



UNIVERSIDAD
POLITÉCNICA
DE MADRID

PROCESO DE
COORDINACIÓN DE LAS
ENSEÑANZAS PR/CL/001



E.T.S. de Ingenieros
Informaticos

ANX-PR/CL/001-01

GUÍA DE APRENDIZAJE

ASIGNATURA

103000637 - Programacion Paralela y Distribuida

PLAN DE ESTUDIOS

10AN - Master Universitario En Ingenieria Informatica

CURSO ACADÉMICO Y SEMESTRE

2019/20 - Primer semestre

Índice

Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	2
4. Competencias y resultados de aprendizaje.....	3
5. Descripción de la asignatura y temario.....	4
6. Cronograma.....	6
7. Actividades y criterios de evaluación.....	8
8. Recursos didácticos.....	11

1. Datos descriptivos

1.1. Datos de la asignatura

Nombre de la asignatura	103000637 - Programacion Paralela y Distribuida
No de créditos	4.5 ECTS
Carácter	Optativa
Curso	Segundo curso
Semestre	Tercer semestre
Período de impartición	Septiembre-Enero
Idioma de impartición	Castellano
Titulación	10AN - Master Universitario En Ingenieria Informatica
Centro responsable de la titulación	10 - Escuela Tecnica Superior de Ingenieros Informaticos
Curso académico	2019-20

2. Profesorado

2.1. Profesorado implicado en la docencia

Nombre	Despacho	Correo electrónico	Horario de tutorías *
Maria De Los Santos Perez Hernandez	4204	maria.s.perez@upm.es	M - 10:30 - 12:30 X - 16:00 - 18:00 J - 16:00 - 18:00
Maria Isabel Garcia Clemente	4105	mariaisabel.garcia@upm.es	L - 11:00 - 12:00 L - 16:00 - 17:00 M - 11:00 - 12:00 X - 11:00 - 12:00 J - 16:30 - 18:30

Jose Luis Pedraza Dominguez	4105	jose Luis.pedraza@upm.es	M - 11:00 - 13:00 X - 12:00 - 14:00 J - 16:00 - 18:00
Antonio Garcia Dopico (Coordinador/a)	4202	antonio.garcia.dopico@upm. es	L - 16:00 - 17:00 M - 10:00 - 12:00 X - 11:00 - 12:00 J - 16:00 - 17:00 V - 11:00 - 12:00

* Las horas de tutoría son orientativas y pueden sufrir modificaciones. Se deberá confirmar los horarios de tutorías con el profesorado.

3. Conocimientos previos recomendados

3.1. Asignaturas previas que se recomienda haber cursado

El plan de estudios Master Universitario en Ingeniería Informática no tiene definidas asignaturas previas recomendadas para esta asignatura.

3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Conocimientos básicos de concurrencia.
- Poseer destrezas fundamentales de la programación

4. Competencias y resultados de aprendizaje

4.1. Competencias

CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

CB7 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio

CE1 - Capacidad para la integración de tecnologías, aplicaciones, servicios y sistemas propios de la Ingeniería Informática, con carácter generalista, y en contextos más amplios y multidisciplinares.

CE10 - Capacidad para comprender y poder aplicar conocimientos avanzados de computación de altas prestaciones y métodos numéricos o computacionales a problemas de ingeniería.

CE16 - Habilidad para hacer conexiones entre los deseos y necesidades del consumidor o cliente y lo que la tecnología puede ofrecer

CE4 - Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.

CG12 - Capacidad de trabajar de forma independiente en su campo profesional

CG9 - Apreciación de los límites del conocimiento actual y de la aplicación práctica de la tecnología más reciente

4.2. Resultados del aprendizaje

RA3 - Aplicar técnicas y herramientas de computación de alto rendimiento para la solución de problemas prácticos

RA109 - Conocer cómo se aplican las técnicas de computación científica en algún campo específico de ciencia o ingeniería

RA39 - Ser capaz de procesar datos masivos

RA5 - Relacionar las necesidades de los algoritmos numéricos en el modelado de problemas con su implementación práctica en hardware/software de alto rendimiento

RA97 - Conocer y saber utilizar técnicas fundamentales de computación de altas prestaciones

5. Descripción de la asignatura y temario

5.1. Descripción de la asignatura

En esta asignatura se aborda la programación de grandes máquinas orientadas a la computación de alto rendimiento (HPC: High Performance Computing). Se verá OpenMP para programar multiprocesadores de memoria compartida, MPI para programar máquinas de memoria distribuida, vectorización para explotar las unidades vectoriales de los procesadores actuales así como los Many Integrated Cores de Intel (MIC) y CUDA para programar GPUs. Tanto los MICs como las GPUs se usan a modo de coprocesadores dentro de clusters más grandes (memoria distribuida) donde cada nodo es un multiprocesador (memoria compartida)

En caso de que haya pocos alumnos (5-6 o incluso menos) se reducirán drásticamente las clases magistrales y se hará más énfasis en trabajos prácticos de los alumnos, orientando la asignatura a la realización de pequeños proyectos que puedan ayudar a los alumnos a ver las distintas partes de la asignatura como un todo, para mostrar como encajan y se complementan.

5.2. Temario de la asignatura

1. Tema 1: Introducción. Necesidad de la programación paralela
 - 1.1. 1.1 Necesidades de cómputo. Arquitecturas de altas prestaciones.
 - 1.2. 1.2 Sistemas de memoria compartida.
 - 1.3. 1.3 Sistemas de memoria distribuida, clusters y cloud.
 - 1.4. 1.4 Coprocesadores (GPUs y MIC)
 - 1.5. 1.5 Análisis de aplicaciones.
2. Tema 2: Profiling
 - 2.1. 2.1 Obtención de los tiempos de ejecución con herramientas de profiling (Valgrind, VTune)
 - 2.2. 2.2 Detección de errores y cuellos de botella (valgrind). Ejercicio práctico
3. Tema 3: Programación paralela
 - 3.1. 3.1 Programación de sistemas de memoria compartida: OpenMP. Ejercicios prácticos
4. Tema 4: Vectorización
 - 4.1. 4.1 Paralelismo de datos mediante vectorización usando OpenMP sobre Intel Xeon
5. Tema 5: GPUs
 - 5.1. 5.1 Introducción a las GPUs. Paralelismo de datos mediante el uso de GPUs como coprocesadores
 - 5.2. 5.2 CUDA: Programación de GPUs
 - 5.3. 5.3 Introducción a la arquitectura de las GPUs
 - 5.4. 5.4 Ejemplos de aplicación
6. Tema 6: Programación distribuida
 - 6.1. 6.1 Programación de sistemas de memoria distribuida: MPI. Ejercicios prácticos
7. Tema 7: E/S de Alto Rendimiento
 - 7.1. 7.1 Computación Intensiva de datos. Motivación.
 - 7.2. 7.2 Sistemas de ficheros paralelos y escalables. Lustre, GPFS, PVFS, HDFS, Ceph, GlusterFS.
 - 7.3. 7.3 Bibliotecas de E/S paralela. MPI-IO, HDF5, NetCDF
 - 7.4. 7.4 Ejemplos de aplicación

6. Cronograma

6.1. Cronograma de la asignatura *

Sem	Actividad presencial en aula	Actividad presencial en laboratorio	Otra actividad presencial	Actividades de evaluación
1	Explicación de contenidos del Tema 1 Duración: 02:00 LM: Actividad del tipo Lección Magistral Explicación de contenidos del Tema 2 Duración: 01:00 LM: Actividad del tipo Lección Magistral			
2	Explicación de contenidos del Tema 2 Duración: 01:00 LM: Actividad del tipo Lección Magistral	Ejercicio de tema 2. Valgrind Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
3	Explicación de contenidos del Tema 3 Duración: 02:00 LM: Actividad del tipo Lección Magistral	Ejercicio de tema 2. Valgrind Duración: 01:00 PL: Actividad del tipo Prácticas de Laboratorio		Entrega de la practica de profiling TI: Técnica del tipo Trabajo Individual Evaluación continua Duración: 00:00
4	Explicación de contenidos del Tema 3 Duración: 01:00 LM: Actividad del tipo Lección Magistral	Ejercicio de tema 3. OpenMP Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
5		Ejercicio de tema 3. OpenMP Duración: 03:00 PL: Actividad del tipo Prácticas de Laboratorio		Entrega de la práctica de OpenMP TI: Técnica del tipo Trabajo Individual Evaluación continua Duración: 00:00
6	Explicación de contenidos del Tema 4 Duración: 03:00 LM: Actividad del tipo Lección Magistral			
7		Ejercicio de tema 4 vectorización Duración: 03:00 PL: Actividad del tipo Prácticas de Laboratorio		Entrega de la práctica de Vectorización TI: Técnica del tipo Trabajo Individual Evaluación continua Duración: 00:00
8	Explicación de contenidos del Tema 5 Duración: 03:00 LM: Actividad del tipo Lección Magistral			
9		Ejercicios de tema 5 GPUs Duración: 03:00 PL: Actividad del tipo Prácticas de Laboratorio		
10		Ejercicios de tema 5 GPUs Duración: 03:00 PL: Actividad del tipo Prácticas de Laboratorio		

11	Explicación de contenidos del Tema 6 Duración: 03:00 LM: Actividad del tipo Lección Magistral			Entrega de la práctica de GPUs TI: Técnica del tipo Trabajo Individual Evaluación continua Duración: 00:00
12		Ejercicio de tema 6. MPI Duración: 03:00 PL: Actividad del tipo Prácticas de Laboratorio		
13	Explicación de contenidos del Tema 7 Duración: 03:00 LM: Actividad del tipo Lección Magistral			Entrega de la práctica de MPI TI: Técnica del tipo Trabajo Individual Evaluación continua Duración: 00:00
14		Ejercicios de tema 7 MPI-IO Duración: 03:00 PL: Actividad del tipo Prácticas de Laboratorio		
15			Tutorías de Proyectos Duración: 03:00 OT: Otras actividades formativas	Entrega de la practica de MPI-IO PI: Técnica del tipo Presentación Individual Evaluación continua Duración: 00:00
16			Tutorías y entregas de Proyectos Duración: 03:00 OT: Otras actividades formativas	Entrega de los proyectos TI: Técnica del tipo Trabajo Individual Evaluación continua Duración: 00:00
17				Los que no hayan aprobado por curso tendrá un plazo extraordinario para volver a entregar los ejercicios prácticos que no ha superado satisfactoriamente TI: Técnica del tipo Trabajo Individual Evaluación sólo prueba final Duración: 01:00

Las horas de actividades formativas no presenciales son aquellas que el estudiante debe dedicar al estudio o al trabajo personal.

Para el cálculo de los valores totales, se estima que por cada crédito ECTS el alumno dedicará dependiendo del plan de estudios, entre 26 y 27 horas de trabajo presencial y no presencial.

* El cronograma sigue una planificación teórica de la asignatura y puede sufrir modificaciones durante el curso.

7. Actividades y criterios de evaluación

7.1. Actividades de evaluación de la asignatura

7.1.1. Evaluación continua

Sem.	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
3	Entrega de la practica de profiling	TI: Técnica del tipo Trabajo Individual	No Presencial	00:00	4%	2 / 10	
5	Entrega de la práctica de OpenMP	TI: Técnica del tipo Trabajo Individual	No Presencial	00:00	4%	2 / 10	CG9 CG12 CE1
7	Entrega de la práctica de Vectorización	TI: Técnica del tipo Trabajo Individual	No Presencial	00:00	4%	2 / 10	CE10 CG12 CE1
11	Entrega de la práctica de GPUs	TI: Técnica del tipo Trabajo Individual	No Presencial	00:00	10%	2 / 10	CG9 CB7 CB10 CE10 CE16 CG12 CE1
13	Entrega de la práctica de MPI	TI: Técnica del tipo Trabajo Individual	No Presencial	00:00	4%	2 / 10	CB7 CB10 CG12 CE1
15	Entrega de la practica de MPI-IO	PI: Técnica del tipo Presentación Individual	No Presencial	00:00	4%	2 / 10	
16	Entrega de los proyectos	TI: Técnica del tipo Trabajo Individual	No Presencial	00:00	70%	2 / 10	CG9 CB7 CB10 CE10 CE16 CG12 CE1 CE4

7.1.2. Evaluación sólo prueba final

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Los que no hayan aprobado por curso tendrá un plazo extraordinario para volver a entregar los ejercicios prácticos que no ha superado satisfactoriamente	TI: Técnica del tipo Trabajo Individual	No Presencial	01:00	100%	5 / 10	CG9 CB7 CB10 CE10 CE16 CG12 CE1 CE4

7.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
El alumno tendrá un plazo extraordinario para volver a entregar los ejercicios prácticos que no ha superado satisfactoriamente. En los criterios de evaluación se amplía la información	TI: Técnica del tipo Trabajo Individual	Presencial	00:00	100%	5 / 10	

7.2. Criterios de evaluación

Sistema de evaluación

La asignatura se evaluará mediante la realización de una serie de proyectos y ejercicios prácticos. El enunciado de los mismos se presentará en clase, en la propia aula y durante el horario regular de la asignatura, en las fechas especificadas en el calendario de la asignatura. El desarrollo de estos proyectos se llevará a cabo tanto de forma presencial en las clases prácticas en el laboratorio, como no presencial usando los recursos ofrecidos por el Centro de Cálculo para tal fin, y apoyándose en las tutorías para la resolución de cualquier aspecto vinculado con el desarrollo de los mismos. Los plazos de entrega se publicarán conjuntamente con el enunciado con suficiente antelación.

La nota final de la asignatura se calculará considerando un peso de 30% para las prácticas presenciales en aula informática, y un 70% para los proyectos. Para aprobar la asignatura, además de tener una nota final mayor o igual a 5, habrá que tener una nota mínima de 2 en cada una de las cuatro partes.

Evaluación en la convocatoria extraordinaria de julio

En caso de que el alumno no haya superado la asignatura deberá presentarse al examen de esta convocatoria. El alumno tendrá un plazo extraordinario para volver a entregar los ejercicios prácticos que no ha superado satisfactoriamente, plazo que finaliza el día anterior al de la celebración del examen.

Plazos de entrega

Las fechas exactas establecidas como plazo máximo para la entrega de las distintas prácticas serán publicadas en el enunciado de cada práctica. No se admitirán prácticas entregadas fuera de plazo.

8. Recursos didácticos

8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Hennessy, J. L, Patterson, D. A.; "Computer Architecture: A quantitative Approach", 5th. Ed.. Morgan Kauffmann Pub., 2012	Bibliografía	Libro
Overview Of The MPI-IO Parallel I/O Interface (1995) Peter Corbett , Dror Feitelson , Sam Fineberg , Yarsun Hsu , Bill Nitzberg , Jean-Pierre Prost , Marc Snir , Bernard Traversat , Parkson Wong	Bibliografía	Libro
Using OpenMP. Chapman B. et al. MIT Press 2008	Bibliografía	Libro
Using MPI. Gropp W. et al. MIT Press. 1999	Bibliografía	Libro
Patterns for parallel programming. Mattson T.G et al. Addison-Wesley. 2005	Bibliografía	Libro
Programming Massively parallel processors. Kirk D.B. et al. Morgan Kaufman. 2010	Bibliografía	Libro
https://software.intel.com/sites/default/files/m/4/8/8/2/a/31848-CompilerAutovectorizationGuide.pdf	Recursos web	Tutorial
The software vectorization handbook. Bik, Aart J.C. Intel Press 2004	Bibliografía	Libro