

Guía docente

230160 - SAD - Software para Aplicaciones Distribuidas

Última modificación: 18/06/2026

Unidad responsable: Escuela Técnica Superior de Ingeniería de Telecomunicación de Barcelona
Unidad que imparte: 744 - ENTEL - Departamento de Ingeniería Telemática.

Titulación: GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN (Plan 2015). (Asignatura optativa).

Curso: 2026 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: MARCEL FERNANDEZ MUÑOZ

Otros: Primer quadrimestre:
MARCEL FERNANDEZ MUÑOZ - 41
JORDI FORGA ALBERICH - 41

CAPACIDADES PREVIAS

Familiaridad en el uso de entornos de programación i protocolos de comunicación.

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Genéricas:

10 ECI N3. EXPERIMENTALIDAD Y CONOCIMIENTO DE LA INSTRUMENTACIÓN: desenvolverse de forma competente en un entorno de laboratorio del ámbito TIC. Operar instrumentación y herramientas propias de las ingenierías de telecomunicación y electrónica e interpretar sus manuales y especificaciones. Evaluar los errores y las limitaciones asociados a las medidas y resultados de simulaciones.

METODOLOGÍAS DOCENTES

Clases expositivas
Clases Laboratorio
Trabajo en grupo (no presencial)
Trabajo individual (no presencial)
Ejercicios
Pruebas de respuesta corta (Control)
Pruebas de respuesta larga (Control)
Pruebas de respuesta larga (Examen Final)
Práctica de laboratorio

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

El objetivo de esta asignatura es formar al alumno en los métodos de programación de aplicaciones distribuidas y sus ámbitos de aplicación.

La primera parte del curso se dedica al aprendizaje del lenguaje de programación Haskell. Se trata de un lenguaje de programación funcional puro que utiliza un paradigma de programación completamente diferente a los lenguajes de programación imperativa, como Java o C/C++. Esto hace que el aprendizaje de este lenguaje no esté exento de cierta dificultad por la falta de costumbre, hasta que se alcanza un cierto grado de conocimiento sobre el mismo. Este lenguaje también se utilizará en la asignatura DAT.

En la última parte de la asignatura se usarán estos conocimientos para diseñar e implementar aplicaciones basadas en el modelo cliente/servidor.

Resultado del aprendizaje:

- Puede programar aplicaciones de gestión del conocimiento distribuido y servicios interactivos, con criterios de usabilidad y accesibilidad de servicios.
- Está familiarizado con los protocolos e interfaces de comunicación en los diferentes niveles de una arquitectura de redes y es capaz de describirlos, programarlos, validarlos y optimizarlos.
- Es capaz de diseñar arquitecturas cliente-servidor y P2P, y adaptar sistemas operativos y máquinas virtuales.
- Utiliza estrategias para preparar y realizar las presentaciones orales y redactar textos y documentos con un contenido coherente, una estructura y un estilo adecuados, y un buen nivel ortográfico y gramatical.
- Utiliza de forma autónoma las herramientas, instrumentos y aplicativos de software disponibles en los laboratorios de las materias básicas y avanzadas. Conoce su funcionamiento y sus limitaciones.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTE

Tipo	Horas	Porcentaje
Horas grupo pequeño	26,0	17.33
Horas grupo grande	39,0	26.00
Horas aprendizaje autónomo	85,0	56.67

Dedicación total: 150 h

CONTENIDOS

Tema1. Aspectos básicos de Haskell

Descripción:

Tipos Básicos. Funciones. Tipificado. Declaraciones Locales. Bloques de Control. Recursividad.

Dedicación: 6h

Grupo grande/Teoría: 6h

Tema2. Tipos de datos

Descripción:

Expresiones, Constructores de Valor Paramétricos, Constructores de Tipo Paramétricos, Patrones, Clases e Instancias, y Listas.

Dedicación: 4h

Grupo grande/Teoría: 4h

Tema3. Funciones

Descripción:

Estrategias de Recursividad. Funciones de Orden Superior.

Dedicación: 6h

Grupo grande/Teoría: 6h

Tema4. Introducción a vFunctores y aplicativos.

Descripción:

Functores y aplicativos.

Dedicación: 4h

Grupo grande/Teoría: 4h

Tema5. Introducción a los Monads.

Descripción:

Monad Maybe, monad reader, monad estado.

Dedicación: 4h

Grupo grande/Teoría: 4h

Tema6. Concurrencia.

Descripción:

Threads. MVars.

Dedicación: 3h

Grupo grande/Teoría: 3h

Tema7. Sockets.

Descripción:

Programación mediante sockets. Modelo cliente-servidor básico.

Dedicación: 6h

Grupo grande/Teoría: 6h

ACTIVIDADES

Práctica1. Introducción a las funciones.

Descripción:

Actividades relacionadas con la definición, composición i uso de funciones puras.

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

Práctica2. Tipos de datos.

Descripción:

Actividades relacionadas con el use de tipos de datos que proporciona el propio lenguaje

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

Práctica3. Tipos recursivos. Listas.

Descripción:

Actividades relacionadas con tipos de datos estructurados que se utilizan en sistemas distribuidos

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

Práctica4. Clases y instancias.

Descripción:

Actividades relacionadas con los mecanismos de abstracción de Haskell.

Dedicación: 4h

Grupo pequeño/Laboratorio: 4h

Práctica5. Mini proyecto.

Dedicación: 4h

Grupo pequeño/Laboratorio: 4h

Práctica5. Aplicativos.

Descripción:

Analizadores de expresiones utilizados en aplicaciones distribuidas mediante programación aplicativa

Dedicación: 4h

Grupo pequeño/Laboratorio: 4h

Práctica6. Monads.

Descripción:

Juego criptográfico para emular mecanismos de seguridad en aplicaciones distribuidas

Dedicación: 4h

Grupo pequeño/Laboratorio: 4h



Pràctica8. Modelo Client-Servidor Básico.

Descripción:

Aplicación distribuida según el paradigma cliente servidor

Dedicación: 2h

Grupo pequeño/Laboratorio: 2h

Práctica7. Concurrencia

Descripción:

Uso de threads y mecanismos de sincronización en aplicaciones distribuidas

Dedicación: 4h

Grupo pequeño/Laboratorio: 4h

SISTEMA DE CALIFICACIÓN

Cualquier acto de fraude académico, plagio o uso o mera tenencia al alcance de medios no autorizados en cualquier actividad de evaluación comportará la calificación de cero (0) en la prueba o entrega afectada. Además, de acuerdo con la normativa de la Universidad, la posible derivación de los hechos para la apertura de un expediente disciplinario implicará que la asignatura quede en el estado provisional de "pendiente de evaluación" hasta la resolución del expediente. La gestión de estas incidencias se lleva a cabo de acuerdo con el [Marco de actuación para la integridad académica en la evaluación de la UPC](#).

Todas las calificaciones se obtienen a partir de controles escritos.

Control mediante examen de mitad de curso: Preguntas sobre la primera mitad de las prácticas de laboratorio: LAB1
Preguntas sobre la primera mitad teórica del curso (recuperable al final): TEO1

Control mediante examen a final de curso: Preguntas sobre todas las prácticas de laboratorio: LAB2
Preguntas sobre toda la parte teórica del curso: TEO2

Obtención de la nota definitiva: $0.4 * (\max(\text{LAB2}, 0.5 * \text{LAB1} + 0.5 * \text{LAB2})) + 0.6 * (\max(\text{TEO2}, 0.65 * \text{TEO2} + 0.35 * \text{TEO1}))$

El control final tiene carácter global, ya que evalúa el conjunto de los contenidos teóricos y prácticos de la asignatura. Las calificaciones obtenidas en este control sustituyen a las calificaciones correspondientes de la evaluación continua cuando el resultado es superior, de acuerdo con la fórmula de calificación establecida.

BIBLIOGRAFÍA

Básica:

- Marlow, Simon. Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithreaded Programming [en línea]. Sebastopol, CA: O'Reilly, 2013 [Consulta: 10/07/2025]. Disponible a: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=1313476>. ISBN 9781306812450.
- Hutton, Graham. Programming in Haskell [en línea]. Cambridge, England: Cambridge University Press, [2016] [Consulta: 10/07/2025]. Disponible a: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pq-origsite=primo&docID=4620865>. ISBN 1316863220.