

## GUÍA DOCENTE 2024-2025

### DATOS GENERALES DE LA ASIGNATURA

<b>ASIGNATURA:</b>	Sistemas Distribuidos y Programación en Paralelo		
<b>PLAN ESTUDIOS:</b>	<b>DE</b>	Grado en Ingeniería Informática	
<b>FACULTAD :</b>	Escuela Politécnica Superior		
<b>CARÁCTER ASIGNATURA:</b>	<b>DE</b>	<b>LA</b>	Obligatorio
<b>ECTS:</b>	6		
<b>CURSO:</b>	Tercero		
<b>SEMESTRE:</b>	Primero		
<b>IDIOMA EN QUE SE IMPARTE:</b>	<b>EN</b>	<b>QUE</b>	<b>SE</b> Español
<b>PROFESORADO:</b>	Daniel Iglesias Santamaría		
<b>DIRECCIÓN DE CORREO ELECTRÓNICO:</b>	daniel.iglesias@uneatlantico.es		

### DATOS ESPECÍFICOS DE LA ASIGNATURA

<b>REQUISITOS PREVIOS:</b>
Se recomienda que para cursar la asignatura de Sistemas distribuidos y Programación en paralelo el alumno haya realizado previamente las asignaturas de Sistemas Operativos, Redes de ordenadores y Programación II.
<b>CONTENIDOS:</b>
Tema 1 : Visual Studio Community y .NET <ul style="list-style-type: none"> <li>1.2 Conceptos básicos de .NET</li> <li>1.3 WCF (Windows Communication Foundation)</li> <li>1.4 .Net remoting</li> </ul>

Tema 2 : Introducción a los Sistemas Distribuidos

- 2.1 Definición
- 2.2 Características
- 2.3 Paradigmas
- 2.4 Tipos de Servidores
- 2.5 Protocolo
- 2.6 Middleware

Tema 3: Arquitecturas Distribuidas

- 3.1 Cliente Servidor
- 3.2 Arquitectura de capas
- 3.3 Thin and fat clients
- 3.4 Arquitectura de objetos distribuidos
- 3.5 Peer to Peer (P2P)
- 3.6 SOA (Service Oriented Architecture)

Tema 4: Arquitectura Peer to Peer

- 4.1 ¿Qué es peer to peer?
- 4.2 Aplicación de Chat (QuickReturnTraderChat)

Tema 5: Servicios Web y Rest (SOA)

- 5.1 Tecnologías que soportan SOA
- 5.2 Servicio Web y Rest
- 5.3 Implementar mejoras.

Tema 6: Programación distribuida en .Net

- 6.1 Conceptos básicos
- 6.2 .Net Remoting

Tema 7: Servicios de Integración y mensajería.

- 7.1 Service Broker
- 7.2 Enterprise Service Bus (ESB)
- 7.3 Microservices

Tema 8: Cloud Computing

- 8.1 Tipos de servicios
- 8.2 Principales proveedores

Tema 9: Map Reduce

- 9.1 Historia
- 9.2 Map and Reduce
- 9.3 Arquitectura

Tema 10: Programación en paralelo

10.1 Asynchronous programming (Async, Await)

10.2 Parallel Programming

## COMPETENCIAS

### COMPETENCIAS GENERALES:

Que los estudiantes sean capaces de:

- CG4 Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- CG6 Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.

### COMPETENCIAS ESPECÍFICAS:

Que los estudiantes sean capaces de:

- CE17 Capacidad de conocer y aplicar las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.
- CE20 Capacidad de conocer y aplicar los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

### RESULTADOS DE APRENDIZAJE:

En esta asignatura se espera que los alumnos alcancen los siguientes resultados de aprendizaje:

- Entender las técnicas de comunicación de procesos.
- Entender un entorno de sistema distribuido y ser capaz de programar aplicaciones en este entorno.
- Conocer y comprender la utilidad y el funcionamiento de la arquitectura TCP/IP.

## METODOLOGÍAS DOCENTES Y ACTIVIDADES FORMATIVAS

### METODOLOGÍAS DOCENTES:

En esta asignatura se ponen en práctica diferentes metodologías docentes con el objetivo de que los alumnos puedan obtener los resultados de aprendizaje definidos anteriormente:

- MD1 Método expositivo
- MD2 Estudio y análisis de casos
- MD4 Aprendizaje basado en problemas
- MD6 Aprendizaje cooperativo / Trabajo en grupos
- MD7 Trabajo autónomo

#### ACTIVIDADES FORMATIVAS:

A partir de las metodologías docentes especificadas anteriormente, en esta asignatura, el alumno participará en las siguientes actividades formativas:

Actividades formativas		Horas
Actividades dirigidas	Clases expositivas	15
	Clases prácticas	18,8
	Seminarios y talleres	11,3
Actividades supervisadas	Supervisión de actividades	3,8
	Tutorías (individual / en grupo)	3,8
Actividades autónomas	Preparación de clases	15
	Estudio personal y lecturas	37,5
	Elaboración de trabajos	30
	Trabajo en campus virtual	7,5

El primer día de clase, el profesor proporcionará información más detallada al respecto.

## SISTEMA DE EVALUACIÓN

#### CONVOCATORIA ORDINARIA:

En la convocatoria ordinaria de esta asignatura se aplican los siguientes instrumentos de evaluación:

Actividades de evaluación		Ponderación
Evaluación continua	1 Examen Parcial	25 %
	Entregas de Portfolios y Ejercicios	20 %
	Interés y participación del alumno en la asignatura	5 %
Evaluación final	Examen Teórico-Práctico	50 %

La calificación del instrumento de la evaluación final (tanto de la convocatoria ordinaria como de la extraordinaria, según corresponda) **no podrá ser inferior, en ningún caso, a 4,0 puntos** (escala 0 a 10) para aprobar la asignatura y consecuentemente poder realizar el cálculo de porcentajes en la calificación final.

#### CONVOCATORIA EXTRAORDINARIA:

La convocatoria extraordinaria tendrá lugar durante el mes de julio (consúltese el calendario académico fijado por la universidad). Esta consistirá en la realización de un Examen Teórico-Práctico con un valor del 50 % de la nota final de la asignatura. El resto de la nota se complementará con la calificación obtenida en la evaluación continua de la convocatoria ordinaria.

## BIBLIOGRAFÍA Y RECURSOS DE REFERENCIA GENERALES

#### BIBLIOGRAFÍA BÁSICA:

Las siguientes referencias son de consulta obligatoria:

- Roy, O. (2013) The Art of Unit Testing. Ed. Manning Publications
- Tanenbaum, A. y Van Steen, M. (2008) Sistemas Distribuidos: Principios y Paradigmas. Ed. Pearson Educación

#### **BIBLIOGRAFÍA COMPLEMENTARIA:**

No aplica.

#### **WEBS DE REFERENCIA:**

- <https://docs.microsoft.com/es es/dotnet/>
- <https://docs.microsoft.com/es es/dotnet/csharp/>
- <https://visualstudio.microsoft.com/es/vs/>
- <https://azure.microsoft.com/es es/overview/what is devops/>

#### **OTRAS FUENTES DE CONSULTA:**

No aplica.