



Universidad
de Alcalá

GUÍA DOCENTE

Programación Avanzada

Grado en
Ingeniería en Tecnologías de Telecomunicación (GITT)

Universidad de Alcalá

Curso Académico 2025/2026

4º Curso - 1^{er} Cuatrimestre (GITT)

GUÍA DOCENTE

Nombre de la asignatura:	Programación Avanzada
Código:	350041 (GITT)
Titulación en la que se imparte:	Grado en Ingeniería en Tecnologías de Telecomunicación (GITT)
Departamento y Área de Conocimiento:	Automática Arquitectura y Tecnología de Computadores
Carácter:	Optativa (Especialidad) (GITT)
Créditos ECTS:	6.0
Curso y cuatrimestre:	4º Curso - 1º Cuatrimestre (GITT)
Profesorado:	Maria Dolores Rodríguez Moreno
Horario de Tutoría:	Se informará el primer día de clase
Idioma en el que se imparte:	Español/English Friendly

1a. PRESENTACIÓN

El desarrollo de software requiere enfoques que faciliten la creación de sistemas complejos, reutilizables y fácilmente mantenibles. Para ello, es fundamental dominar modelos de programación que permitan estructurar el código de manera modular, favoreciendo la abstracción y la encapsulación. Entre estos modelos destaca el paradigma de programación orientado a objetos, ampliamente utilizado en el ámbito profesional y académico.

Esta asignatura profundiza en el diseño y construcción de aplicaciones utilizando dicho paradigma, partiendo de los conocimientos adquiridos en asignaturas previas como Sistemas Informáticos y Programación. A través de una aproximación teórico-práctica, el estudiantado desarrollará la capacidad de analizar problemas, modelar soluciones mediante clases e implementar programas estructurados en torno a objetos y sus relaciones.

La asignatura se organiza en cuatro bloques temáticos, en los que se abordan tanto los fundamentos teóricos como la aplicación práctica de los conceptos:

1. Conceptos generales sobre la programación orientada a objetos, donde se presentan las diferencias entre este paradigma y la programación estructurada, así como los elementos fundamentales del lenguaje orientado a objetos.
2. Clases, métodos y herencia, que profundiza en el diseño de clases, el uso de métodos y atributos, y los principios de encapsulación, herencia y polimorfismo. También se introducen técnicas como la organización del código mediante módulos.
3. Estructuras de datos y librerías, centrado en el uso de estructuras de datos dinámicas y flexibles, así como en la integración de bibliotecas externas y especializadas para ampliar la funcionalidad del software desarrollado.
4. Excepciones y ficheros, donde se abordan los mecanismos para el tratamiento de errores mediante excepciones, incluyendo la creación de excepciones personalizadas, y se exploran las operaciones de entrada/salida con ficheros y bases de datos.

1b. COURSE SUMMARY

Software development requires approaches that facilitate the creation of complex, reusable, and easily maintainable systems. To this end, it is essential to master programming models that allow for modular code organization and promote abstraction and encapsulation. Among these models, the object-oriented programming paradigm is widely used in both professional and academic settings.

This course focuses on the design and construction of applications using this paradigm, building on the knowledge acquired in previous courses such as Computer Systems and Programming. Through a theoretical and practical approach, students will develop the ability to analyze problems, model solutions using classes, and implement structured programs based on objects and their interactions.

The course is organized into four thematic blocks, covering both the theoretical foundations and the practical application of the concepts:

1. General concepts of object-oriented programming, presenting the differences between this paradigm and structured programming, as well as the fundamental elements of the object-oriented language.
2. Classes, methods, and inheritance, focusing on class design, use of methods and attributes, and the principles of encapsulation, inheritance, and polymorphism. Code organization through modules is also introduced.
3. Data structures and libraries are centered on the use of dynamic and flexible data structures, along with the integration of external and specialized libraries to extend the functionality of the developed

software.

4. Exceptions and files, addressing mechanisms for error handling through exceptions, including the creation of custom exceptions, and exploring input/output operations with files and databases.

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 Diferencias entre POO y programación estructurada. Introducción a los elementos de la POO. Elementos del lenguaje.	6 T + 6 P horas
Clases, métodos y herencia Concepto de clases, objetos, métodos y atributos. Encapsulación, herencia y polimorfismo. Métodos especiales. Módulos.	12 T + 12 P horas
Estructuras y Librerías Estructuras de datos flexibles. Librerías externas y especializadas.	6 T + 6 P horas
Excepciones y ficheros Conceptos de excepciones. Creación de excepciones personalizadas. Trabajar con ficheros y bases de datos.	4 T + 4 P horas

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

Clases Presenciales	<ul style="list-style-type: none"> • Presentación y/o revisión de conceptos prácticos. • Resolución de problemas. • Sesiones prácticas de laboratorio: orientadas a consolidar los conceptos explicados en clase, mediante la implementación de programas y el uso de herramientas de desarrollo. • Ejecución y depuración de código. • Actividades prácticas en grupo.
Tutorías individuales, grupales y vía web (foro, correo, etc.)	<ul style="list-style-type: none"> • Resolución de dudas. • Revisión de prácticas y seguimiento individualizado del progreso. • Apoyo al aprendizaje autónomo.
Trabajo autónomo	<ul style="list-style-type: none"> • Lecturas complementarias y revisión de materiales de clase. • Uso de modelos de lenguaje (LLMs) como herramienta de apoyo para comprender conceptos, depurar código y generar soluciones orientadas a objetos.

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 Aprendizajes](#) 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 alumnado. Para ello se definen los siguientes:

- CE1.** Identifica y comprende los conceptos e ideas principales de la programación orientada a objetos, diferenciándolos de otros enfoques de programación.
- CE2.** Diseña e implementa clases, instancias y métodos aplicando los principios de encapsulación.
- CE3.** Emplea mecanismos de herencia y polimorfismo para estructurar jerarquías de clases y reutilizar código.
- CE4.** Distingue y aplica los conceptos e ideas principales de estructuras de datos, adaptando su elección al contexto del programa.
- CE5.** Organiza el código de manera modular, haciendo uso adecuado de funciones, módulos y librerías.
- CE6.** Aplica métodos especiales para extender el comportamiento de las clases de forma controlada y coherente.
- CE7.** Gestiona errores mediante excepciones, incluyendo la creación de excepciones personalizadas, y trabaja con flujos de entrada/salida y bases de datos.
- CE8.** Integra de forma efectiva los conceptos aprendidos para desarrollar aplicaciones funcionales, estructuradas y mantenibles.

INSTRUMENTOS DE EVALUACIÓN

Esta sección especifica los instrumentos de evaluación que serán aplicados a cada uno de los Criterios de Evaluación.

- Pruebas de Laboratorio (PRL): Examen en PC o en papel sobre contenidos de las prácticas. Tendrá un peso del 30% en la calificación final.
- Prueba de evaluación Intermedia (PEI): La prueba de Evaluación Intermedia tiene un peso del 30% en la calificación final.
- Prueba de Evaluación Final (PEF): La Prueba de Evaluación Final tiene un peso del 40% en la calificación final.
- Prácticas de laboratorio (PL): Actividades prácticas que el estudiantado debe desarrollar y entregar dentro del calendario establecido. Podrán incluir ejercicios de codificación, resolución de problemas, uso de librerías o pruebas funcionales.

Convocatoria ordinaria: evaluación continua / examen final

La evaluación continua constará de dos partes:

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

El total de las actividades planteadas en este bloque se comunicarán al alumnado durante la clase de presentación y comprenden:

* Una prueba de evaluación intermedia (**PEI**).

* Prueba de laboratorio (**PRL**).

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.

Además, el alumno deberá entregar los ejercicios prácticos (**PL**) en la fecha establecida en el calendario proporcionado al comienzo del curso. Es condición necesaria entregar las **PLs** para hacer la **PRL**.

Se considerará como no presentado al alumnado que no haya presentado ninguna práctica ni haya

realizado ninguna prueba.

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 según fija la normativa de la UAH. Dicho examen podrá realizarse de forma oral y/o escrita.

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

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.

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

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
CTE1, CTE4, CTE7	RA1-RA3	CE1, CE2, CE3, CE4, CE8	PEI	30%
CTE1, CTE4, CTE7	RA1-RA3	CE2, CE3, CE4, CE5, CE6, CE7, CE8	PRL	30%
CTE1, CTE4, CTE7	RA1-RA3	CE1 - CE8	PEF	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
CTE1, CTE4, CTE7	RA1-RA3	CE1-CE8	PEF	100%

La metodología de enseñanza-aprendizaje y el proceso de evaluación se ajustarán cuando sea necesario, con las orientaciones de la Unidad de Atención a la Diversidad, para aplicar adaptaciones curriculares al estudiantado con necesidades específicas.

6. BIBLIOGRAFÍA

6.1. Bibliografía básica

- Object Oriented Programming with Python for Beginners: Mastering the Foundations of OOP. From Principles to Practice. 2 in 1 Guide. Autor: SAM CAMPBELL.
- Python Tutorial - Tapa blanda. GuidoVan Rossum (2012).

6.2. Bibliografía complementaria

- Lenguajes de programación. Diseño e Implementación. Terence W.Pratt. Marvin V. Zelkowitz. Prentice Hall.

NOTA INFORMATIVA

Durante el desarrollo de las pruebas de evaluación han de seguirse las pautas marcadas en el Reglamento por el que se establecen las Normas de Convivencia de la Universidad de Alcalá, así como las posibles implicaciones de las irregularidades cometidas durante dichas pruebas, incluyendo las consecuencias por cometer fraude académico según el Reglamento de Régimen Disciplinario del Estudiantado de la Universidad de Alcalá.