if velocidades[i] > distancia:
            nuevas_velocidades[i] = distancia # Reducir velocidad para evitar
colisión
        if np.random.rand() < p_freno and nuevas_velocidades[i] > 0:
            nuevas_velocidades[i] -= 1 # Frenado aleatorio

        # Actualizar posición
        nueva_pos = (pos_actual + nuevas_velocidades[i]) % num_celdas
        nueva_carretera[nueva_pos] = i
        posiciones[i] = nueva_pos

    carretera = nueva_carretera
    velocidades = nuevas_velocidades

# Función para visualizar el tráfico
def mostrar_trafico():
    plt.imshow([carretera], cmap="Greys", aspect="auto")
    plt.xticks([])
    plt.yticks([])
    plt.show()

```

```
# Simulación
for _ in range(num_pasos):
    actualizar_trafico()
    mostrar_trafico()
    time.sleep(0.2) # Pausa para visualizar los cambios en tiempo real
```

Anexo 2. Simulación de tráfico en partículas

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import imageio.v2 as imageio

# Parámetros de la simulación
num_celdas = 100
num_autos = 30
num_pasos = 50
v_max = 5
p_freno = 0.2

# Inicialización
carretera = np.full(num_celdas, -1)
posiciones = np.random.choice(range(num_celdas), num_autos, replace=False)
velocidades = np.zeros(num_autos, dtype=int)

for i, pos in enumerate(posiciones):
    carretera[pos] = i

# Función de actualización
def actualizar_trafico():
    global carretera, velocidades
    nueva_carretera = np.full(num_celdas, -1)
    nuevas_velocidades = np.copy(velocidades)

    for i in range(num_autos):
        pos_actual = posiciones[i]
        distancia = 1
        while (carretera[(pos_actual + distancia) % num_celdas] == -1) and distancia
        <= v_max:
            distancia += 1
        distancia -= 1

        if velocidades[i] < v_max:
            nuevas_velocidades[i] += 1
        if velocidades[i] > distancia:
            nuevas_velocidades[i] = distancia
```

```

    if np.random.rand() < p_freno and nuevas_velocidades[i] > 0:
        nuevas_velocidades[i] -= 1

    nueva_pos = (pos_actual + nuevas_velocidades[i]) % num_celdas
    nueva_carretera[nueva_pos] = i
    posiciones[i] = nueva_pos

    carretera = nueva_carretera
    velocidades = nuevas_velocidades

# Crear la animación con partículas en movimiento
fig, ax = plt.subplots(figsize=(10, 2))
ax.set_xlim(0, num_celdas)
ax.set_ylim(-1, 1)
ax.set_yticks([])
ax.set_xticks([])
puntos, = ax.plot([], [], 'ro', markersize=8) # 'ro' significa puntos rojos (autos)

def init():
    puntos.set_data([], [])
    return puntos,

def update(frame):
    actualizar_trafico()
    puntos.set_data(posiciones, np.zeros(num_autos)) # Autos en una sola fila
    return puntos,

ani = animation.FuncAnimation(fig, update, frames=num_pasos, init_func=init,
                               blit=True)

# Guardar la animación como GIF
ani.save("simulacion_trafico_particulas.gif", writer='pillow', fps=10)

# Mostrar el GIF en Google Colab
from IPython.display import Image
Image(filename="simulacion_trafico_particulas.gif")

```

Anexo 3. Simulación de tráfico y semáforo

```

import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import imageio.v2 as imageio

# Parámetros de la simulación
road_length = 50          # Longitud de la carretera

```

```

num_vehicles = 10      # Número de vehículos
max_speed = 5         # Velocidad máxima de los vehículos
num_steps = 30        # Número de pasos en la simulación
light_duration = 5    # Duración del semáforo verde

# Inicialización de posiciones y velocidades
positions = np.sort(np.random.choice(road_length, num_vehicles, replace=False)) # Posiciones iniciales
speeds = np.zeros(num_vehicles, dtype=int) # Velocidades iniciales en cero
light_state = 'green' # Estado inicial del semáforo
frames = [] # Lista para almacenar los frames de la simulación

# Función para actualizar las posiciones de los vehículos
def update_positions():
    global positions, speeds, light_state
    for i in range(num_vehicles):
        # Si el semáforo está en rojo y el vehículo está cerca del semáforo, se detiene
        if light_state == 'red' and positions[i] >= road_length - 5:
            speeds[i] = 0 # Detener el vehículo
        else:
            # Calcular distancia al vehículo de adelante
            distance = (positions[(i + 1) % num_vehicles] - positions[i] - 1) % road_length
            # Aumentar la velocidad hasta el máximo o distancia
            speeds[i] = min(speeds[i] + 1, max_speed, distance)
            # Actualizar la posición
            positions[i] = (positions[i] + speeds[i]) % road_length

# Función para cambiar el estado del semáforo
def change_light(step):
    global light_state
    if step % (light_duration * 2) < light_duration:
        light_state = 'green'
    else:
        light_state = 'red'

# Simulación paso a paso y creación de imágenes
for step in range(num_steps):
    change_light(step) # Cambiar el estado del semáforo
    update_positions() # Actualizar posiciones y velocidades de vehículos

    plt.figure(figsize=(10, 2))
    plt.plot(positions, np.zeros(num_vehicles), 'o', color='black')
    plt.xlim(0, road_length)
    plt.ylim(-1, 1)

```

```
plt.title(f"Simulación de Tráfico - Paso {step + 1} | Semáforo:
{light_state.capitalize()}")
plt.axvline(x=road_length - 5, color='red' if light_state == 'red' else 'green',
linestyle='--', label='Semáforo')
plt.legend()
plt.axis('off')

# Guardar la figura como imagen
plt.savefig(f"frame_{step}.png")
plt.close()

# Crear GIF a partir de las imágenes guardadas
imagenes = [imageio.imread(f"frame_{step}.png") for step in range(num_steps)]
imageio.mimsave("simulacion_trafico_semaforo.gif", imagenes, fps=5)

# Mostrar el GIF en Google Colab
from IPython.display import Image
Image(filename="simulacion_trafico_semaforo.gif")
```