

**CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL**



**PLANIFICACIÓN DE LA DOCENCIA  
UNIVERSITARIA**

**GUÍA DOCENTE**

**PARADIGMAS DE PROGRAMACIÓN**

# 1. DATOS DE IDENTIFICACIÓN DE LA ASIGNATURA.

Título:	Grado en Ingeniería del Software
Facultad:	Centro Universitario de Tecnología y Arte Digital (U-TAD)
Materia:	Programación
Denominación de la asignatura:	Paradigmas de Programación
Curso:	4
Cuatrimestre:	1
Carácter:	Obligatoria
Créditos ECTS:	3
Modalidad/es de enseñanza:	Híbrido Presencial
Idioma:	Castellano
Profesor/a:	Álvaro San Juan Cervera
E-mail:	alvaro.san@u-tad.com
Teléfono:	

## 2. DESCRIPCIÓN DE LA ASIGNATURA.

### 2.1 Descripción de la materia

Esta asignatura pertenece a la materia de programación. Esta materia se dedica al estudio de las técnicas y los lenguajes de programación en los que se fundamentarán los estudios del grado de ingeniería del software.

### 2.2 Descripción de la asignatura

Durante la carrera hemos visitado los principales paradigmas de programación de las últimas décadas mediante la programación estructurada y la programación orientada a objetos. En esta asignatura nos centraremos en otros enfoques como la programación funcional y la programación reactiva.

### 3. COMPETENCIAS

#### Competencias Básicas y Generales

CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

CB2: Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB3: Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas.

CG3 - Conocimiento de los fundamentos científicos aplicables a la resolución de problemas informáticos

CG4 - Capacidad para simplificar y optimizar los sistemas informáticos atendiendo a la comprensión de su complejidad

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

CG10 - Uso de técnicas creativas para la realización de proyectos informáticos

#### Competencias Específicas

CE1 - Conocimiento de la estructura de las computadoras, de los conceptos de codificación, manipulación, tratamiento de la información y uso de lenguajes de bajo nivel

CE7 - Conocimiento de los principales tipos de estructuras de datos y utilización de las librerías y de los técnicas algorítmicas asociadas a dichas estructuras junto con los órdenes de complejidad que caracterizan a dichas técnicas

CE8 - Conocimiento de los distintos paradigmas detrás de los lenguajes de programación

CE9 - Conocimiento de las estructuras de control, variables, sintaxis de programación y gestión del uso de la memoria de manera eficaz en el desarrollo de una aplicación informática

CE10 - Capacidad para manejar un gestor de versiones de código y generar la documentación de una aplicación de forma automática.

CE15 - Conocimiento de la tolerancia a los fallos, la adaptabilidad, el balance de carga y la predictividad del sistema para el desarrollo de aplicaciones distribuidas

CE17 - Conocimiento de las características de paralelización de tarjetas gráficas y de arquitecturas de altas prestaciones para el desarrollo de aplicaciones.

CE20 - Capacidad para testar el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas

CE23 - Conocimiento de los principios de la inteligencia artificial y uso de algoritmos de búsqueda deterministas y máquinas de estado

## 4. CONTENIDOS

- Introducción a los paradigmas de programación
- Programación reactiva
- Programación funcional

## 5. ACTIVIDADES FORMATIVAS Y MODALIDADES DE ENSEÑANZAS

### 5.1 Modalidades de enseñanza

La asignatura se desarrollará a través de los siguientes métodos y técnicas generales, que se aplicarán diferencialmente según las características propias de la asignatura:

- **Método expositivo/Lección magistral:** el profesor desarrollará, mediante clases magistrales y dinámicas los contenidos recogidos en el temario.
- **Estudio de casos:** análisis de casos reales relacionados con la asignatura.
- **Resolución de ejercicios y problemas:** los estudiantes desarrollarán las soluciones adecuadas aplicando procedimientos de transformación de la información disponible y la interpretación de los resultados.
- **Aprendizaje basado en problemas:** utilización de problemas como punto de partida para la adquisición de conocimientos nuevos.
- **Aprendizaje orientado a proyectos:** se pide a los alumnos que, en pequeños grupos, planifiquen, creen y evalúen un proyecto que responda a las necesidades planteadas en una determinada situación.
- **Aprendizaje cooperativo:** Los estudiantes trabajan en grupo para realizar las tareas de manera colectiva.

## 5.2 Actividades formativas

Actividad Formativa	Horas	Presencialidad
AF1 Clases teóricas / Expositivas	15	100%
AF2 Clases Prácticas	12	100%
AF3 Tutorías	3	50%
AF4 Estudio independiente y trabajo autónomo del alumno	29	0%
AF5 Elaboración de trabajos (en grupo o individuales)	14	0%
AF6: Actividades de Evaluación	2	100%

## 6. DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
Introducción a los paradigmas de programación	Semanas 1-2
Programación reactiva	Semanas 3-8
Programación funcional	Semanas 9-15

## 7. SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
SE1 Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	0%	30%
SE2 Evaluación de trabajos, proyectos, informes, memorias	30%	80%
SE3 Prueba Objetiva	10%	60%

## 8. CRITERIOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN RESPECTO A LA CALIFICACIÓN FINAL (%)
Prácticas	80%
Examen final	20%

- Desarrollaremos dos prácticas (una por paradigma), que irán acompañadas de una memoria explicativa, y deberán defenderse con una defensa oral mediante Blackboard.
- Para poder superar satisfactoriamente la asignatura, será necesario que el alumno obtenga un 4 en cada práctica y en el examen final. No se hará media si no se cumplen todos los requisitos. La media total debe superar el 5.
- En caso de no conseguir el aprobado en la convocatoria ordinaria, el alumno deberá volver a presentarse a las partes suspensas, manteniendo los mismos criterios de evaluación para la convocatoria extraordinaria.
- No se conservarán calificaciones de ningún tipo entre distintos cursos académicos.
- No está permitido consumir bebidas ni comidas en el aula. Tampoco está permitida la presencia de cualquier tipo de bebida en las mesas, incluso en envases cerrados.
- Se demandará del alumno una participación activa.
- Se exigirá al alumno un buen comportamiento en todo momento durante el desarrollo de las clases. El mal comportamiento que impida el normal desarrollo de la clase puede conllevar la expulsión del aula por un tiempo a determinar por el profesor.

## 9. BIBLIOGRAFÍA / WEBGRAFÍA

<b>Básica:</b>
Roland Kuhn, Brian Hanafée, Jamie Allen. Reactive Design Patterns. Manning Publications 2017 Amit Rathore. Clojure in Action: Elegant Applications on the JVM. Manning Publications 2011
<b>Recomendada:</b>
Brian Marick. Functional Programming for the Object-Oriented Programmer. 2015.

## 10. MATERIAL, SOFTWARE Y HERRAMIENTAS NECESARIAS

<b>MATERIALES:</b>
- Ordenador personal capaz de ejecutar Python 3.6 - Clojure