

**FICHA IDENTIFICATIVA****Datos de la Asignatura**

|                        |              |
|------------------------|--------------|
| <b>Código</b>          | 34796        |
| <b>Nombre</b>          | Programación |
| <b>Ciclo</b>           | Grado        |
| <b>Créditos ECTS</b>   | 6.0          |
| <b>Curso académico</b> | 2023 - 2024  |

**Titulación(es)**

| <b>Titulación</b>                                          | <b>Centro</b>                          | <b>Curso</b> | <b>Periodo</b>       |
|------------------------------------------------------------|----------------------------------------|--------------|----------------------|
| 1402 - Grado de Ingeniería Electrónica de Telecomunicación | Escuela Técnica Superior de Ingeniería | 2            | Segundo cuatrimestre |

**Materias**

| <b>Titulación</b>                                          | <b>Materia</b>   | <b>Caracter</b> |
|------------------------------------------------------------|------------------|-----------------|
| 1402 - Grado de Ingeniería Electrónica de Telecomunicación | 9 - Programación | Obligatoria     |

**Coordinación**

| <b>Nombre</b>          | <b>Departamento</b> |
|------------------------|---------------------|
| ORTIGOSA ARAQUE, NURIA | 240 - Informática   |

**RESUMEN**

La asignatura Programación tiene 6 ECTS, se imparte en el segundo cuatrimestre del segundo curso en el Grado en Ingeniería Electrónica de Telecomunicación. Su finalidad es la de proporcionar a los alumnos una introducción al lenguaje de programación Java y proporcionar una visión amplia de diferentes APIs para el desarrollo de aplicaciones en red y distribuidas. Tras cursar la asignatura los alumnos deben ser capaces de desarrollar aplicaciones en red y distribuidas usando correctamente la orientación a objetos, tipos parametrizados, jerarquías de clases, la concurrencia y la sincronización de tareas concurrentes.

Los objetivos generales se enumeran a continuación:

- Programar aplicaciones utilizando correctamente los conceptos de orientación a objetos.
- Declarar y usar de forma apropiada jerarquías de clases, clases abstractas, interfaces y tipos parametrizados.
- Desarrollar aplicaciones que utilicen concurrencia y recursos compartidos que sincronicen tareas mediante semáforos o monitores.
- Crear flujos de entrada o salida apropiados según las especificaciones.
- Desarrollar aplicaciones en red usando diferentes protocolos.



- Usar entornos de desarrollo integrados para el desarrollo, depuración y ejecución de las aplicaciones. Usar las herramientas apropiadas para compilar y ejecutar aplicaciones.

## CONOCIMIENTOS PREVIOS

### Relación con otras asignaturas de la misma titulación

No se han especificado restricciones de matrícula con otras asignaturas del plan de estudios.

### Otros tipos de requisitos

Sin haber requisitos previos de matrícula, para una adecuada comprensión de la asignatura se recomienda cursar la materia: Informática.

## COMPETENCIAS

### 1402 - Grado de Ingeniería Electrónica de Telecomunicación

- G3 - Conocimiento de materias básicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías, así como que le dote de una gran versatilidad para adaptarse a nuevas situaciones.
- G4 - Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, y de comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación.
- R7 - Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación.

## RESULTADOS DE APRENDIZAJE

- Declarar clases para crear nuevos tipos, creación objetos y envío de mensajes. Usar los tipos definidos por el lenguaje (R7).
- Declarar y usar jerarquías de clases (R7).
- Declarar y usar tipos parametrizados y usar los que proporciona el lenguaje (R7).
- Declarar y usar hilos para la ejecución concurrente de código (R7).
- Explicar y usar semáforos y monitores para la sincronización de tareas en el acceso a recursos compartidos (R7).
- Enumerar y usar flujos de entrada y de salida. Usar flujos que añadan funcionalidad a los flujos de bajo nivel (R3).
- Declarar clases cuyas instancias se puedan serializar a través de flujos (R7).
- Desarrollar aplicaciones que utilicen el protocolo UDP tanto unicast como multicast (R7).
- Desarrollar aplicaciones que utilicen el protocolo TCP mediante sockets (R7).
- Usar entornos de desarrollo integrados para el desarrollo, depuración y ejecución de las aplicaciones (R2).



Además, el alumno debe desarrollar las siguientes destrezas:

- Encontrar e interpretar la información del API de Java (G3).
- Discutir las ventajas/desventajas de diferentes aproximaciones a la hora de resolver un problema dado (G4).
- Elaborar, redactar y presentar textos académicos. Además se realizará énfasis en que cualquier pieza escrita debe tener unos estándares de calidad basados en la cohesión, la claridad, la lógica, etc., de los argumentos y evidencias que contiene (G4).

## DESCRIPCIÓN DE CONTENIDOS

### 1. Orientación a objetos con Java

Revisión de conceptos: Clases, métodos, objetos, mensajes y encapsulación.

Referencias frente a tipos primitivos

Herencia, jerarquías de clases, clases abstractas, interfaces, polimorfismo

Tipos parametrizados: declaración y uso.

Excepciones: declaración y tratamiento.

### 2. Programación concurrente

Tareas concurrentes a nivel lógico: hilos

Problemas en el acceso a recursos compartidos: sección crítica

Mecanismos de sincronización de tareas concurrentes: semáforos y monitores

### 3. Entrada / Salida

Flujos orientados a bytes de entrada y salida de bajo nivel y filtrados

Flujos orientados a caracteres de entrada y salida de bajo nivel y filtrados

Serialización de objetos

### 4. Programación en red y distribuida

Aplicaciones basadas en los protocolos UDP, TCP y HTTP

**VOLUMEN DE TRABAJO**

| ACTIVIDAD                                      | Horas         | % Presencial |
|------------------------------------------------|---------------|--------------|
| Clases de teoría                               | 30,00         | 100          |
| Prácticas en laboratorio                       | 20,00         | 100          |
| Prácticas en aula                              | 10,00         | 100          |
| Elaboración de trabajos en grupo               | 10,00         | 0            |
| Estudio y trabajo autónomo                     | 20,00         | 0            |
| Lecturas de material complementario            | 10,00         | 0            |
| Preparación de clases de teoría                | 10,00         | 0            |
| Preparación de clases prácticas y de problemas | 40,00         | 0            |
| <b>TOTAL</b>                                   | <b>150,00</b> |              |

**METODOLOGÍA DOCENTE**

Las metodologías que se proponen para esta asignatura son:

- Clases de teoría. Desarrollo de los temas proporcionando una visión global e integradora, analizando con mayor detalle los aspectos clave y de mayor complejidad, fomentando, en todo momento, la participación del alumnado. Previo a la asistencia a cada clase, se propone un tests online para que el alumno revise la información a tratar en clase, y así potenciar la participación y la resolución de dudas en el aula.
- Clases prácticas en aula. Complementan las lecciones expositivas con el objetivo de aplicar los conceptos básicos y ampliarlos con el conocimiento y la experiencia que vayan adquiriendo durante la realización de los trabajos propuestos. En estas clases se resuelven cuestiones y problemas en el aula.
- Clases prácticas en laboratorio. Las prácticas se realizarán individualmente o en pequeños grupos en los que los alumnos desarrollarán los contenidos teóricos y prácticos mediante su aplicación en casos realistas utilizando el material específico correspondiente y bajo la supervisión del profesor. En cada práctica, los alumnos realizarán un ejercicio previo a la práctica que se entregará antes de la realización de la práctica. Este ejercicio tiene como objetivo analizar el trabajo que se va a realizar en el laboratorio, y relacionarlo con los conceptos vistos en las clases teóricas. La asistencia a las prácticas es obligatoria y es una actividad no recuperable.
- Trabajo autónomo del estudiante. Realización de trabajos, cuestiones, problemas fuera del aula, búsquedas bibliográficas, así como la preparación de clases y exámenes (estudio). La realización de estas actividades será unas veces individual, para potenciar la autonomía del estudiante, y otras veces en pequeños grupos, para potenciar la capacidad de integración en grupos de trabajo, así como la capacidad de liderazgo y de coordinación.
- Aula Virtual. Se utilizar la plataforma de e-learning (Aula Virtual) de la Universitat de València como soporte de comunicación con el alumnado, y también para la realización de tests y ejercicios correctores online. A través de ella se tendrá acceso al material didáctico utilizado en clase.



## EVALUACIÓN

La evaluación de la asignatura tendrá en cuenta las siguientes dimensiones:

- Evaluación continua, basada en la participación y grado de implicación en el proceso de enseñanza-aprendizaje: tests online previos a las clases de teoría y la asistencia regular a las actividades presenciales previstas y la resolución de cuestiones y problemas propuestos que se plantearán en el Aula virtual (*N\_Continua*) (R2, R3).
- Prueba intermedia individual de evaluación continua (examen parcial) que contendrá cuestiones teóricas-prácticas y problemas. Esta prueba eliminará materia para el examen final siempre que la nota sea mayor o igual que 5, y hará media con la nota del examen final (*Nproves*) (R7).
- Evaluación de las actividades desarrolladas en el laboratorio. Puntualmente se podrán realizar exposiciones orales (individualmente y/o en grupo) para evaluar la capacidad de elaboración de documentos y transmisión de conocimientos (*N\_Practicas*). Para evaluar la nota de cada sesión, el trabajo previo puntuará un 35% y el desarrollo de la práctica un 65% (R7).

$$\text{Nota Final} = 10\% N\_Continua + 60\% (N\_Pruebas) + 30\% N\_Practicas$$

Para poder calcular la nota final todas las pruebas individuales (examen parcial y final) tienen que tener una nota igual o superior a 4.

En segunda convocatoria, se realizará un examen correspondiente a teoría/problemas/laboratorio. La nota final en la segunda convocatoria se realizará con la misma propuesta que la primera convocatoria o con una evaluación alternativa:

- Evaluación alternativa a la evaluación continua tendrá un examen final individual que contendrá tanto cuestiones teórico-prácticas como de problemas (*N\_Examen*).

$$\text{Nota Final} = 70\% (N\_Examen) + 30\% N\_Practiques$$

Para poder calcular la nota Final *N\_Examen* tendrá que ser mayor o igual a 4.

“En cualquier caso, el sistema de evaluación se regirá por el establecido en el Reglament de Avaluació i Qualificació de la Universitat de València per a Graus i Màsters

<https://webges.uv.es/uvTaeWeb/MuestraInformacionEdictoPublicoFrontAction.do?accion=inicio&idEdictoSeleccionado=5639>”.

## REFERENCIAS

### Básicas

- Java. Cómo Programar. P. J. Deitel y H. M. Deitel. Pearson Educación, Séptima edición, 2008
- Java Network Programming and Distributed Computing. David Reilly, Michael Reilly. Addison-Wesley



### **Complementarias**

- Core Java 2. Volumen I. Cay S. Horstmann ; Gary Cornell, Prentice Hall, séptima edición, 2005
- Core Java 2. Volumen II. Cay S. Horstmann ; Gary Cornell, Prentice Hall, séptima edición, 2006