



UNIVERSIDAD
POLITÉCNICA
DE MADRID

PROCESO DE
COORDINACIÓN DE LAS
ENSEÑANZAS PR/CL/001



E.T.S. de Ingenieros
Informáticos

ANX-PR/CL/001-01

GUÍA DE APRENDIZAJE

ASIGNATURA

103000751 - Desarrollo De Aplicaciones Distribuidas En Tiempo Real

PLAN DE ESTUDIOS

10AK - Master Universitario En Software Y Sistemas

CURSO ACADÉMICO Y SEMESTRE

2024/25 - Primer semestre

Índice

Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	2
4. Competencias y resultados de aprendizaje.....	2
5. Descripción de la asignatura y temario.....	4
6. Cronograma.....	8
7. Actividades y criterios de evaluación.....	10
8. Recursos didácticos.....	14
9. Adendas.....	15

1. Datos descriptivos

1.1. Datos de la asignatura

Nombre de la asignatura	103000751 - Desarrollo de Aplicaciones Distribuidas en Tiempo Real
No de créditos	4 ECTS
Carácter	Optativa
Curso	Primer curso
Semestre	Primer semestre
Período de impartición	Septiembre-Enero
Idioma de impartición	Castellano
Titulación	10AK - Master Universitario en Software y Sistemas
Centro responsable de la titulación	10 - E.T.S. De Ingenieros Informáticos
Curso académico	2024-25

2. Profesorado

2.1. Profesorado implicado en la docencia

Nombre	Despacho	Correo electrónico	Horario de tutorías *
Santiago Tapia Fernandez (Coordinador/a)		santiago.tapia@upm.es	- -
Juan Pedro Brito Mendez		juanpedro.brito@upm.es	Sin horario.

* 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 Software y Sistemas no tiene definidas asignaturas previas recomendadas para esta asignatura.

3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Programación, especialmente Programación Orientada a Objetos

4. Competencias y resultados de aprendizaje

4.1. Competencias

CG12 - Comprensión amplia de las técnicas y métodos aplicables en una especialización concreta, así como de sus límites

CG13 - Apreciación de los límites del conocimiento actual y de la aplicación práctica de la tecnología más reciente.

CG4 - 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.

CG7 - Especificación y realización de tareas informáticas complejas, poco definidas o no familiares

CG8 - Planteamiento y resolución de problemas también en áreas nuevas y emergentes de su disciplina

CG9 - Aplicación de los métodos de resolución de problemas más recientes o innovadores y que puedan implicar el uso de otras disciplinas

CGI20 - Adquirir conocimientos científicos avanzados del campo de la informática que le permitan generar nuevas ideas dentro de una línea de investigación.

CGI23 - Capacidad de leer y comprender publicaciones dentro de su ámbito de estudio/investigación, así como su catalogación y valor científico

4.2. Resultados del aprendizaje

RA41 - Aplicar efectivamente las técnicas de verificación y validación de software

RA60 - Ser capaz de entender artículos de investigación en el área de sistemas distribuidos

RA88 - Evaluar si un programa se ha desarrollado adecuadamente siguiendo un paradigma de programación atendiendo al enunciado del problema

RA52 - Ser capaz de relacionar una línea de investigación emergente en sistemas distribuidos con los fundamentos de los sistemas distribuidos

RA15 - Knowledge of techniques for proving code correctness

RA49 - Conocer principales líneas activas en sistemas distribuidos

RA36 - Familiaridad con la idea de requisito de diseño y de implementación

RA50 - Ser capaz de encontrar e identificar artículos seminales de un área de investigación en sistemas distribuidos

RA51 - Ser capaz de estudiar y analizar un nuevo área de investigación en sistemas distribuidos

RA53 - Ser capaz de identificar los principales retos que aborda una línea de investigación emergente en sistemas distribuidos

RA87 - Modelar un programa con la estructura de un paradigma de programación a partir del enunciado de un problema

RA14 - Familiarity with the idea of design and implementation requirement

5. Descripción de la asignatura y temario

5.1. Descripción de la asignatura

Resumen

Una aplicación distribuida es un sistema formado por uno o varios programas que se ejecutan en varios dispositivos conectados a una red de comunicaciones. Si, a su vez, se requiere que sea de tiempo real (no estricto, es decir, sin deadlines críticas) es necesario que los tiempos de comunicación y ejecución en cada programa estén acotados y sean previsibles. Se excluyen de esta asignatura los sistemas hard real time, es decir, aquellos sistemas donde no cumplir una restricción de tiempo implica un fallo crítico.

Cumplir la restricción de tiempo real es más sencillo si se utiliza un lenguaje de programación cercanos a la máquina, más específicamente, el uso de C/C++ resulta una elección casi obligada dado que son lenguajes compilados y se ejecutan en nativo (directamente sobre el microprocesador físico). Entre estos dos lenguajes se va a escoger C++ como prioritario dado que es un lenguaje orientado a objetos y resulta más adecuado para desarrollar sistemas potencialmente muy complejos.

El desarrollo de software en C++ y, especialmente, cuando el software tiene que cumplir criterios de eficiencia y previsibilidad en tiempos de ejecución, exige dominar el propio lenguaje y las últimas herramientas que se utilizan el desarrollo. Como parte de la asignatura se estudiará: la sintaxis del lenguaje, el diseño orientado a objetos, la gestión de compilación de un proyecto software, la automatización de pruebas, el control de versiones, la arquitectura de software (programas, librerías dinámicas, plugins, ?), etc. Todo ello con el objetivo de producir un software eficaz, previsible y fiable.

La restricción de tiempo real también genera restricciones respecto de las comunicaciones, de nuevo, no es recomendable el uso de middlewares no desarrollados específicamente para su uso en tiempo real y que impliquen elevar los tiempos (o hacerlos imprevisibles). En este sentido se estudiarán varias alternativas de middleware según sus características y funcionalidades.

Objetivos

- * Conocer la metodología, técnicas y herramientas del desarrollo de software en C++, especialmente, aquellas que proporcionan eficacia, previsibilidad y fiabilidad.
- * Aplicar el conocimiento anterior a casos simples de desarrollo.
- * Conocer los conceptos claves respecto del desarrollado de aplicaciones distribuidas en tiempo real:

sincronización, temporización, eventos y su gestión, patrones de diseño de comunicaciones, etc.

- * Aplicar dichos conocimientos al desarrollo de aplicaciones.
- * Conocer los desafíos de investigación en el área de las aplicaciones distribuidas en tiempo real.
- * Aplicar dichos conocimientos al planteamiento de una línea de investigación en este área.

Temario

La asignatura consta de un bloque de introducción para establecer los objetivos y características generales de las aplicaciones distribuidas en tiempo real y dos bloques que desarrollan cada una de las características: el tiempo real y la ejecución distribuida.

Aunque puede ser posible desarrollar software para tiempo real en cualquier lenguaje, las características de C++ respecto de su rendimiento y de su capacidad para definir cómo se gestiona la memoria lo convierten en una opción muy sólida para este tipo de aplicaciones. En el segundo bloque de la asignatura se abordan todos los aspectos de la programación en C++, desde la propia sintaxis del lenguaje hasta el diseño, pasando por las herramientas de gestión de versiones o el desarrollo de pruebas automáticas. Todos estos temas se van a desarrollar a través de casos prácticos, los alumnos tendrán que desarrollar sus propias implementaciones, en cada caso se introducirán nuevos conceptos de programación y herramientas hasta cubrir todos los ítems del temario. Dentro de este bloque los temas 2.5 y 2.7 corresponden específicamente con aspectos directamente relacionados con las aplicaciones en tiempo real, el resto de temas del curso son generales para cualquier curso de programación, pero en ellos se resaltarán aquellas características de los programas que permiten su uso en tiempo real. En el curso, en general, no se abordará el tema del denominado tiempo real duro (Hard Real Time).

En el tercer bloque se aborda las comunicaciones necesarias para distribuir la ejecución de la aplicación. En primer lugar se van a estudiar de forma general las arquitecturas y diseños más comunes para implementar aplicaciones distribuidas para, a continuación, estudiar casos concretos de middleware y su programación en C++. Igual que en el bloque anterior se van a proponer casos prácticos que los alumnos deben resolver usando alguna de las alternativas de middleware propuestas. Como sección destacable se incluye el uso de un sistema de logging para la monitorización y la evaluación de las aplicaciones, así como el análisis empírico de las aplicaciones.

Metodología

En general, la metodología docente será "Aprendizaje Basado en Retos", si bien normalmente este tipo de metodología suele aplicarse usando retos dilatados en el tiempo para esta asignatura se trabajará sobre nano o

mini retos, su duración será entre 1 y 3 semanas máximo. Se trabajará en equipo para resolver los retos planteados. Se evaluará la participación activa en los trabajos en equipo.

En cada clase presencial:

1. El profesor presentará brevemente los conceptos teóricos relacionados con el siguiente reto a resolver.
2. El profesor presentará el planteamiento del reto.
3. Cuando los alumnos terminen un reto, presentarán la solución y se revisará por parte del propio profesor y de resto los alumnos.

Dado que el proceso de revisión de las soluciones es parte fundamental del aprendizaje del alumno, **la asistencia a las clases es imprescindible**. La evaluación tendrá en cuenta tanto el trabajo realizado para completar los retos como la participación en el proceso de revisión de las soluciones.

5.2. Temario de la asignatura

1. Introducción a las Aplicaciones Distribuidas en Tiempo Real
2. Desarrollo de Software para Tiempo Real
 - 2.1. Programación en C++
 - 2.2. Herramientas para el desarrollo de software
 - 2.2.1. Gestión de versiones
 - 2.2.2. Construcción del software y optimización
 - 2.2.3. Despliegue
 - 2.3. Diseño Orientado a Objetos
 - 2.4. Aplicabilidad de la restricción de Tiempo Real
 - 2.5. Pruebas automáticas y Análisis de la Eficiencia
 - 2.6. Módulos software: librerías y plugins
 - 2.7. Eventos: métodos sincronos y asincronos, y callbacks
3. Middleware para Aplicaciones Distribuidas
 - 3.1. Arquitectura y patrones de diseño para comunicaciones
 - 3.2. Data marshalling

3.3. Tecnologías de Middleware para Tiempo Real

3.4. Programación de aplicaciones distribuidas en C++

3.5. Monitorización y depuración: sistemas de logging

3.6. Calidad de servicio: latencia, fiabilidad y escalabilidad en sistemas distribuidos

6. Cronograma

6.1. Cronograma de la asignatura *

Sem	Actividad tipo 1	Actividad tipo 2	Tele-enseñanza	Actividades de evaluación
1	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			
2	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			
3	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00
4	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00
5	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00
6	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00
7	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00
8	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00
9	Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos			Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00

10	<p>Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos</p>			<p>Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00</p>
11	<p>Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos</p>			<p>Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00</p>
12	<p>Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos</p>			<p>Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00</p>
13	<p>Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos</p>			<p>Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00</p>
14	<p>Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos</p>			<p>Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00</p>
15	<p>Presentación de los retos y su revisión Duración: 02:00 AR: Aprendizaje basado en retos</p> <p>Examen Teórico - Práctico Duración: 02:00 OT: Otras actividades formativas / Evaluación</p>			<p>Presentación de Soluciones PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 01:00</p> <p>Examen Teórico - Práctico EX: Técnica del tipo Examen Escrito Evaluación Progresiva Presencial Duración: 02:00</p>
16				
17				<p>Examen Teórico - Práctico EX: Técnica del tipo Examen Escrito Evaluación Global Presencial Duración: 02:00</p> <p>Resolución de un reto y su presentación PG: Técnica del tipo Presentación en Grupo Evaluación Global No presencial Duración: 04:00</p>

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.

7. Actividades y criterios de evaluación

7.1. Actividades de evaluación de la asignatura

7.1.1. Evaluación (progresiva)

Sem.	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
3	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
4	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
5	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
6	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
7	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13

							CGI23
8	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
9	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
10	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
11	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
12	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
13	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23

14	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
15	Presentación de Soluciones	PG: Técnica del tipo Presentación en Grupo	Presencial	01:00	5%	0 / 10	CG9 CGI20 CG12 CG4 CG7 CG8 CG13 CGI23
15	Examen Teórico - Práctico	EX: Técnica del tipo Examen Escrito	Presencial	02:00	35%	3 / 10	

7.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Examen Teórico - Práctico	EX: Técnica del tipo Examen Escrito	Presencial	02:00	35%	3 / 10	CGI20 CG12 CG4 CGI23
17	Resolución de un reto y su presentación	PG: Técnica del tipo Presentación en Grupo	No Presencial	04:00	65%	0 / 10	CG12 CG4 CG7 CG8 CG13 CGI23 CG9 CGI20

7.1.3. Evaluación convocatoria extraordinaria

No se ha definido la evaluación extraordinaria.

7.2. Criterios de evaluación

Evaluación progresiva.

La evaluación progresiva se realizará a través de la evaluación de los retos que constituyen la base de la metodología de la asignatura. En cada reto, los alumnos deben aplicar todas las técnicas y recursos que se han ido incorporando en los retos anteriores, de esta manera el conocimiento de los mismos y su capacidad de aplicarlos para resolver problemas concretos se irá evaluando progresivamente según los van adquiriendo. Los retos se realizarán en equipos determinados por el profesor.

Aunque la propia resolución de los retos no es presencial, los alumnos deben **participar presencialmente** en la revisión de todos los retos y presentar la resolución de algunos de ellos. La revisión de las soluciones aportadas en cada reto tiene un valor formativo y, por eso se evalúa la asistencia y aprovechamiento, junto con la evaluación de la solución aportada.

Adicionalmente se realizará un examen escrito individual tipo cuestionario con preguntas de respuesta corta sobre los conceptos y aplicación de los distintos temas de la asignatura.

Evaluación global.

El examen teórico - práctico tendrá el mismo formato que en la evaluación progresiva.

En la evaluación global se sustituyen los retos semanales realizados para la evaluación progresiva por la realización y presentación de un único reto presencialmente. En ningún caso la resolución del reto se podrá realizar en remoto, debe ser presencialmente.

Dado que en la evaluación global se deben incluir todo el temario y evaluar todas las competencias adquiridas que ha sido trabajadas semana a semana durante el curso completo, en esta modalidad de evaluación el reto tendrá un alcance y complejidad propio de un reto de fin de curso y se incluirán todos los temas y herramientas trabajados durante el curso. Si el reto no se puede completar en el tiempo previsto se entiende por no superado y, por tanto, suspenso.

Evaluación Extraordinaria.

Para la evaluación extraordinaria se vuelven a realizar las dos pruebas de evaluación de la evaluación global: "examen teórico - práctico" y "resolución de un reto y su presentación".

8. Recursos didácticos

8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Scott Meyers: "Effective c++", 3ª Edición Addison-Wesley	Bibliografía	Programación en C++
Scott Meyers: "Effective Modern c++", 3º Edición Addison-Wesley, O'Reilly	Bibliografía	Programación en C++
George Coulouris: "Distributed System. Concepts and Design", 5ª Edición, Addison-Wesley.	Bibliografía	Conceptos general sobre sistemas distribuidos
Pieter Hintjens: "OMQ - The Guide", https://zguide.zeromq.org/	Recursos web	Documentación de ZeroMQ (Middleware y descripción de patrones de comunicaciones)

9. Adendas

- Se añaden a las competencias de la asignatura, las siguientes competencias específicas del máster: CEM2, CEM4 y CEM6. CEM2: Analizar y sintetizar soluciones a problemas que requieran aproximaciones novedosas para la definición de la infraestructura computacional que permita el procesamiento y el análisis de datos de diversa naturaleza. CEM4: Analizar y evaluar los diferentes paradigmas y enfoques de ingeniería de construcción y gestión de sistemas basados en software. CEM6: Realizar trabajos de investigación en las principales líneas de investigación activas en el área de los paradigmas de la computación distribuida, sus aplicaciones prácticas y la gestión de la infraestructura necesaria