Programación semanal

Recuerda que la suma de las puntuaciones de todas las actividades es de 15 puntos. Puedes hacer las que prefieras hasta conseguir un máximo de 10 puntos (que es la calificación máxima que se puede obtener en la evaluación continua.

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	**Resolución de problemas en las clases en directo**	**Actividades (15.0 puntos)**	**Experiencias en mi Porfolio**
Semana 1	**Tema 1.** Lenguajes y herramientas de programación científica 1.1. Introducción y objetivos 1.2. Clasificación de lenguajes de programación 1.3. Lenguajes científicos 1.4. Evaluación de rendimiento: *profiling* 1.5. Control de versiones: *git*	**Clase 1 (90 min.):** • Presentación de la asignatura • Presentación del proyecto • Problema: ¿Cómo configurar un entorno de desarrollo para programación científica?	*Asistencia a 2 clases en directo a lo largo de la asignatura. (0.5 puntos cada una)* Test Tema 1 (0.1 puntos)	1. Documentación sobre la instalación y configuración del entorno de desarrollo Python, incluyendo la elección del IDE y la configuración básica para programación científica. 2. Creación y descripción de un primer programa en Python que realice operaciones matemáticas básicas, mostrando la captura de pantalla del código y el resultado de ejecución. 3. Reflexión personal sobre los desafíos encontrados durante la configuración del entorno de desarrollo y cómo se resolvieron.

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 2	**Tema 2**. Introducción a Python 2.1. Introducción y objetivos 2.2. Entorno de desarrollo 2.3. Elementos básicos del lenguaje 2.4. Funciones 2.5. Manejo de archivos 2.6. Excepciones 2.7. Módulos 2.8. Programación orientada a objetos (POO) 2.9. Referencias bibliográficas 2.10. Cuaderno de ejercicios	**Clase 2 (60 min.):** • Problema: ¿Cómo se aplican los conceptos básicos de Python para resolver problemas sencillos?	Test Tema 2 (0.1 puntos)	1. Implementación de ejercicios que utilizan estructuras de control de flujo en Python, como loops y condicionales, con explicaciones del código y resultados. 2. Análisis comparativo de la sintaxis de Python con otro lenguaje de programación conocido por el estudiante, destacando las diferencias y similitudes. 3. Reflexión sobre el proceso de aprendizaje de Python y las estructuras de control, incluyendo los errores más comunes y cómo se superaron.

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 3	**Tema 3**. Eficiencia computacional 3.1. Introducción y objetivos 3.2 Complejidad en tiempo vs. espacio 3.3 Notaciones asintóticas 3.4 Referencias bibliográficas 3.5 Cuaderno de ejercicios	**Clase 3 (60 min.):** • Problema: ¿Cómo se evalúa la eficiencia de diferentes estructuras de datos y algoritmos en Python?	Test Tema 3 (0.1 puntos)	1. Análisis de la complejidad temporal y espacial de varios algoritmos implementados en Python. 2. Implementación y comparación de diferentes estructuras de datos para resolver un conjunto de problemas dados. 3. Reflexión sobre la selección de estructuras de datos y algoritmos específicos para optimizar la eficiencia computacional en diferentes escenarios. 4. Descripción y análisis de algoritmos de ordenación y búsqueda implementados, incluyendo casos de prueba y rendimiento. 5. Experimentación con técnicas de programación dinámica y heurísticas para resolver problemas específicos, documentando la eficacia y eficiencia de estas técnicas 6. Reflexión sobre el proceso de selección de algoritmos adecuados para diferentes tipos de problemas y el impacto de esta selección en el rendimiento del programa.
Semana 4	**Tema 4**. Estructuras de datos 4.1. Introducción y objetivos 4.2. Arrays y listas 4.3. Pilas y colas 4.4. Tablas hash 4.5. Grafos 4.6. Árboles 4.7. Montículos 4.8. Referencias bibliográficas 4.9. Cuaderno de ejercicios	**Clase 4 (60 min.):** • Problema: ¿Qué estructura de datos sería más eficiente para un problema de procesamiento de datos específico? • Presentación de la actividad 1	Actividad 1. Recorridos iterativos de un árbol binario (5.0 puntos) Test Tema 4 (0.1 puntos)	

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 5	**Tema 5**. Métodos algorítmicos de resolución de problemas 5.1. Introducción y objetivos 5.2. Algoritmos de ordenación 5.3. Algoritmos de búsqueda 5.4. Algoritmos voraces 5.5. Programación dinámica 5.6. Heurísticas 5.7. Referencias bibliográficas 5.8. Cuaderno de ejercicios	**Clase 5 (60 min.):** • Problema: ¿Cómo se seleccionan e implementan algoritmos avanzados para optimizar la solución de problemas complejos?		
Semana 6	**Tema 5**. Métodos algorítmicos de resolución de problemas (continuación) 5.1. Introducción y objetivos 5.2. Algoritmos de ordenación 5.3. Algoritmos de búsqueda 5.4. Algoritmos voraces 5.5. Programación dinámica 5.6. Heurísticas 5.7. Referencias bibliográficas 5.8. Cuaderno de ejercicios	**Clase 6 (60 min.):** • Problema: ¿Cuál es el impacto de utilizar programación dinámica y heurísticas en la complejidad temporal y espacial de la solución de problemas comunes?	Test Tema 5 (0.1 puntos)	

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 7	**Tema 6**. Computación científica en Python 6.1. Introducción y objetivos 6.2. IPyhton 6.3. NumPy y SciPy 6.4. Visualizaciones gráficas con Matplotlib 6.5. Manipulación de datos con Pandas 6.6. Referencias bibliográficas 6.7. Cuaderno de ejercicios	**Clase 7 (90 min.):** • Problema: ¿Cómo se aplican las bibliotecas de Python para el análisis y visualización de datos científicos? • Resolución de la actividad 1		
Semana 8	**Tema 6**. Computación científica en Python (continuación) 6.1. Introducción y objetivos 6.2. IPyhton 6.3. NumPy y SciPy 6.4. Visualizaciones gráficas con Matplotlib 6.5. Manipulación de datos con Pandas 6.6. Referencias bibliográficas 6.7. Cuaderno de ejercicios	**Clase 8 (60 min.):** • Problema: ¿De qué manera se pueden combinar Numpy, SciPy, y Pandas para procesar y analizar un conjunto de datos desde su importación hasta la obtención de resultados interpretables? • Presentación actividad grupal	Actividad 2. Polinomios de Chebyshev y procesamiento de imágenes en Jupyter Notebooks (3.0 puntos) Test Tema 6 (0.1 puntos)	
Semana 9	**Tema 7**. Programación concurrente 7.1. Introducción y objetivos 7.2. El módulo threading 7.3. Métodos de sincronización 7.4. Comunicación entre hilos 7.5. Problemas en la programación concurrente 7.6. Ejemplos clásicos de la programación concurrente 7.7. Referencias bibliográficas 7.8. Cuaderno de ejercicios	**Clase 9 (90 min.):** • Problema: ¿Qué técnicas de programación concurrente y paralela pueden aplicarse para mejorar la eficiencia computacional? • Sesión de recomendaciones para el examen		

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 10	**Tema 7**. Programación concurrente (continuación) 7.1. Introducción y objetivos 7.2. El módulo threading 7.3. Métodos de sincronización 7.4. Comunicación entre hilos 7.5. Problemas en la programación concurrente 7.6. Ejemplos clásicos de la programación concurrente	**Clase 10 (60 min.):** • Problema: ¿Cómo determinar cuándo utilizar multiprocessing frente a threading en aplicaciones de Python, considerando las características del problema y del entorno de ejecución?	Test Tema 7 (0.1 puntos)	
Semana 11	**Tema 8**. Programación paralela y distribuida 8.1. Introducción y objetivos 8.2. Modelos de computación paralela 8.3. Evaluación del rendimiento en programas paralelos 8.4. El módulo multiprocessing 8.5. Paso de mensajes con MPI 8.6. Introducción a la programación distribuida. Llamadas a procedimientos remotos 8.7. Gestión con Celery 8.8. Referencias bibliográficas 8.9. Cuaderno de ejercicios	**Clase 11 (90 min.):** • Problema: ¿Cómo se diseñan y gestionan aplicaciones distribuidas para maximizar la eficiencia y escalabilidad? • Resolución de la actividad grupal		

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 12	**Tema 8**. Programación paralela y distribuida (continuación) 8.1. Introducción y objetivos 8.2. Modelos de computación paralela 8.3. Evaluación del rendimiento en programas paralelos 8.4. El módulo multiprocessing 8.5. Paso de mensajes con MPI 8.6. Introducción a la programación distribuida. Llamadas a procedimientos remotos 8.7. Gestión con Celery 8.8. Referencias bibliográficas 8.9. Cuaderno de ejercicios	**Clase 12 (60 min.)** • Problema: ¿Cuáles son los principales desafíos al implementar paso de mensajes y gestión de tareas con Celery en aplicaciones distribuidas y cómo se pueden superar? • Presentación de la actividad proyecto **Clase 13 (120 min.)** • Laboratorio	Actividad 3. Laboratorio: concurrencia y paralelismo en Python (5.0 puntos) Test Tema 8 (0.1 puntos)	

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 13	**Tema 9**. Programación heterogénea 9.1. Introducción y objetivos 9.2. La GPU como dispositivo de cómputo general 9.3. CUDA y PyCUDA 9.4. OpenCL y PyOpenCL 9.5. Programación GPGPU con Numba 9.6. Referencias bibliográficas 9.7. Cuaderno de ejercicios	**Clase 14 (60 min.):** • Problema: ¿Cómo se aprovechan las capacidades de las GPU para acelerar el cómputo científico?		1. Experimentación con CUDA y PyCUDA para realizar cálculos intensivos en GPU, destacando un caso específico donde se mejoró significativamente el rendimiento. 2. Comparativa de rendimiento entre código CPU tradicional y código GPU optimizado, incluyendo métricas específicas de aceleración. 3. Documentación del proceso de contenerización de una aplicación Python con Docker, destacando los desafíos y soluciones encontradas. 4. Análisis de la elección de un servicio en la nube específico para el despliegue de una aplicación, basado en características como escalabilidad, costo y facilidad de uso.
Semana 14	**Tema 9**. Programación heterogénea (continuación) 9.1. Introducción y objetivos 9.2. La GPU como dispositivo de cómputo general 9.3. CUDA y PyCUDA 9.4. OpenCL y PyOpenCL 9.5. Programación GPGPU con Numba 9.6. Referencias bibliográficas 9.7. Cuaderno de ejercicios	**Clase 15 (60 min.):** • Problema: ¿Qué consideraciones son cruciales al elegir entre CUDA, PyCUDA, y Numba para programación en GPU en proyectos de computación científica?	Test Tema 9 (0.1 puntos)	

Bloque 1. Fundamentos de programación científica	Bloque 2. Estructuras de datos y algoritmos	Bloque 3. Computación científica avanzada	Bloque 4. Computación de alto rendimiento y en la nube

	Temas	Resolución de problemas en las clases en directo	Actividades (15.0 puntos)	Experiencias en mi Porfolio
Semana 15	**Tema 10**. Computación en la nube 10.1. Introducción y objetivos 10.2. Modelos de despliegue 10.3. Categorización: IaaS, PaaS, FaaS, SaaS 10.4. Ventajas e inconvenientes de la nube	**Clase 16 (60 min.):** • Problema: ¿Qué estrategias se deben considerar para el despliegue eficiente de aplicaciones Python en la nube utilizando Docker, teniendo en cuenta los diferentes servicios en la nube disponibles? **Clase 17 (60 min.)** • Resolución de la Actividad proyecto • Sesión de repaso	Test Tema 10 (0.1 puntos)	
Semana 16	Semana de exámenes			