GUÍA DOCENTE

PROGRAMACIÓN DE SISTEMAS DISTRIBUIDOS

GRADO EN INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2025-2026

Denominación de la asignatura:	Programación de sistemas distribuidos
Titulación:	Grado en Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Programación
Curso:	3º
Cuatrimestre:	1
Carácter:	OB
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Marcos Novalbos Mendiguchía / Marcos.novalbos@u-tad.com German Horacio Viscuso / german.viscuso@live.u-tad.com Ivan Garcia Gatuti / ivan.gatuti@ext.live.u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

Esta asignatura pertenece a la materia de programación. Esta materia se dedica al estudio de las técnicas y los lenguajes de programación en los que se fundamentarán los estudios del grado de ingeniería del software.

Descripción de la asignatura

Esta asignatura pretende enseñar los principios computacionales que permiten implementar los sistemas distribuidos, para más adelante implementar aplicaciones que se adapten lo mejor posible. Se plantea como un laboratorio donde los alumnos programarán aplicaciones distribuidas sobre dos entornos:

- Sistemas tipo cluster, programación paralela orientada a procesos
- Sistemas en la nube, programación paralela orientada a servicios

Se pedirá a los alumnos que realicen configuraciones sobre una red de ordenadores que será accedida de forma remota en la medida de lo posible. Sobre esa red de ordenadores se implementarán aplicaciones que responderán a las peticiones de usuarios remotos, que cubrirán distintos servicios.

En la medida de lo posible, se usarán los servicios que provee Amazon AWS. En concreto, se podrán al menos usar los servicios EC2, Lambda, RDS, E3. Adicionalmente, se anima a los alumnos a explorar otros servicios disponibles en ese sistema, en caso de que se puedan aprovechar para sus aplicaciones

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE DE LA MATERIA

Competencias (genéricas, específicas y transversales)

COMPTETENCIAS BÁSICAS Y GENERALES:

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas

CG3 - Conocimiento de los fundamentos científicos aplicables a la resolución de problemas informáticos

CG4 - Capacidad para simplificar y optimizar los sistemas informáticos atendiendo a la comprensión de su complejida

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

CG10 - Uso de técnicas creativas para la realización de proyectos informáticos

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

COMPETENCIAS ESPECÍFICAS

CE1 - Conocimiento de la estructura de las computadoras, de los conceptos de codificación, manipulación, tratamiento de la información y uso de lenguajes de bajo nivel

CE7 - Conocimiento de los principales tipos de estructuras de datos y utilización de las librerías y de los técnicas algorítmicas asociadas a dichas estructuras junto con los órdenes de complejidad que caracterizan a dichas técnicas

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación

CE9 - Conocimiento de las estructuras de control, variables, sintaxis de programación y gestión del uso de la memoria de manera eficaz en el desarrollo de una aplicación informática

CE10 - Capacidad para manejar un gestor de versiones de código y generar la documentación de una aplicación de forma automática.

CE15 - Conocimiento de la tolerancia a los fallos, la adaptabilidad, el balance de carga y la predictividad del sistema para el desarrollo de aplicaciones distribuidas

CE17 - Conocimiento de las características de paralelización de tarjetas gráficas y de arquitecturas de altas prestaciones para el desarrollo de aplicaciones.

CE20 - Capacidad para testar el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas

CE23 - Conocimiento de los principios de la inteligencia artificial y uso de algoritmos de búsqueda deterministas y máquinas de estado

Resultados de aprendizaje

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

- Entender y manejar el concepto de memoria dinámica

- Identificar clases de objetos con los datos de un problema.

- Crear clases y objetos y manipularlos.

- Entender y utilizar los mecanismos de herencia, polimorfismo y sobrecarga de operadores.

- Identificar las relaciones entre clases en distintos casos de uso.

- Dominar un lenguaje de programación orientado a objetos.

- Dominar los patrones de programación

- Conocer las distintas formas de resolución de problemas desde el punto de vista de la algoritmia, como, por ejemplo, el esquema divide y vencerás,

- programación dinámica, backtracking o algoritmos genéticos.

- Estudiar la complejidad de un determinado algoritmo, interpretar dicha complejidad y analizar posibles optimizaciones.

- Codificar un programa que sea capaz de encontrar el camino óptimo que une dos nodos de un grafo siguiendo los distintos algoritmos de pathfinding.

- Crear y entrenar redes neuronales que solucionen problemas concretos.

CONTENIDO

Arquitectura de aplicaciones distribuidas

Algoritmos distribuidos

Tolerancia a fallos

Adaptabilidad y balance de cargas

TEMARIO

Tema 1. Introducción a los sistemas distribuidos

Historia e inicios

Ventajas de su uso

Introducción general a sistemas clúster, grid y cloud

Tema 2. Sistemas de memoria distribuida y Clúster

Computación Cliente/Servidor

Llamadas y a procedimientos remotos

Paso de mensajes

RPC

Clústers

Gestión de procesos distribuidos

Migración de procesos

Exclusión mutua e interbloqueo

Programación en sistemas tipo clúster (Tema práctico)

Computación paralela

Metodología de programación paralela

Virtualización

Paradigma de paso de mensajes

Herramientas:

Virtualbox, Docker, kubernetes, AWS/EC2

Prácticas

Tema 3. Programación orientada a servicios en cloud

Introducción

Fundamentos del Cloud

Arquitectura Cloud

Ejemplo de un Cloud:

Amazon Web Services

EC2

S3

Lambda

RDS

Prácticas:

Programación usando servicios AWS

Implementación de servicios

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
Clases teóricas / Expositivas	36	36
Clases Prácticas	19	19
Tutorías	4	2
Estudio independiente y trabajo autónomo del alumno	52	0
Elaboración de trabajos (en grupo o individuales)	34	0
Actividades de Evaluación	6	6

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

UNIDADES DIDÁCTICAS / TEMAS PERÍODO TEMPORAL

Tema 1. Introducción a los sistemas distribuidos Semana 1

Tema 2 (primera parte). Sistemas de memoria distribuida y Cluster Semana 2

Introducción. Programación basada en paso de mensajes Semana 3 y 4

Práctica 1: Configuración de un cluster virtual (Red EC2+Aplicación distribuida) Semana 5 y 6

Tema 2 (segunda parte). Sistemas de memoria distribuida y Cluster Semana 7-8

Práctica 2: Programación de procesos remotos (Kubernetes+Docker) Semanas 9-10

Tema 3. Grid y Cloud computing Semana 11-12

Práctica 3: Programación orientada a servicios (AWS/Cloud) Semana 13-15

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	0	30
Evaluación de trabajos, proyectos, informes, memorias	30	80
Prueba Objetiva	10	60

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	10	0
Evaluación de trabajos, proyectos, informes, memorias	70	80
Prueba Objetiva	20	20

Consideraciones generales acerca de la evaluación

Evaluación Ordinaria

• La evaluación de la participación en clase, en prácticas o en proyectos de la asignatura se realizará a partir de la asistencia y la participación activa en clase y en el resto de las actividades desarrolladas durante el curso. Es obligatorio tener una asistencia de al menos el 70% para poder optar al 10% de participación (la asistencia no se evalúa, pero es necesario un mínimo). A lo largo de la convocatoria se pedirán entregas de ejercicios específicos para este aspecto, representando el 10% de la calificación final de la asignatura en la convocatoria ordinaria (obligatorio para optar a matrícula).

• Para optar al 10% de asistencia y participación se seguirá el siguiente criterio:

Asistencia a al menos el 70% de las sesiones: El alumno deberá estar presente cuando se pase lista al inicio de la clase. En caso de no estar presente en ese momento y no justificar debidamente su ausencia, ese día contará como una falta no justificada, restando puntos a la nota final

Se tendrá en cuenta la entrega en tiempo y forma de los ejercicios pedidos a lo largo de la asignatura para esta sección del 10%: Se espera una participación activa/constante a lo largo del curso. En caso de no entregar cualquier ejercicio/actividad pedida a tiempo, se perderá un porcentaje anunciado por el profesor en la parte de participación.

• A lo largo del curso se plantearán prácticas OBLIGATORIAS que deberán ser entregadas antes de la fecha indicada a través de la plataforma virtual. Estos trabajos conllevarán una prueba/examen para demostrar el conocimiento adquirido y la notas de las pruebas supondrá un 70% (ordinaria) de la calificación final de la asignatura en la convocatoria ordinaria.

• Todas las pruebas/exámenes prácticos son obligatorios, deberán ser evaluados con un mínimo del 50% de la nota máxima para poder aprobar. Aquellos alumnos que no superen esa nota o que decida

descartarla voluntariamente, deberán realizar una prueba/examen complementaria con en evaluación extraordinaria. RESUMIENDO: Se proponen prácticas para que el alumno las realice, estudie y entregue antes del examen de prácticas, y la nota final de la misma será la nota del examen.

• Para aprobar la asignatura en la convocatoria ordinaria, es imprescindible que la nota del examen/prueba objetiva sea al menos 5.0 (sobre 10), y que todas las prácticas tengan una nota mínima de 4.0 cada una de ellas por separado, dando como media de prácticas un mínimo de 5.0. En resumen, se debe tener una nota mínima de 5.0 en media de prácticas y un 5.0 en examen teórico para poder aprobar, se hace media si algún examen de prácticas tiene una nota mínima de 4.0. En caso de no cumplirse alguno de estos requisitos, la asignatura se considerará automáticamente suspensa independientemente del resto de calificaciones.

Evaluación Extraordinaria

• En caso de no conseguir el aprobado en la convocatoria ordinaria de enero, el alumno podrá presentarse a la convocatoria extraordinaria de julio. Se guardarán las notas de las entregas de prácticas y exámenes realizados hasta ese momento. El cálculo de la nota se realizará de la misma manera que en primera convocatoria, teniendo en cuenta las nuevas ponderaciones (80% prácticas, 20% Prueba Objetiva/Examen teórico).

• La evaluación extraordinaria se aprobará realizando un examen que contenga las partes suspensas en ordinaria:

Pruebas/Exámenes prácticos

Examen teórico

Se podrá aprobar si la media de los ejercicios/pruebas entregados en el examen es de 5.0

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía Básica:

• Burns, Brendan. Designing Distributed Systems: Patterns and Paradigms forScalable, Reliable Services. " O'Reilly Media, Inc.", 2018.

• Wittig, Michael, Andreas Wittig, and Ben Whaley. Amazon web services inaction. Manning, 2016.

• Golden, Bernard. Amazon web services for dummies. John Wiley & Sons, 2013.

Bibliografía Recomendada:

• MySQL Manuals https://dev.mysql.com/doc/

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula
Aula teórica

Equipo de proyección y pizarra

Materiales:
Ordenador personal con Windows, Linux u OSX (al menos 8Gb ram y 50GB de HDDlibre)

Software:
Recomendado, instalación física de Linux (No virtualizada):

- Ubuntu 20.04 como mínimo (.iso para instalar desde VirtualBox)

- QTCreator u otro IDE de programación

- Librerías gcc, g++

- Acceso a cuenta AWS (preferible AWS Educate)