

UD1: Fundamentos de Programación.

JOSÉ CLIMENT PENADÉS

Índice

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

Ubicación y justificación

- ▶ Primera unidad del modulo.
- ▶ 1º Evaluación, inicio del curso.
- ▶ 25 horas = 12'5 días (2 horas por día)
- ▶ RA 1: Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
- ▶ Software propio industrial.
- ▶ Necesidad de crear, actualizar y mantener software

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	Identificar los distintos bloques y estructuras básicas que integran un programa informático.
OD2	Reconocer y definir variables, constantes y literales en un entorno de programación.
OD3	Aplicar operadores y expresiones para solucionar problemas aritméticos y lógicos.
OD4	Comprender y utilizar conversiones de tipo, tanto implícitas como explícitas, para garantizar la integridad de los datos.
OD5	Emplear comentarios en el código para documentarlo y mejorar su legibilidad.
OD6	Representar algoritmos mediante pseudocódigo y diagramas UML.
OD7	Diseñar programas utilizando estructuras básicas de control.
OD8	Utilizar entornos de desarrollo integrados (IDE) para crear y gestionar proyectos.

Competencias

- ▶ Competencias pps:
 - ▶ C1: Desarrollar aplicaciones multiplataforma con acceso a bases de datos utilizando lenguajes, librerías y herramientas adecuados a las especificaciones.
 - ▶ 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.

Contenidos

Contenidos didácticos detallados

1. Identificación de los elementos de un programa informático.

1.1. Estructura y bloques fundamentales.

1.2. Soluciones y proyectos.

1.3. Utilización de los entornos integrados de desarrollo.

1.4. Variables.

1.5. Tipos de datos. Operaciones.

1.6. Literales.

1.7. Constantes.

1.8. Operadores y expresiones.

1.9. Conversiones de tipo.

1.10. Comentarios.

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

S1	<ul style="list-style-type: none">- Presentación del curso y cuestionario de evaluación inicial.- Inscripción en Aules.- Introducción a la programación (lenguajes, características, etc.)
S2 – S3	<ul style="list-style-type: none">- Debate y explicación de los bloques de un programa informático.- Actividades de identificación de los bloques de un programa informático.- Explicación de algoritmia.- Actividades de algoritmia.
S4 – S6	<ul style="list-style-type: none">- Introducción al pseudocódigo.- Actividades de algoritmia con pseudocódigo.
S7 – S9	<ul style="list-style-type: none">- Presentación y práctica de variables, constantes y literales: definición, declaración y asignación.- Explicación sobre soluciones y proyectos en un IDE.- Ejercicios interactivos en el IDE.

Sesiones

S10 S13	-	<ul style="list-style-type: none">- Explicación sobre tipos de datos, operaciones/operadores y expresiones.- Ejercicios sobre operadores aritméticos y lógicos y tipos de datos.- Introducción a conversiones de tipo (implícitas y explícitas).
S14 S17	-	<ul style="list-style-type: none">- Resolución de prácticas/problemas que integren operadores y conversiones.- Revisión de los ejercicios y resolución de dudas.
S18		<ul style="list-style-type: none">- Introducción a la documentación. Comentarios.- Ejercicios sobre comentarios.
S19 S23	-	<ul style="list-style-type: none">- Introducción a la representación gráfica de algoritmos: diagramas UML y diagramas de flujo.- Ejercicios grupales sobre el diseño diagramas.
S24 S25	-	<ul style="list-style-type: none">- Actividades de ampliación y refuerzo.- Cuestionario sobre los conceptos vistos durante la unidad.- Resolución de dudas.

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

Criterios de calificación: 1a – 1i

- a) Se han identificado los bloques que componen la estructura de un programa informático.
- b) Se han creado proyectos de desarrollo de aplicaciones.
- c) Se han utilizado entornos integrados de desarrollo.
- d) Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- e) Se ha modificado el código de un programa para crear y utilizar variables.
- f) Se han creado y utilizado constantes y literales.
- g) Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- h) Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.
- i) Se han introducido comentarios en el código.

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: Revisión de un código base en un equipo de desarrollo**
- ▶ **Contexto laboral:**
 - ▶ Eres parte de un equipo de desarrollo de software que trabaja en una aplicación para la gestión de tareas de oficina. Tu primer encargo es revisar un fragmento de código que ha dejado un compañero y agregarle comentarios para que el resto del equipo lo entienda.
- ▶ **Contenidos trabajados:**
 - ▶ 1.1 Estructura y bloques fundamentales
 - ▶ 1.2 Soluciones y proyectos
 - ▶ 1.3 Entornos de desarrollo
 - ▶ 1.10 Comentarios
- ▶ **Tarea:**
 - ▶ Copiar el código en su entorno
 - ▶ Ejecutarlo (aunque no haga nada visible, deben comprobar que no da error).
 - ▶ Agregar comentarios explicativos en cada línea como si fueran notas para un compañero que acaba de incorporarse al equipo

Actividad 2

- ▶ **Actividad: Cálculos y expresiones básicas en un módulo de nómina**
- ▶ **Contexto laboral:**
 - ▶ Formas parte del equipo que desarrolla el sistema de nómina de la empresa. Te encargan probar algunas operaciones matemáticas con las variables que representan el salario y los impuestos.
- ▶ **Contenidos trabajados:**
 - ▶ 1.8 Operadores y expresiones
 - ▶ 1.9 Conversiones de tipo (básicas implícitas, sin pedir entrada ni salida)
- ▶ **Tarea:**
 - ▶ Leer el código y explicar con comentarios qué se está calculando.
 - ▶ Añadir una variable extra con un pequeño aumento (por ejemplo, aumento = 50) y recalcular el sueldo total.
 - ▶ Reflexionar: ¿Qué pasa si cambiamos impuesto a 0,20? (pueden observar el nuevo valor de sueldo neto al cambiar la línea de código)