

---

Planificación de la Docencia Universitaria  
**Grado en Diseño y Desarrollo de Videojuegos**

---

Guía Docente

Curso Académico 2020/2021

# Programación Orientada a Objetos (C#)

---

---

## Datos de identificación de la asignatura

### Título

Grado en Diseño y Desarrollo de Videojuegos

### Tipo de asignatura (básica, obligatoria u optativa)

Obligatoria

### Módulo

Ciencias Aplicadas y Tecnología

### Créditos ECTS

5

### Denominación de la asignatura

Programación Orientada a Objetos

### Modalidad/es de enseñanza

Presencial

### Código

40013

### Profesor

Sandra Garrido Romero

### Curso

Segundo

### Lengua vehicular

Español

### Semestre

Primero

---

## Profesorado de la asignatura

### Profesor

Sandra Garrido Romero

### Tutorías académicas

Consultar en el Campus Virtual el documento "Horarios de Tutorías - Grado en Diseño y Desarrollo de Videojuegos".

### Datos de contacto

[sandra.garrido@esne.es](mailto:sandra.garrido@esne.es)

## Requisitos previos

### Esenciales

Conocimientos básicos de programación.

### Aconsejables

Los propios del título. Asignaturas “Fundamentos de la Programación” y “Programación II”.

---

## Sentido y aportaciones de la asignatura al plan de estudios

### Campo de conocimiento al que pertenece la asignatura

Esta asignatura pertenece a la rama de Ingeniería y Arquitectura, áreas de conocimiento de Lenguajes y Sistemas Informáticos, y Ciencias de la Computación. Pertenece al Módulo de Ciencias Aplicadas y Tecnología del título.

### Relación de interdisciplinariedad con otras asignaturas del currículum

El objetivo de la asignatura de Programación Orientada a Objetos es que los alumnos aprendan y trabajen con el paradigma de la programación orientada a objetos moderna. La asignatura cuenta con un fuerte componente práctico donde la programación será un punto fundamental en el desarrollo de los proyectos. Esta asignatura se relaciona de manera directa con las asignaturas de primer curso “Fundamentos de la Programación” y “Programación II”, con los contenidos de segundo semestre de la asignatura de segundo curso “Animaciones y

Scripting”, y con las asignaturas de tercer curso “Desarrollo para Dispositivos Móviles” e “Ingeniería del Conocimiento: IA”.

Se utiliza C# como lenguaje de programación orientado a objetos avanzado, introducido en las asignaturas de programación de primer curso y utilizado también en el segundo semestre de la asignatura “Animaciones y Scripting” donde se usará la herramienta Unity donde C# se usa como lenguaje de desarrollo.

### Aportaciones al plan de estudios e interés profesional de la asignatura

Gracias a esta asignatura el alumno será capaz de plantear eficazmente desarrollos de programación tanto individualmente como en equipo, anticipando las distintas fases del mismo y pudiendo utilizar lenguajes, técnicas, librerías, metodologías, normas y estándares de programación usados en la industria.

---

## Resultados de aprendizaje en relación con las competencias que desarrolla la materia

### Competencias generales

**CG7.** Conocerá metodologías, normas y estándares, técnicas y programas de uso específico, y será capaz de utilizarlos en el desarrollo de videojuegos.

### Competencias específicas

**CE7.** Entenderá el proceso de diseño y desarrollo de un videojuego como un proyecto multidisciplinar, y será capaz de plantear dicho proyecto en términos de trabajo en equipo. Asimismo, será capaz de idear proyectos de videojuegos, plasmando dichas ideas gráficamente y por escrito de forma estructurada, ordenada y comprensible.

**CE8.** Será capaz de prototipar un sistema a partir de un diseño.

**CE13.**Comprenderá en qué momento es adecuado utilizar un paradigma u otro de programación y sabrá aplicarlo correctamente mediante los elementos de desarrollo que manejará (frameworks, kits de desarrollo, etc.).

---

## Resultados de aprendizaje relacionados con la asignatura

- El alumno podrá aplicar las metodologías y técnicas estándares a sus proyectos, independientemente de cuál sea su naturaleza, ya que conocerá los aspectos básicos del desarrollo.
- Asimismo, será capaz de idear proyectos de videojuegos, plasmando dichas ideas gráficamente y por escrito de forma estructurada, ordenada y comprensible.
- El alumno podrá implementar software mediante el paradigma de programación orientada a objetos en un entorno gráfico.
- El alumno podrá aplicar con coherencia en el desarrollo de los videojuegos principios matemáticos y físicos, tanto en el diseño 2D como 3D, animación y programación.
- El alumno podrá decidir entre las distintas técnicas de desarrollo de software que conoce en función de las necesidades actuales de su proyecto.
- El alumno podrá identificar los distintos pasos del ciclo de un proyecto, empleando las técnicas adecuadas a cada uno de ellos, mejorando así la calidad final del software.

---

## Contenidos / Temario / Unidades didácticas

### Breve descripción de los contenidos

- Introducción a la programación orientada a objetos: La complejidad inherente al software. Crisis del software. Factores en la calidad del software. El papel de la abstracción. Orientación a objetos. Características. Modularidad. Diseño de módulos. Acoplamiento y cohesión. Tipos abstractos de datos
- Conceptos Fundamentales de Programación Orientada a Objetos (POO). Clases y objetos. Estructura interna de un objeto. Mensajes.
- Herencia y polimorfismo: Herencia. Ligadura dinámica. Polimorfismo. Clases abstractas. Interfaces.  
Control de errores y excepciones en C#: Utilidad de las excepciones. Empleo de las excepciones. Lanzamiento de excepciones. Captura y tratamiento de excepciones.
- Estructuras de datos en C#: Listas enlazadas. Pilas. Colas.

### Temario desarrollado

- **Tema 1. Introducción a la programación orientada a objetos**
  - La complejidad inherente al software.
  - Crisis del software. Factores en la calidad del software.
  - El papel de la abstracción. Orientación a objetos.
  - Características. Modularidad. Diseño de módulos. Acoplamiento y cohesión.
  - Tipos abstractos de datos.
- **Tema 2. Conceptos fundamentales de Programación Orientada a Objetos (POO)**
  - Clases y objetos. Estructura interna de un objeto.

- Mensajes. Relaciones de agregación.
  
- **Tema 3. Herencia y polimorfismo**
  - Herencia
  - Ligadura dinámica.
  - Polimorfismo.
  - Clases abstractas. Interfaces.
  
- **Tema 4. Control de errores y excepciones en C#**
  - Utilidad de las excepciones.
  - Empleo de las excepciones. Lanzamiento de excepciones.
  - Captura y tratamiento de excepciones.
  
- **Tema 4. Estructuras de datos en C#**
  - Listas enlazadas.
  - Pilas. Colas.

---

## Cronograma

Unidades Didácticas / Temas	Período Temporal
<b>Tema 1.</b> Introducción a la programación orientada a objetos.	Septiembre
<b>Tema 2.</b> Conceptos fundamentales de Programación Orientada a Objetos (POO)	Septiembre/Octubre
<b>Tema 3.</b> Herencia y polimorfismo	Octubre
<b>Tema 4.</b> Control de errores y excepciones en C#.	Noviembre
<b>Tema 5.</b> Estructuras de datos en C#	Diciembre/Enero

## Modalidades organizativas y métodos de enseñanza

Modalidad organizativa	Método de enseñanza	Competencias relacionadas	Horas		
			Presencial	Trabajo autónomo	Total
<p><b>Clases teóricas.</b>            Actividad formativa en el aula que, utilizando la metodología expositiva, prioriza la acción docente del profesor.</p>	<p><b>Exposición de los temas.</b>            Explicar planificación de la asignatura: programa, apuntes y bibliografía.</p> <p>Repasos al inicio de la clase. Resolución de dudas: temas y lecturas. Pruebas de evaluación.</p>	CG7, CE7, CE8, CE13	30	5	35
<p><b>Clases prácticas.</b>            Actividad formativa en el aula-taller que, bajo la guía del profesor, se ordena a la resolución individual o cooperativa de ejercicios y problemas o a la ejecución de trabajos técnicos o artísticos.</p>	<p><b>Resolución de ejercicios.</b>            Debates sobre los temas y especialmente sobre ejercicios y lecturas.</p> <p>Presentaciones.            Pruebas de evaluación.</p>	CG7, CE7, CE8, CE13	20	15	35

<p><b>Tutorías.</b> Actividad formativa fuera del aula que fomenta el aprendizaje autónomo, con el apoyo de la acción de guía y seguimiento por medio de un tutor.</p>	<p><b>Preparación de clase mediante lectura de los temas.</b> Planificación de debates y comentarios mediante la preparación de las lecturas. Resolución de ejercicios.</p> <p>Comentarios y resolución de dudas presencialmente o por correo electrónico.</p>	CG7, CE7, CE8, CE13	10	-	10
<p><b>Trabajo personal del alumno.</b> Actividad formativa fuera del aula que, sin una guía directa del profesor o tutor, fomenta el aprendizaje autónomo del alumno.</p>	<p>Lecturas: preparación y búsqueda de información complementaria. Estudio personal.</p> <p>Preparación de comentarios y debates. Tutorías libres y voluntarias.</p>	CG7, CE7, CE8, CE13	-	50	50

## Sistema de evaluación

### General

Actividades de Evaluación	Criterios de Evaluación	Valoración respecto a la Calificación Final
Prácticas grupales	<ul style="list-style-type: none"> <li>- Contenidos: cumplimiento de los requisitos indicados en el enunciado (25%)</li> <li>- Aplicación correcta del paradigma OO (30%)</li> <li>- Documentación (5%)</li> </ul>	60%
Exámenes / Pruebas objetivas	<ul style="list-style-type: none"> <li>- Examen al final del cuatrimestre.</li> </ul>	30%
Asistencia Participativa	<ul style="list-style-type: none"> <li>- Asistencia 80% clases y actitud activa.</li> </ul>	10%

### Consideraciones generales acerca de la evaluación

#### Asistencia a Clase

- La asistencia a clase es obligatoria. Se aplica la norma del 80% de asistencia recogida en la Normativa Académica disponible en la pestaña de Documentos de Interés General del Campus Virtual.
- La Dirección/Coordinación de la Titulación podrá considerar situaciones excepcionales, previo informe documental, debiendo ser aprobadas por la Dirección Académica de ESNE.

- Se exigirá puntualidad al alumno en el comienzo de las clases. Una vez transcurridos cinco minutos de cortesía, el profesor podrá denegar la entrada en el aula.

### Entregas de Trabajos

- Los alumnos deben presentar y aprobar todas las entregas que se les soliciten. La no entrega de un trabajo supondrá suspender la asignatura.
- Los trabajos deben entregarse en las fechas que solicite el profesor, no se admiten entregas posteriores. Si por causas excepcionales se acepta un trabajo fuera de plazo, la máxima calificación a obtener será 5.
- En los trabajos en grupo, la calificación será individual por cada alumno, atendiendo a criterios de conocimiento de la materia, esfuerzo, presentación, asistencia a tutorías, etc. Por tanto, miembros de un mismo grupo pueden tener calificaciones diferentes.

### Evaluación en Convocatoria Ordinaria

- Para superar la asignatura es necesario que la suma ponderada de todas las actividades de evaluación sea de al menos 5 puntos. Para realizar la suma ponderada, la nota de la parte de Exámenes/Pruebas objetivas debe sumar más de 3,5 puntos y la parte de Prácticas debe sumar más de 5 puntos.

### Alumnos de las ramas de Arte y Diseño

- Están obligados a asistir a la parte de C# y serán evaluados acorde al sistema de evaluación explicado en esta Guía Docente.
- Están exentos de la asistencia y del sistema de evaluación de la parte de la asignatura relativa al lenguaje C++ recogidos en la Guía Docente del profesor Pablo Nogueira. No obstante, aquellos alumnos de estas dos ramas que opten por realizar entregas en la parte de C++ podrán sumar los puntos obtenidos en ellas a su

calificación en la parte de C#, siempre que superen los mínimos establecidos en el enunciado de cada entrega. Los criterios de evaluación están detallados en la Guía Docente de la parte C++.

### Alumnos de la rama de Programación

- Deben asistir a las clases de C# y de C++ y serán evaluados acorde a los sistemas de evaluación de cada parte. El porcentaje en su calificación final será el siguiente:
  - Parte C#: 40% de la calificación final.
  - Parte C++: 60% de la calificación final.
- Ambas partes deben estar aprobadas conforme a los criterios establecidos y publicados en sus correspondientes guías.

### **Evaluación en Convocatoria Extraordinaria**

- En la evaluación extraordinaria, los alumnos deben volver a presentar los trabajos que no hayan sido aprobados en convocatoria ordinaria. En caso de no tener el examen con la nota mínima exigida entonces será necesario volver a realizarlo. También se podrá solicitar la realización de nuevas entregas. Se podrá requerir una defensa pública de todas las entregas.

---

## **Bibliografía / Webgrafía**

### **Bibliografía básica**

- Joyanes Aguilar L., (1998), Programación orientada a objetos. McGraw-Hill / Interamericana de España, S.A.
- Albahari J. y Albahari B., (2010), C# 4.0 Pocket Reference. O'Reilly.

## Bibliografía complementaria

- VVAA. Microsoft C#. Curso de programación. Ra – Ma.
- Arriola N., Cosio L., C# Guía Total del Programador. RedUsers.
- Wirfs-Brock R. y col. Designing Object-Oriented Software. Prentice Hall.
- Taylor D. Object-Oriented Information Systems: Planning and Implementations. John Wiley & Sons.