

# UD2: Programación Orientada a Objetos (POO).

JOSÉ CLIMENT PENADÉS

# Índice

- ▶ Ubicación y justificación
- ▶ Normativa
- ▶ Objetivos
- ▶ Competencias y criterios de calificación
- ▶ Contenidos
- ▶ Metodología
- ▶ Actividades
- ▶ Agrupamientos
- ▶ Sesiones
- ▶ Criterios de evaluación
- ▶ Evaluación
- ▶ Atención a la diversidad
- ▶ Recursos y bibliografía

# Ubicación y justificación

- ▶ Segunda unidad del modulo.
- ▶ 1º Evaluación, inicio del curso.
- ▶ 40 horas = 20 días (2 horas por día)
- ▶ RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
- ▶ RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
- ▶ Software propio industrial.
- ▶ Paradigma actual mas extendido

# Normativa

- ▶ Ley Orgánica de educación (2/2006)
- ▶ Ley Orgánica de Modificación de la LOE (3/2020)
- ▶ Ley Orgánica de la Formación Profesional (3/2022)
- ▶ Real decreto del calendario de implantación (278/2023)
- ▶ Instrucciones para el alumnado procedente de otros países(16/07/2019)
- ▶ Orden sobre la prevención de riesgos laborales (1/2014)
- ▶ Instrucciones de inicio de curso (08/08/2024)
- ▶ Real decreto de ordenación del Sistema de Formación Profesional.(659/2023)
- ▶ Título del CFGS DAM (450/2010)
- ▶ Actualización del CFGS DAM(405/2023)
- ▶ Modificación del CFGS DAM (500/2024)
- ▶ Currículo nacional del CFGS DAM (EDU/2000/2010)
- ▶ Currículo autonómico del CFGS DAM (58/2012)
- ▶ Decreto de equidad e inclusión(104/2018)
- ▶ Orden de respuesta educativa a la inclusión(20/2019)
- ▶ Decreto de igualdad y convivencia (195/2022)

# Objetivos

Objetivos	
OD1	Definir el concepto de clase y de objeto, identificando sus características fundamentales.
OD2	Describir y analizar los elementos que componen la estructura de un objeto (atributos, métodos y constructores).
OD3	Diseñar y declarar clases que incluyan atributos, métodos y constructores simples.
OD4	Instanciar objetos y utilizar sus métodos y propiedades.
OD5	Desarrollar programas orientados a objetos básicos utilizando un entorno de desarrollo integrado (IDE).
OD6	Utilizar librerías de objetos para aprovechar funcionalidades predefinidas.
OD7	Distinguir entre elementos estáticos y dinámicos.
OD8	Representar la estructura de clases mediante diagramas UML.

# Competencias, criterios de calificación

- ▶ Competencias pps:
  - ▶ C3: Establecer vías eficaces de relación profesional y comunicación con sus superiores, compañeros y subordinados, respetando la autonomía y competencias de las distintas personas.
  - ▶ C4: Mantener el espíritu de innovación y actualización en el ámbito de su trabajo para adaptarse a los cambios tecnológicos y organizativos de su entorno profesional.
- ▶ Criterios de evaluación:
  - ▶ 2a – 2i, 4a – 4f

# Contenidos

## 4. Desarrollo de clases.

4.1. Concepto de clase.

4.2. Estructura y miembros de una clase.

4.3. Creación de propiedades y atributos.

4.4. Creación de métodos.

4.5. Creación de constructores.

4.6. Codificación y edición de objetos de las clases desarrolladas.

4.7. Encapsulación y visibilidad.

## 4.8. Utilización de clases y objetos.

### 2. Utilización de objetos.

2.1. Elementos de la programación orientada a objetos.

2.2. Características de los objetos.

2.3. Estructura de los objetos.

2.4. Instanciación de objetos.

2.5. Utilización de métodos.

2.6. Utilización de propiedades.

2.8. Codificación de métodos estáticos.

2.9. Utilización de métodos estáticos.

2.10. Parámetros y valores devueltos.

2.11. Librerías de objetos.

2.12. Constructores.

2.13. Destrucción de objetos y liberación de memoria.

2.14. Codificación, edición y compilación de programas simples orientados a objetos.

# Metodología

- ▶ Práctica y motivadora
- ▶ Orientaciones metodológicas:
  - ▶ Conocimientos previos del alumnado
  - ▶ Aprendizaje significativo y funcional
  - ▶ Metodologías activas



# Actividades E/A

- ▶ Actividades de presentación
- ▶ Actividades de evaluación de conocimientos previos
- ▶ Actividades de desarrollo de contenidos
- ▶ Actividades de cierre o síntesis-resumen
- ▶ Actividades de diagnóstico
- ▶ Actividades de refuerzo y ampliación
- ▶ Actividades evaluables o de evaluación

# Agrupamientos

- ▶ Trabajo individual
- ▶ Trabajo cooperativo en grupos pequeños
- ▶ Trabajos grupo-clase

# Sesiones

Planificación de las sesiones	
S1 – S3	<ul style="list-style-type: none"><li>- Introducción al paradigma orientado a objetos: conceptos de clase y objeto, diferencias con la programación estructurada.</li></ul>
S4 – S6	<ul style="list-style-type: none"><li>- Explicación teórica sobre objetos, características, estructura, e instanciación.</li><li>- Ejemplos y ejercicios sobre objetos.</li></ul>
S7 – S9	<ul style="list-style-type: none"><li>- Explicación teórica sobre clases, métodos, atributos y propiedades.</li><li>- Ejemplos y ejercicios sobre clases, métodos, atributos y propiedades.</li></ul>
S10 – S11	<ul style="list-style-type: none"><li>- Introducción a los constructores/destructores: su función e importancia en la inicialización de objetos.</li><li>- Ejercicios con constructores y destructores.</li></ul>
S12 – S20	<ul style="list-style-type: none"><li>- Ejercicios prácticos: manipulación de clases y objetos.</li><li>- Ejercicios prácticos: de uso de constructores/destructores.</li><li>- Ejercicios combinando todos los conceptos aprendidos.</li><li>- Resolución de dudas.</li></ul>

# Sesiones

S21 S25	-	<ul style="list-style-type: none"><li>- Introducción al uso de métodos estáticos.</li><li>- Uso de librerías de objetos disponibles.</li><li>- Ejercicios prácticos.</li></ul>
S26 S30	-	<ul style="list-style-type: none"><li>- Explicación de encapsulación, parámetros y valores devueltos: definición de atributos y métodos, públicos/privados.</li><li>- Ejercicios de encapsulación.</li></ul>
S31 S34	-	<ul style="list-style-type: none"><li>- Representación gráfica de las clases: elaboración de diagramas UML básicos y discusión en grupo.</li></ul>
S35 S37	-	<ul style="list-style-type: none"><li>- Implementación de ejercicios integradores que combinen los conceptos aprendidos.</li></ul>
S38 S40	-	<ul style="list-style-type: none"><li>- Actividades de ampliación y refuerzo.</li><li>- Cuestionario sobre los conceptos vistos durante la unidad.</li><li>- Resolución de dudas.</li></ul>

# Criterios de evaluación

RA2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

- a) Se han identificado los fundamentos de la programación orientada a objetos.
- b) Se han escrito programas simples.
- c) Se han instanciado objetos a partir de clases predefinidas.
- d) Se han utilizado métodos y propiedades de los objetos.
- e) Se han escrito llamadas a métodos estáticos.
- f) Se han utilizado parámetros en la llamada a métodos.
- g) Se han incorporado y utilizado librerías de objetos.
- h) Se han utilizado constructores.
- i) Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

# Criterios de evaluación

RA4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

- a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- b) Se han definido clases.
- c) Se han definido propiedades y métodos.
- d) Se han creado constructores.
- e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
- f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.

# Evaluación

## ▶ Proceso de enseñanza

- ▶ Autoevaluación de la metodología
- ▶ A mitad de la unidad y al final
- ▶ Rubrica de autoevaluación y cuaderno del docente

## ▶ Proceso de aprendizaje

- ▶ Criterios de evaluación del RA
- ▶ Evaluación inicial, continua y final
- ▶ Rubricas de evaluación
- ▶ Actividades y observación

# Evaluación

- ▶ Actitud/trabajo diario 10%.
- ▶ Proyecto 30%.
- ▶ Prueba final 20%.
- ▶ Pruebas y actividades 40%.



# Atención a la diversidad

- ▶ Sesiones de refuerzo y ampliación
- ▶ Adaptaciones en acceso y tiempo según la normativa

# Recursos y bibliografía

- ▶ Usaremos el ordenador, proyector, pizarra, documentación y material de elaboración propia que el alumnado tendrá disponible en aulas.
- ▶ Además, usaremos un IDE, en este caso eclipse y un procesador de textos (LibreOffice Writer)

# Actividad 1

- ▶ **Actividad:** Modelado de objetos en una aplicación de gestión de reservas para una albufera
- ▶ **Contexto laboral:**
  - ▶ Trabajas en una empresa de desarrollo de software que colabora con una oficina de turismo de la Albufera de Valencia. Debes crear el modelo de datos inicial de una aplicación que permitirá a los visitantes reservar rutas en barca. Empezarás definiendo los objetos básicos del sistema.
- ▶ **Contenidos trabajados:**
  - ▶ 2.1 Elementos de la programación orientada a objetos
  - ▶ 2.2 Características de los objetos
  - ▶ 2.3 Estructura de los objetos
  - ▶ 2.4 Instanciación de objetos
  - ▶ 4.1 Concepto de clase
  - ▶ 4.2 Estructura y miembros de una clase
  - ▶ 4.5 Creación de constructores
- ▶ **Tarea:**
  - ▶ Crea una clase "ReservaBarca" con los atributos "nombreCliente", "fecha" y "numeroPersonas".
  - ▶ Incluye un constructor para inicializar los datos y un método que imprima un resumen de la reserva.

# Actividad 2

- ▶ **Actividad:** Métodos estáticos para describir tradiciones valencianas
- ▶ **Contexto laboral:**
  - ▶ Formas parte de un equipo de desarrollo que está creando una guía digital interactiva sobre las fiestas valencianas. La aplicación contiene descripciones estáticas de cada tradición.
- ▶ **Contenidos trabajados:**
  - ▶ 2.8 Codificación de métodos estáticos
  - ▶ 2.9 Utilización de métodos estáticos
- ▶ **Tarea:**
  - ▶ Crear una clase “FiestaValenciana” con métodos estáticos que impriman descripciones de distintas fiestas