



Curso Académico 2022-23

PROGRAMACIÓN PARALELA

Ficha Docente

ASIGNATURA

Nombre de asignatura (Código GeA): PROGRAMACIÓN PARALELA (800619)

Créditos: 6

Créditos presenciales: 6,00

Créditos no presenciales:

Semestre: 8

PLAN/ES DONDE SE IMPARTE

Titulación: GRADO EN MATEMÁTICAS

Plan: GRADO EN MATEMÁTICAS

Curso: 4 **Ciclo:** 1

Carácter: Optativa

Duración/es: Segundo cuatrimestre (actas en Jun. y Jul.)

Idioma/s en que se imparte: Español

Módulo/Materia: CONTENIDOS AVANZADOS DE CIENCIAS DE LA COMPUTACIÓN/PARADIGMAS DE PROGRAMACIÓN

PROFESOR COORDINADOR

Nombre	Departamento	Centro	Correo electrónico	Teléfono
LLANA DIAZ, LUIS FERNANDO	Sistemas Informáticos y Computación	Facultad de Informática	llana@ucm.es	

PROFESORADO

Nombre	Departamento	Centro	Correo electrónico	Teléfono
LLANA DIAZ, LUIS FERNANDO	Sistemas Informáticos y Computación	Facultad de Informática	llana@ucm.es	

SINOPSIS

BREVE DESCRIPTOR:

El curso proporciona una introducción a la programación concurrente, distribuida y paralela. Esta programación es necesaria en entornos en los que hay múltiples procesadores interconectados entre sí ya sea de forma local o remota. El curso plantea un recorrido general y en anchura por las diversas técnicas de programación que se emplean en resolver problemas prototípicos que surgen en este tipo de sistemas: sincronización de tareas, protocolos de comunicación, paralelización de soluciones...

El curso también se centra en aprender las cuestiones prácticas necesarias de un lenguaje de programación concreto (Python) que permitan resolver y programar los algoritmos prototípicos estudiados y resolver problemas nuevos. Especial atención se dedica a los módulos de Python que permiten hacer computación en clusters utilizando sistemas de ficheros distribuidos: MapReduce y Spark.

REQUISITOS:

Es adecuado para seguir el curso con fluidez contar con los conocimientos de programación que habitualmente se consiguen con un primer curso en esta materia: variables y tipos de datos, estructuras de control, programación estructurada. . . También es recomendable cursar la asignatura programación declarativa, para tener fundamentos de programación funcional.

OBJETIVOS:

Adquirir los conocimientos elementales y la terminología adecuada que rodea al mundo de la programación paralela/concurrente/distribuida.
Entender los múltiples factores que intervienen en la definición de un problema –comunicación, topología, dependencias. . . – y cuyo estudio es fundamental para afrontar una solución.
Entender la importancia del correcto diseño –complejidad algorítmica, adecuación, escalabilidad. . . – y el estudio teórico en la resolución de problemas mediante programas.
Utilización de los módulos existentes en Python para diseñar e implementar soluciones a problemas concurrentes, paralelos y distribuidos.
Comprender el uso potencial de la programación paralela en multitud de ámbitos reales: simulación, la computación científica, gráficos y visión por ordenador, entornos colaborativos, web, bigdata. . .



Curso Académico 2022-23

PROGRAMACIÓN PARALELA

Ficha Docente

COMPETENCIAS:

Generales

Utilizar los conocimientos matemáticos para modelar y resolver problemas complejos mediante algoritmos.
Valorar diferentes soluciones alternativas y elegir de acuerdo a las situaciones específicas entre las herramientas y las técnicas adecuadas para llevar a cabo la implementación de dichas soluciones.
Saber abstraer en un problema complejo las propiedades y características esenciales reconociendo su rango de aplicabilidad y limitaciones.

Transversales:

Ser capaz de mostrar creatividad, iniciativa y espíritu emprendedor para afrontar los retos de su actividad y saber valorar las soluciones a dichos retos en el contexto industrial, económico, administrativo, medio ambiental y social.
Tener la capacidad de reunir e interpretar datos relevantes para emitir juicios que incluyan una reflexión profunda sobre temas y problemas a resolver.
Demostrar razonamiento crítico y gestionar información científica y técnica de calidad, bibliografía, bases de datos especializadas y recursos accesibles a través de Internet.

Específicas:

Diseñar programas que solucionen problemas en el ámbito de la programación concurrente/distribuida/paralela.
Utilizar herramientas informáticas para implementar y depurar programas.

Otras:

CONTENIDOS TEMÁTICOS:

- ? Conceptos Básicos y Terminología
- ? Programación concurrente
 - ? Modelo de entrelazado
 - ? Exclusión mutua
 - ? Primitivas de concurrencia
 - ? Problemas prototípicos
- ? Programación distribuida
 - ? Comunicación síncrona y asíncrona
 - ? Paso de mensajes y protocolos
 - ? Internet of Things (IoT)
 - ? Problemas prototípicos
- ? Programación paralela
 - ? Clusters, Grids, GPUs
 - ? Sistemas distribuidos de ficheros: Hadoop
 - ? Técnicas Map Reduce
 - ? Programación en Spark

ACTIVIDADES DOCENTES:

Clases teóricas:

En las que el profesor presenta los conceptos y técnicas relevantes de los contenidos y muestra las referencias bibliográficas o enlaces web a seguir para profundizar en dichos temas.

Seminarios:

Clases prácticas:

En las que en un trabajo guiado por el profesor el alumnado diseña y busca soluciones a ejercicios propuestos.

Trabajos de campo:

Prácticas clínicas:



Curso Académico 2022-23

PROGRAMACIÓN PARALELA

Ficha Docente

Laboratorios:

En las que los alumnos implementan y depuran los programas que resuelven los problemas estudiados en las clases prácticas.

Exposiciones:

Presentaciones:

En las que los alumnos, de forma individual o por grupos, a petición del profesor, preparan una exposición ante sus compañeros de una solución (bien sea diseño, implementación o ambas) a alguno de los problemas propuestos.

Otras actividades:

TOTAL:

EVALUACIÓN:

Convocatoria ordinaria:

Participación: 20%. La participación en los foros, presentación de ejercicios, la realización de los trabajos voluntarios.

Evaluación continua: 60%. Es necesario tener aptos todos los trabajos obligatorios.

Exámenes de teoría: 20%.

Convocatoria extraordinaria:

Examen: 100%

BIBLIOGRAFÍA BÁSICA:

Libros:

M. Ben-Ari, Principles of Concurrent and Distributed Programming, Prentice-Hall, 2006.

Thomas Rauber and Gudula Rünger, Parallel Programming For Multicore and Cluster Systems, Springer-Verlag, 2010.

Donald Miner y Adam Shook, MapReduce Design Patterns, O'Reilly, 2012

Timothy G. Mattson, Beverly A. Sanders and Berna L. Massingill, Patterns for Parallel Programming, Addison-Wesley, 2005.

Nancy A. Lynch, Distributed Algorithms, Morgan Kaufmann, 1996.

Karau H., Konwinski A., Wendell P., and Zaharia M. Learning Spark. O'Reilly. 2015

Internet:

Documentación de Python, <http://docs.python.org/release/2.7.1/>

Documentación de Python MapReduce <http://mrjob.readthedocs.io/en/latest/>

Documentación de Spark <https://spark.apache.org/>

OTRA INFORMACIÓN RELEVANTE