



## GUÍA DOCENTE

Fundamentos de Programación

# GRADO EN INGENIERÍA DE VIDEOJUEGOS

**MODALIDAD: PRESENCIAL**

**CURSO ACADÉMICO: 2025-2026**

## DESCRIPCIÓN DE LA ASIGNATURA

Denominación de la asignatura:	<b>Fundamentos de Programación</b>
Titulación:	Grado en Ingeniería de Videojuegos
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Fundamentos Informáticos
Curso:	1
Cuatrimestre:	1
Carácter:	B
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	carlos.vazquez@ext.live.u-tad.com
Página Web:	<a href="http://www.u-tad.com/">http://www.u-tad.com/</a>

### Descripción de la materia

Esta materia aborda los fundamentos de la programación, la arquitectura de las computadoras, las redes, los sistemas operativos y las bases de datos. Los estudiantes aprenderán cómo funcionan las computadoras, cómo se comunican entre sí y cómo almacenan y recuperan información.

### Descripción de la asignatura

Esta asignatura introduce al estudiante en los fundamentos de la programación estructurada y orientada a objetos utilizando el lenguaje C++. Se abordan conceptos esenciales como tipos de datos, estructuras de control, funciones, arrays, punteros, manejo de memoria dinámica y clases. Además, se promueve el desarrollo de habilidades para la resolución de problemas mediante la implementación de algoritmos y programas eficientes y bien estructurados. El objetivo final es que el alumnado sea capaz de desarrollar programas básicos en C++ aplicando buenas prácticas de programación y principios de diseño modular.

## RESULTADOS DE APRENDIZAJE DE LA MATERIA (CONOCIMIENTOS, HABILIDADES Y COMPETENCIAS)

K1 Conocer los principios y paradigmas de los lenguajes de programación atendiendo a las estructuras de control, variables, sintaxis de programación y gestión del uso de la memoria de manera eficaz.

K2 Entender los procesos asociados a la ingeniería del software para el desarrollo de aplicaciones.

K3 Entender el funcionamiento tanto a nivel software como nivel hardware de los diferentes dispositivos tecnológicos donde se va a ejecutar el videojuego

K7 Conocer la estructura de las computadoras, los conceptos de codificación, manipulación, tratamiento de la información y lenguajes de bajo nivel.

K8 Diseñar esquemas de bases de datos normalizados utilizando modelos de entidad-relación y álgebra relacional, así como realizar consultas en lenguajes procedurales.

K9 Conocer las fases del diseño e implementación de un plan de pruebas para el testeo de una aplicación informática.

K14 Comprender el funcionamiento de las redes de ordenadores, las tecnologías de comunicación así como las distintas topologías y sus protocolos de comunicación.

K16 Conocer las diferentes fases del ciclo de vida del desarrollo de una aplicación software, entendiendo las necesidades de documentación y gestión.

S1 Resolver problemas de tratamiento y gestión de datos que se plantean en el desarrollo de videojuegos.

S2 Elaborar diagramas UML y documentos de diseño software en base a las especificaciones recibidas en el documento de diseño de juego.

S5 Testar el funcionamiento y la funcionalidad de una aplicación informática, a través de planes de pruebas y técnicas de diseño.

S12 Configurar los diferentes equipos de red en arquitecturas cliente-servidor en aplicaciones de videojuegos multijugador.

s13 Configurar un sistema operativo a través del manejo de los comandos de usuario.

## CONTENIDO

Programación estructurada en lenguajes de programación como C.

Sintaxis de programación, tipos de datos, funciones y modularidad.

Uso de bibliotecas estándar.

## TEMARIO

Unidad 1: Introducción a la programación.

- Lenguajes de programación.
- Historia y evolución.
- Paradigmas de programación.
- Lenguajes compilador vs. interpretados.
- Entornos de desarrollo y compiladores.
- Introducción a la programación Estructurada y Orientada a Objetos.

Unidad 2: Sintaxis del lenguaje.

Unidad 3: Tipos de datos básicos y complejos.

- Tipos básicos: int, float, double, char, etc.
- Tipos compuestos: structs, enum, union, arrays.
- Ámbito de declaración y uso.

Unidad 4: Funciones y métodos (modularidad).

- Sintaxis.
- Uso, unicidad y reutilización.

Unidad 5: Estructuras básicas de datos.

- Vectores, listas, colas, pilas.
- Uso de Librerías.
- Punteros y referencias.
- Características avanzadas de C++ 20.

Unidad 6: Programación Orientada a objetos.

- Nociones básicas y sintaxis.
- Abstracción, encapsulamiento, herencia y polimorfismo.
- Clases e instancias.

## ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

### Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
<i>Lección magistral</i>	22	22
<i>Resolución de ejercicios</i>	36	36

<i>Estudio independiente y trabajo autónomo del alumno</i>	36	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	54	0
<i>Actividades de Evaluación</i>	2	2

### Metodologías docentes

M1 -Metodología clásica (lecciones magistrales)

M2 -Aprendizaje basado en problemas

M4 -Aprendizaje cooperativo

M5 -Aprendizaje por investigación(ABI)

M6 -Metodología aula invertida (Flipped classroom )

M7 -Gamificación

### DESARROLLO TEMPORAL

Unidad 1: 1 semana.

Unidad 2: 2 semanas.

Unidad 3: 2 semanas.

Unidad 4: 3 semanas.

Unidad 5: 4 semanas.

Unidad 6: 3 semanas.

### SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
-------------------------	---	---

<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	20	50
<i>Prueba Objetiva</i>	50	70

## CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	40	40
<i>Prueba Objetiva</i>	50	50

### Consideraciones generales acerca de la evaluación

Convocatoria ordinaria:

- Se realizarán varios ejercicios durante el curso en función de los contenidos vistos, los cuales suman un total del 40% de la asignatura. Al final del curso se realizará una prueba objetiva que supondrá un 50% de la nota final. La participación en clase/evolución del alumno será un 10%.
- La prueba objetiva consiste en desarrollar un ejercicio/examen de manera presencial.

Convocatoria Extraordinaria:

- Para aprobar la asignatura en la convocatoria extraordinaria, se requerirá la presentación de todos, o algunos (a criterio del profesor) los trabajos finales suspensos, o no entregados, y la superación de la consiguiente prueba objetiva. En extraordinaria las actividades valdrán 40% y la prueba objetiva un 50%.
- La entrega de cualquier ejercicio o práctica fuera de plazo será penalizada.
- No se permitirá el uso de móviles ni de sistemas de servicios de juego como Steam, y similares, durante las clases.

- No se permite el uso de Smartwatches o de móviles durante los exámenes. Dichos aparatos tendrán que estar guardados y fuera de la vista del alumno durante la realización del examen.
- Toda detección de plagio, copia o uso de malas prácticas (como puede ser el uso de IAs) en un trabajo o examen implicará el suspenso de este trabajo con un cero, el reporte al Claustro y coordinador académico y la aplicación de la normativa vigente, lo que puede conllevar penalizaciones muy serias para el alumno.

## **BIBLIOGRAFÍA / WEBGRAFÍA**

STROUSTRUP, Bjarne (2014), Programming: Principles and Practice Using C++ (2ª Edición).

LIPPMAN Stanley, LAJOIE Josée y MOO Barbara (2012), C++ Primer (5ª Edición).

MEYERS, Scott (2014), Effective Modern C++, O'Reilly Media, Incorporated. - (chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://ananyapam7.github.io/resources/C++/Scott\_Meyers\_Effective\_Modern\_C++.pdf)

D'ANGELO Giuseppe & Čukić Ivan, The Practical Programmer's Guide to C++20 (KDAB) - (<https://www.kdab.com/the-practical-programmers-guide-to-cpp20/>)

## **MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS**

### **Tipología del aula**

Aula con ordenador y pizarra.

### **Materiales:**

Ordenador, Visual Studio (especificado por el profesor)

### **Software:**