

Curso Académico: ( 2021 / 2022 )

Fecha de revisión: 28-06-2021

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: GARCIA CARBALLEIRA, FELIX

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 3 Cuatrimestre : 2

**REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)**

Sistemas Operativos (Curso 2 - cuatrimestre 2)  
Redes de ordenadores (Curso 3 - cuatrimestre 1)

**OBJETIVOS**

El objetivo de este curso es describir los principales conceptos necesarios para diseñar y desarrollar aplicaciones y sistemas distribuidos.

**DESCRIPCIÓN DE CONTENIDOS: PROGRAMA**

Los descriptores asociados con la asignatura son:  
Comunicación entre procesos; Middleware; RPC; almacenamiento distribuido; aplicaciones distribuidas; sincronización; tolerancia a fallos.

Temario:

Temario:

1. Introducción y conceptos básicos
  - Conceptos básicos
  - Redes e interconexión
  - Ventajas de los sistemas distribuidos
  - Paradigmas de computación distribuida
  - Diseño de sistemas distribuidos
2. Comunicación y sincronización entre procesos
  - Mecanismos de comunicación en sistemas de memoria compartida
  - Mecanismos de comunicación en sistemas de paso de mensajes
  - Servicios POSIX de comunicación y sincronización
  - Threads en Java/Python
3. Paso de mensajes y aplicaciones cliente-servidor
  - Modelo de comunicación mediante paso de mensajes
  - Aspectos de diseño
  - Colas de mensajes POSIX
  - Aplicaciones cliente-servidor
  - Diseño de servidores concurrentes
4. Comunicación con sockets
  - Modelo de comunicación con sockets
  - API de sockets en POSIX
  - API de sockets en Java/Python
  - Guía de diseño de aplicaciones cliente-servidor con sockets
5. Llamadas a procedimientos remotos
  - Funcionamiento de las RPC
  - Lenguaje de definición de interfaces
  - Aplanamiento y transferencia de mensajes
  - ONC-RPC
  - Invocación de métodos remotos en JAVA (RMI)
6. Servicios web

- Protocolo HTTP
- SOAP
- Desarrollo de servicios web

#### 7. Servicios distribuidos

- Servicio de nombres
- Sincronización en sistemas distribuidos
- Relojes físicos y lógicos
- Exclusión mutua distribuida
- Algoritmos de elección
- Comunicación multicast

#### 8. Sistemas de almacenamiento distribuidos

- Estructura de un sistema de ficheros distribuido
- Servicio de ficheros y directorios
- Implementación de servicios de ficheros distribuidos
- Ejemplo: NFS
- Sistemas de ficheros de discos compartidos
- Sistemas de ficheros paralelos
- Redes de almacenamiento

#### 9. Tolerancia a fallos en sistemas distribuidos

- Conceptos básicos sobre tolerancia a fallos
- Tolerancia a fallos software
- Detectores de fallos
- Replicación
- Protocolos de consenso

### ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

\* Clases magistrales: 1 ECTS. Tienen por objetivo alcanzar las competencias específicas cognitivas de la asignatura, así como las competencias transversales capacidad de análisis y abstracción.

\* Clases prácticas: 1 ECTS. Tienen por objetivo iniciar el desarrollo de las competencias específicas instrumentales, así como las competencias transversales resolución de problemas y aplicación de conocimientos.

\* Ejercicios de evaluación continua: 1,5 ECTS. Iniciados durante las clases prácticas y terminados fuera de las mismas, tienen por objetivo completar el desarrollo de las competencias específicas instrumentales e iniciar el desarrollo de las competencias específicas actitudinales, así como las competencias transversales resolución de problemas y aplicación de conocimientos.

\* Trabajos prácticos: 2 ECTS. Desarrollados sin presencia del profesor, tienen por objetivo completar e integrar el desarrollo de todas las competencias específicas y transversales, en la resolución de dos casos prácticos donde queden bien documentados el planteamiento del problema, la elección del método de resolución, los resultados obtenidos y la interpretación de los mismos.

\* Tutorías: TUTORÍAS. Asistencia individualizada (tutorías individuales) o en grupo (tutorías colectivas) a los estudiantes por parte del profesor.

\* Examen final: 0,5 ECTS. Tiene por objeto incidir y complementar en el desarrollo de las capacidades específicas cognitivas y procedimentales. Refleja especialmente el aprovechamiento de las clases magistrales.

### SISTEMA DE EVALUACIÓN

La evaluación será continua y consta de las siguientes partes:

#### 1. Evaluación continua (80%)

- Ejercicios, trabajos y proyectos de programación: 40%
- Práctica: 40%

#### 2. Examen final (20%)

La nota mínima en el examen final será de 4.

La nota media mínima de todas las prácticas será de 4.

Se considera que un alumno sigue el proceso de evaluación continua cuando:

- Entrega todos los trabajos y prácticas propuestos, teniendo en cuenta que la nota mínima necesaria para cada práctica es de 2.
- La nota media de todas las prácticas es de 4.

El examen final de la convocatoria extraordinaria incluirá los conceptos teóricos y prácticos de la asignatura.

<b>Peso porcentual del Examen Final:</b>	20
<b>Peso porcentual del resto de la evaluación:</b>	80

#### BIBLIOGRAFÍA BÁSICA

- G. Coulouris, J. Dollimore, T. Kindberg, G. Blair Distributed Systems, Concepts and design. 5ª edition. 2011, Addison-Wesley.

#### BIBLIOGRAFÍA COMPLEMENTARIA

- B. A. Forouzan Transmisión de datos y redes de comunicaciones. Cuarta edición, McGraw Hill.
- Distributed Systems: principles and paradigms Andrew S. Tanenbaum , Maarten van Steen, Pearson.
- F. García, J. Carretero, A. Calderón, J. Fernández, J. M. Pérez Problemas resueltos de programación en C, Thomson.
- L. H. Etzkorn Introduction to Middleware: Web Services, Object Components, and Cloud Computing, CRC Press, 2017
- Pankaj Jalote Fault Tolerance in Distributed Systems, Prentice-Hall.
- Richard Stevens UNIX Network Programming, Prentice Hall.

#### RECURSOS ELECTRÓNICOS BÁSICOS

- Félix García Carballeira . Material de sistemas distribuidos: <http://www.arcos.inf.uc3m.es/~infosd>