



FICHA DE TALLER

□ Datos generales:

Título:	Simulación numérica de problemas de difusión: del modelo al paralelismo				
Instructor o docente:	Alicia Margarita de la Mora Cebada				
Duración total de taller:	5 sesiones virtuales de 2 hrs. cada una.				
Área:	<input type="checkbox"/>	Formación para la docencia	<input type="checkbox"/>	Formación para la gestión universitaria	<input checked="" type="checkbox"/> Otra
Modalidad de impartición:	<input type="checkbox"/>	Presencial	<input checked="" type="checkbox"/>	A distancia	<input type="checkbox"/> Híbrida
Fechas:	1, 8, 15, 22 y 29 de octubre de 2025 de 12 a 14 horas				

Descripción:

Destinatarios:

- Profesionales de TI interesados en la implementación de simulaciones numéricas y soluciones paralelas para el análisis de fenómenos físicos.
- Personal académico, técnico o investigador que requiera formación práctica sobre métodos numéricos aplicados, programación científica y cómputo de alto rendimiento.
- Estudiantes de programas de licenciatura y posgrado en ciencias exactas, ingeniería o áreas afines, con interés en el modelado computacional.

Objetivo(s) o competencia(s):

Objetivo general:

Desarrollar e implementar una solución numérica a la ecuación de difusión utilizando diferencias finitas en Python y C, e introducir técnicas de paralelización con MPI.

Competencias específicas que desarrollará el participante:

- Comprender la formulación de la ecuación de difusión en 1D y 2D, y su discretización.
- Implementar esquemas numéricos explícitos en Python y C.
- Utilizar una solución analítica conocida (tipo senoidal) para validar los resultados numéricos.
- Medir y comparar el rendimiento de las implementaciones.
- Paralelizar las soluciones utilizando MPI en ambos lenguajes.
- Visualizar resultados y analizar el impacto del paralelismo.

Metodología y Actividades:

- Sesiones teórico-prácticas con acompañamiento docente y uso de códigos guía.
- Desarrollo individual de ejercicios de implementación y modificación de código.
- Visualización gráfica, análisis numérico y comparación de tiempos de ejecución entre versiones.
- Discusión grupal orientada al análisis de escalabilidad, eficiencia y comportamiento del paralelismo.



Mecanismo y criterios de evaluación:

- Ejercicios prácticos individuales: implementación y modificación de códigos numéricos y paralelos durante el taller.
- Actividad de validación: comparación con soluciones analíticas y discusión de resultados.
- Pruebas de escalabilidad: análisis en tiempos de ejecución y comportamiento del paralelismo.
- Reflexión final: participación en la discusión colectiva sobre los aprendizajes obtenidos.
- Formulario de retroalimentación: evaluación de la experiencia por parte del participante.

Criterios:

- Asistencia mínima del 80%.
- Cumplimiento y entrega de ejercicios prácticos.
- Comprensión de los conceptos aplicados en la implementación.
- Participación activa durante las sesiones.
- Capacidad de análisis en la interpretación de resultados.

Temario:

No. De sesión	Fecha	Hora	Temas a abordar
Sesión 1	1 de octubre	12:00 – 14:00	Introducción a la ecuación de difusión, diferencias finitas y herramientas básicas
Sesión 2	8 de octubre	12:00 – 14:00	Implementación secuencial del esquema explícito en Python y visualización de resultados
Sesión 3	15 de octubre	12:00 – 14:00	Implementación del mismo esquema en C, compilación y comparación de desempeño
Sesión 4	22 de octubre	12:00 – 14:00	Introducción a MPI y paralelización del código en Python (mpi4py) y C
Sesión 5	29 de octubre	12:00 – 14:00	Evaluación de rendimiento, escalabilidad y cierre del taller



Bibliografía

- Burden, R. L., Faires, J. D., & Burden, A. M. (2016). Numerical analysis (Tenth edition.). Cengage Learning.
- Gropp, W., Lusk, E., & Skjellum, A. (2000). Using mpi: Portable parallel programming with the message passing interface. MIT Press.
- MPI for Python—MPI for Python 4.1.0 documentation. (s/f). Retrieved from <https://mpi4py.readthedocs.io/en/stable/>
- Pacheco, P. S. (2009). Parallel programming with MPI (Nachdr.). Morgan Kaufmann.
- Python 3.13 documentation. (s/f). Python Documentation. Retrieved from <https://docs.python.org/3/>
- Zaccane, G. (2015). Python parallel programming cookbook: Master efficient parallel programming to build powerful applications using Python (1st ed). Packt Publishing.

Recursos y materiales requeridos

- Computadora personal con sistema operativo Windows, macOS o Linux.
- Memoria RAM mínima de 8 GB, recomendable para la ejecución fluida de simulaciones numéricas y pruebas locales.
- Conexión a Internet estable, indispensable para el acceso a plataformas en la nube, materiales del taller y ejecución remota.
- Cuenta de Google activa, necesaria para el uso de Google Colab durante las sesiones introductorias.
- Instalación local de Python 3 (recomendado)
- Conocimientos previos: fundamentos de ecuaciones diferenciales y nociones básicas de programación en Python o C.