



Universidad  
de Alcalá

# GUÍA DOCENTE

## Programación Avanzada

**Grado en**  
Ingeniería en Tecnologías de Telecomunicación (GITT)

**Universidad de Alcalá**

---

**Curso Académico 2022/2023**

4º Curso - 1<sup>er</sup> Cuatrimestre (GITT)

# GUÍA DOCENTE

Nombre de la asignatura:	<b>Programación Avanzada</b>
Código:	<b>350041 (GITT)</b>
Titulación en la que se imparte:	<b>Grado en Ingeniería en Tecnologías de Telecomunicación (GITT)</b>
Departamento y Área de Conocimiento:	<b>Automática Arquitectura y Tecnología de Computadores</b>
Carácter:	<b>Optativa (Especialidad) (GITT)</b>
Créditos ECTS:	<b>6.0</b>
Curso y cuatrimestre:	<b>4º Curso - 1º Cuatrimestre (GITT)</b>
Profesorado:	María del Mar Lendínez Chica
Horario de Tutoría:	Se informará el primer día de clase
Idioma en el que se imparte:	Español

## 1a. PRESENTACIÓN

Esta asignatura aborda un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de ordenador. El resultado es lo que conocemos como programación orientada a objetos (POO).

Se recomienda haber cursado las asignaturas de Sistemas Informáticos y Programación.

## 1b. COURSE SUMMARY

This course tackles the programming paradigm which uses objects and their interactions to design application software. This approach is known as Oriented-Object Programming (OOP). It is highly recommended that applicants have completed both courses: Information Systems and Computer Programming prior to attending this class.

## 2. COMPETENCIAS

### Competencias de Carácter Profesional

Esta asignatura proporciona la(s) siguiente(s) competencia(s) de carácter profesional definida(s) en el apartado 5 del Anexo de la Orden CIN/352/2009:

**CTE7** - Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas

### Resultados de aprendizaje

Al terminar con éxito esta asignatura/enseñanza, los estudiantes serán capaces de:

- RA1.** Describir las diferencias entre el paradigma de programación orientada a objetos y otros.
- RA2.** Diseñar, a partir de la especificación textual de un problema, un conjunto de clases relacionadas entre sí de cuya interacción se obtenga una solución a dicho problema.
- RA3.** Construir programas orientados a objetos utilizando un lenguaje de programación específico.

## 3. CONTENIDOS

Bloques de contenido	Total de clases, créditos u horas
Conceptos generales sobre la POO	8 T + 8 P horas
Clases y sobrecarga de operadores	14 T + 14 P horas
Programación genérica	2 T + 2 P horas
Excepciones y flujos	4 T + 4 P horas

## Cronograma

Semana / Sesión	Contenido
01 <sup>a</sup>	Introducción y conceptos más relevantes de Programación Orientada a Objetos. Pensar en objetos. Clases y objetos. Mensajes y métodos. Diseño de una clase de objetos. Constructores. Herencia.
02 <sup>a</sup>	Qué aporta C++ a los programadores de C. Elementos del lenguaje. Operadores C++. Ejercicios. Práctica: Asignación dinámica de memoria.
03 <sup>a</sup>	Parámetros por omisión. Funciones en línea. Funciones sobrecargadas. Operadores sobrecargados. Referencias. Espacios de nombres: directriz using. Ejercicios. Práctica: Asignación dinámica de memoria.
04 <sup>a</sup>	Flujos de entrada salida estándar. Contenedores: vector, map y string. Excepciones. Los operadores new y delete. Ejercicios. Práctica. Contenedores.
05 <sup>a</sup>	Clases. Definición de una clase. Atributos. Métodos de una clase. Control de acceso a los miembros de la clase. Clases en ficheros de cabecera. Métodos sobrecargados y métodos con parámetros con valores por omisión. El puntero implícito this. Métodos y objetos constantes. Práctica. Clases: atributos y métodos.
06 <sup>a</sup>	Iniciación de un objeto: constructor, asignación de objetos, constructor copia. Destrucción de objetos. Punteros como atributos de una clase. Miembros static de una clase. Atributos que son objetos. Clases internas. Integridad de los datos. Devolver un puntero o una referencia. Matrices de objetos. Funciones amigas de una clase. Punteros a miembros de una clase. Ejercicios. Práctica. Clases: constructores, destructores y matrices de objetos.
07 <sup>a</sup>	Sobrecargar un operador. Sobrecarga de operadores binarios. Sobrecarga de operadores unarios. Incremento y decremento. Operadores unarios/binarios. Ejercicios. Práctica: Sobrecarga de operadores.
08 <sup>a</sup>	Conversión de tipos definidos por el usuario: conversión mediante constructores, operadores de conversión. Asignación. Indexación. Llamada a función. Desreferencia. Sobrecarga de los operadores new y delete. Ejercicios. Práctica: Operadores de conversión.
09 <sup>a</sup>	Clases derivadas y herencia. Definir una clase derivada. Control de acceso a los miembros de las clases. Qué miembros hereda una clase derivada. Atributos con el mismo nombre. Redefinir métodos de la clase base. Constructores de clases derivadas. Copia de objetos. Destructores de clases derivadas. Jerarquía de clases. Ejercicios. Práctica: Clases derivadas.

10 <sup>a</sup>	<p>Funciones amigas. Punteros y referencias. Conversiones implícitas. Conversiones explícitas. Métodos virtuales: constructores virtuales y destructores virtuales. Ejercicios.</p> <p>Práctica. Clases derivadas: punteros y referencias.</p>
11 <sup>a</sup>	<p>Información de tipos durante la ejecución. Polimorfismo. Clases abstractas. Herencia múltiple: clases base virtuales, redefinición de métodos de clases base virtuales, conversiones entre clases.</p> <p>Práctica. Clases derivadas: polimorfismo.</p>
12 <sup>a</sup>	<p>Programación genérica. Definición de una plantilla. Funciones genéricas. Clases genéricas.</p> <p>Práctica: Clases genéricas.</p>
13 <sup>a</sup>	<p>Excepciones: manejo y creación. Excepciones no esperadas. Flujo de ejecución. Gestión de recursos.</p> <p>Práctica: Excepciones.</p>
14 <sup>a</sup>	<p>Flujos. Visión general de un fichero. Descripción de los búferes y flujos. Acceso secuencial. Acceso aleatorio.</p> <p>Práctica: Flujos de E/S.</p>

## 4. METODOLOGÍAS DE ENSEÑANZA APRENDIZAJE. ACTIVIDADES FORMATIVAS

### 4.1. Distribución de créditos

Número de horas presenciales:	58 horas (56 horas de clase presencial +2 horas de evaluación)
Número de horas del trabajo propio del estudiante:	92 horas (Incluye horas de estudio, elaboración de actividades, preparación de exámenes)
Total horas	150 horas

### 4.2. Estrategias metodológicas, materiales y recursos didácticos

Exposición teórica con diapositivas y demostraciones con el ordenador.

Resolución de problemas en grupo y resolución de prácticas individuales y trabajos por cada alumno.

Exposiciones de los alumnos.

## 5. EVALUACIÓN: Procedimientos, criterios de evaluación y calificación

Preferentemente se ofrecerá a los alumnos un sistema de evaluación continua que tenga características de evaluación formativa de manera que sirva de realimentación en el proceso de enseñanza-aprendizaje por parte del alumno.

## 5.1. PROCEDIMIENTOS

La evaluación debe estar inspirada en los criterios de evaluación continua (Normativa de Evaluación de los Aprendizajes, NEA, art 3). No obstante, respetando la normativa de la Universidad de Alcalá se pone a disposición del alumno un proceso alternativo de evaluación final de acuerdo a la Normativa de Evaluación de los Aprendizaje (aprobada en Consejo de Gobierno de 24 de marzo de 2011 y modificada en Consejo de Gobierno de 5 de mayo de 2016) según lo indicado en su Artículo 10, los alumnos tendrán un plazo de quince días desde el inicio del curso para solicitar por escrito al Director de la Escuela Politécnica Superior su intención de acogerse al modelo de evaluación no continua aduciendo las razones que estimen convenientes. La evaluación del proceso de aprendizaje de todos los alumnos que no cursen solicitud al respecto o vean denegada la misma se realizará, por defecto, de acuerdo al modelo de evaluación continua. El estudiante dispone de dos convocatorias para superar la asignatura, una ordinaria y otra extraordinaria.

## 5.2. EVALUACIÓN

### CRITERIOS DE EVALUACIÓN

Los criterios de evaluación deben atender al grado de adquisición de las competencias por parte del estudiante. Para ello se definen los siguientes:

- CE1.** El alumno identifica los conceptos e ideas principales de la programación orientada a objetos.
- CE2.** El alumno asocia y aplica los conceptos e ideas principales de las clases, sobrecarga de operadores, clases derivadas y polimorfismo.
- CE3.** El alumno distingue y aplica los conceptos e ideas principales de la programación genérica.
- CE4.** El alumno entiende y aplica los conceptos e ideas principales de las excepciones y flujos.
- CE5.** El alumno integra y aplica los contenidos a problemas orientados a objetos desarrollados bajo el lenguaje C++.

### INSTRUMENTOS DE EVALUACIÓN

A continuación, se especifican los instrumentos de calificación para las convocatorias ordinaria y extraordinaria.

[Convocatoria ordinaria: evaluación continua / examen final](#)

La evaluación continua constará de dos partes:

1. Prácticas y/o exámenes intermedios: **60%** de la nota final.

El total de las actividades planteadas en este bloque se comunicarán al alumno durante la clase de presentación. Dichas actividades comprenden:

Realización y evaluación de ejercicios prácticos.

Evaluación de conceptos teóricos y problemas planteados sobre los mismos.

Están previstas 5 pruebas de evaluación parcial (**PEI**) del total de las actividades planteadas. La duración no será superior a 2 horas por cada una de ellas. Estas pruebas se realizarán en horas de prácticas o de teoría, por lo que no requerirán tiempo extra.

2. Realización de una prueba examen final (**PEF**) teórico-práctico a la finalización del periodo docente: **40%** de la nota final. La duración de este examen será más o menos de 2 horas.

Para superar la asignatura, es necesario obtener una nota media mínima de 5 puntos sobre 10 entre todos los ejercicios prácticos. La evaluación de cada supuesto práctico incluye la práctica (**PL**) a realizar y la teoría (**T**) relacionada con la misma.

Además, el alumno deberá entregar los ejercicios prácticos (**PL**) en la fecha establecida para cada uno de ellos en un calendario proporcionado al comienzo del curso. La fecha límite para entregar un ejercicio práctico será justamente una semana después de la fecha de entrega establecida y, en este caso, se aplicará una penalización de un 50% sobre la nota obtenida en la evaluación del mismo. Los ejercicios entregados posteriormente a la fecha límite se calificarán con una nota de 0 puntos. Las entregas de las prácticas, salvo que, de forma expresa, se indique lo contrario se realizarán exclusivamente dentro del grupo de laboratorio asignado al alumno.

Se considerará como no presentado a aquel alumno que no haya realizado ninguna de las pruebas parciales asociadas a la evaluación de las prácticas de laboratorio.

#### Evaluación mediante examen final:

El examen final constará de una única prueba acerca de los contenidos teóricos-prácticos que constituirá el 100% de la nota de la asignatura.

Tendrán derecho a realizar este examen aquellos alumnos a los que se les haya concedido la evaluación mediante examen final. Dicho examen podrá realizarse de forma oral y/o escrita.

Para superar el examen final, los alumnos deberán entregar las prácticas, que aún no le hayan sido evaluadas, el día del examen final y aprobar la parte relacionada con la evaluación de las prácticas con una nota mínima de 5 puntos sobre 10. La evaluación de cada supuesto práctico incluye la práctica a realizar y la teoría relacionada con la misma.

Los contenidos teóricos-prácticos incluyen las prácticas del calendario de prácticas de la asignatura. Todas las prácticas entregadas fuera del calendario establecido llevarán la penalización expuesta en el punto 2 de la evaluación continua.

#### Convocatoria extraordinaria: examen final

Los alumnos que no hayan superado la asignatura en la convocatoria ordinaria, podrán optar por otra convocatoria extraordinaria que constará de una única prueba acerca de los contenidos teóricos-prácticos que constituirá el 100% de la nota de la asignatura. Dicho examen podrá realizarse de forma oral y/o escrita.

Para superar el examen final, los alumnos deberán entregar las prácticas el día del examen final y aprobar la parte relacionada con la evaluación de las prácticas con una nota mínima de 5 puntos sobre 10. La evaluación de cada supuesto práctico incluye la práctica a realizar y la teoría relacionada con la misma.

Los contenidos teóricos-prácticos incluyen las prácticas del calendario de prácticas de la asignatura.

#### Normativa Reguladora de los Procesos de Evaluación de los Aprendizajes:

Según cita la Normativa Reguladora de los Procesos de Evaluación de los Aprendizajes, aprobada en Consejo de Gobierno el 24 de marzo de 2011:

El plagio, entendido como la copia de textos sin citar su procedencia y dándolos como de elaboración propia, conllevará automáticamente la calificación de suspenso en la asignatura en la que se hubiera detectado. Esta consecuencia debe entenderse sin perjuicio de las responsabilidades disciplinarias en las que pudiesen incurrir los estudiantes que plagien.

## CRITERIOS DE CALIFICACIÓN

En la convocatoria ordinaria – evaluación continua la relación entre los criterios, instrumentos y calificación es la siguiente:

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CTE7	RA1	CE1, CE2, CE5	PEI1: PL1-PL2, T	15%
	RA2-RA3	CE2, CE5	PEI2: PL3-PL5, T	22,5%
			PEI3: PL6, T	7,5%
			PEI4: PL7, T	7,5%
		CE3, CE4, CE5	PEI5: PL8, T	7,5%
CE1-CE5	PEF: PL1-PL8, T	40%		

En la convocatoria ordinaria – examen final o en la convocatoria extraordinaria – examen final la relación entre los criterios, instrumentos y calificación es la siguiente:

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CTE7	RA1-RA3	CE1-CE5	PEF: PL1-PL8	50%
CTE7	RA1-RA3	CE1-CE5	PEF: T	50%

## 6. BIBLIOGRAFÍA

### 6.1. Bibliografía básica

- Programación orientada a objetos con C++, Ed. RA-MA. Autor: Fco. Javier Ceballos.

### 6.2. Bibliografía complementaria

- El lenguaje de programación C++. Stroustrup B.
- C++ Primer Plus. Stephen Prata.
- A tour of C++. Stroustrup B.
- Como programar en C/C++. Ed. Prentice Hall. H.M.Deitel & P.J.Deitel.
- Lenguajes de programación. Diseño e Implementación. Terence W.Pratt. Marvin V. Zelkowitz. Prentice Hall.

## **NOTA INFORMATIVA**

La Universidad de Alcalá garantiza a sus estudiantes que, si por exigencias sanitarias las autoridades competentes impidieran la presencialidad total o parcial de la actividad docente, los planes docentes alcanzarían sus objetivos a través de una metodología de enseñanza-aprendizaje y evaluación en formato online, que retornaría a la modalidad presencial en cuanto cesaran dichos impedimentos.