



GUÍA DOCENTE

PROGRAMACIÓN DE BAJO NIVEL GRADO EN INGENIERÍA DEL SOFTWARE

MODALIDAD: A DISTANCIA

CURSO ACADÉMICO: 2023-2024

Denominación de la asignatura:	Programación de bajo nivel
Titulación:	Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Optatividad
Curso:	4º
Cuatrimestre:	2
Carácter:	OP
Créditos ECTS:	3
Modalidad de enseñanza:	A distancia
Idioma:	Castellano
Profesor / Email:	Marcos Novalbos Mendiguchía / marcos.novalbos@u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

Esta materia recoge algunos contenidos avanzados y/o especializados que pudiera requerir un ingeniero del software generalista

Descripción de la asignatura

Esta asignatura es relevante para adquirir conocimientos de programación orientada a hardware específico, resaltando la importancia de la eficiencia de código, el acceso directo a memoria y la gestión de recursos

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

Competencias (genéricas, específicas y transversales)

COMPETENCIAS BÁSICAS Y GENERALES

CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en

libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

CB2: Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB3: Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

CG1: Entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas.

CG2: Desarrollar soluciones informáticas que sean respetuosas con el medio ambiente, los deberes sociales y recursos naturales, además de cumplir con la legislación y ética.

CG3: Aplicar los fundamentos científicos para la resolución de problemas informáticos

CG4: Entender la complejidad, simplificar y optimizar los sistemas informáticos

CG6: Trabajar en entornos de trabajo multidisciplinares demostrando capacidad de trabajo en equipo, versatilidad, flexibilidad, creatividad y respeto por el trabajo de los compañeros de otras áreas.

CG7: Aplicar los fundamentos creativos de generación de ideas en los proyectos de desarrollo software para entornos digitales.

CG9: Aprender, modificar y producir nuevas tecnologías informáticas

CG10: Aplicar las técnicas creativas para la realización de proyectos informáticos

CG11: Buscar, analizar y gestionar la información para poder extraer conocimiento de la misma.

COMPETENCIAS ESPECÍFICAS

CE10: Generar documentación de una aplicación de forma automática así como entender y manejar adecuadamente un gestor de versiones de código las que utilizan principios de ingeniería inversa.

CE15: Desarrollar aplicaciones distribuidas teniendo en cuenta la tolerancia de los fallos, la adaptabilidad, el balance de carga y la predictividad del sistema.

CE17: Desarrollar aplicaciones que utilicen las características de paralelización de tarjetas gráficas y arquitecturas de altas prestaciones.

CE20: Testar en profundidad el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas.

CE21: Evaluar la calidad de una aplicación informática desde el punto de vista de su diseño e implementación, aplicando métricas, procedimientos y estándares de medición de calidad del software.

COMPETENCIAS TRANSVERSALES

CT1: Conocer la definición y el alcance, así como poner en práctica los fundamentos de las metodologías de gestión de proyectos de desarrollo tecnológico.

CT2: Conocer los principales agentes del sector y el ciclo de vida completo de un proyecto en desarrollo y comercialización de contenidos digitales

CT4: Actualizar el conocimiento adquirido en el manejo de herramientas y tecnologías digitales en función del estado actual del sector y de las tecnologías empleadas.

CT5: Poseer las habilidades necesarias para el emprendimiento digital.

Resultados de aprendizaje

Al acabar la titulación, el graduado o graduada será capaz de:

- Entender el ciclo de aseguramiento de la calidad del software
- Diseñar un plan de prueba de software
- Conocer los entornos de prueba más habituales de la industria
- Desarrollar una aplicación intensiva en el uso de GPU
- Ser capaz de medir rendimiento en aplicaciones distribuidas.

CONTENIDO

Programación concurrente sobre la Unidad de Procesamiento Gráfico

Profiling de sistemas distribuidos

TEMARIO

Tema 1: Introducción a lenguajes de bajo nivel

- Repaso C/C++
- Repaso a arquitecturas HW

Tema 2A: Uso de ficheros "binarios"

- Implementación de librería de lectura para un formato dado

Tema 2B: Juegos de instrucciones vectorial

- Uso de juegos de instrucciones vectorial para X86_64: AVX/SSE

Tema 3: Coprocesadores

- Introducción a paralelismo
- Uso de GPUs como coprocesador: NVidia CUDA

Tema 4: LLlamadas a sistema y gestión de memoria

- Implementación de un gestor de memoria tipo malloc

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DE APRENDIZAJE

Actividades formativas

Actividad Formativa	Horas totales	Horas síncronas
<i>Sesiones teóricas virtuales síncronas</i>	3,00	3
<i>Sesiones teóricas virtuales asíncronas</i>	11,00	0
<i>Sesiones prácticas virtuales síncronas</i>	2,00	2
<i>Sesiones prácticas virtuales asíncronas</i>	3,00	0
<i>Debate y discusión oral y/o escrita.</i>	5,00	0
<i>Tutorías</i>	2,00	2
<i>Estudio independiente y trabajo autónomo del alumno</i>	25,00	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	19,00	0
<i>Actividades de Evaluación</i>	3,00	3
<i>Test de autoevaluación</i>	2,00	0
<i>Prácticas externas</i>	0,00	0
<i>Preparación y defensa virtual del TFG</i>	0,00	0
<i>Seguimiento de proyectos</i>	0,00	0
TOTAL	75	10

Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo
Método expositivo o lección magistral
Método del caso
Aprendizaje basado en la resolución de problemas
Aprendizaje cooperativo o colaborativo
Aprendizaje por indagación
Metodología flipped classroom o aula invertida
Gamificación

DESARROLLO TEMPORAL

Presentación - semana 1
Unidad 1 - semana 2-5
Unidad 2 - semana 6-9
Unidad 3 - semana 10-12
Repaso - semana 13-14
Evaluación - semana 15

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	0	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	0	70
<i>Prueba Objetiva</i>	0	50

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
-------------------------	------------------------	-----------------------------

<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	20	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	20	20
<i>Prueba Objetiva</i>	60	70

Consideraciones específicas acerca de la evaluación

Será necesario que obtener una nota mínima de 4 puntos (sobre 10) en la prueba final presencial para que se realice la media con las actividades formativas.

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía Básica:

- ADDISON WESLEY Ed 1(2010); CUDA by Example

Bibliografía Recomendada:

- PRENTICE HALL Ed. 0006 (2010) Assembly Language for X86

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Materiales:

Ordenadores con sistema operativo Linux y tarjetas gráficas NVidia serie GTX400 o superior.

No valen Máquinas virtuales, es necesario un Linux “nativo” para poder realizar todas las prácticas

Software:

Sistema Operativo Linux