

Guía docente de la asignatura

Metodología de la Programación (2961118)



Fecha de aprobación: 27/06/2024

Grado	Grado en Ingeniería Informática	Rama	Ingeniería y Arquitectura				
Módulo	Formación Básica	Materia	Informática				
Curso	1º	Semestre	2º	Créditos	6	Tipo	Troncal

PRERREQUISITOS Y/O RECOMENDACIONES

Los estudiantes no necesitan tener asignaturas aprobadas como requisito indispensable para cursar esta asignatura, no obstante, se recomienda la adquisición de los conocimientos y competencias de las materias de formación básica, teniendo especial relevancia los procedentes de la materia de “Fundamentos de Programación”.

BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Grado)

- Tipos de datos del lenguaje de alto nivel y su representación interna.
- Referencias de memoria y memoria dinámica.
- Encapsulamiento y ocultamiento de la información.
- Diseño modular y creación de bibliotecas.
- Herramientas de depuración, pruebas y validación.
- Gestión de errores.
- Mantenimiento del software.
- E/S, ficheros.
- Proyecto informático de programación.

COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

COMPETENCIAS ESPECÍFICAS

- CE04 - Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.
- CE05 - Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.



RESULTADOS DE APRENDIZAJE (Objetivos)

- Comprender la relación entre tipos de alto nivel y la representación a bajo nivel de dicha información
- Distinguir los conceptos de eficiencia en tiempo y espacio, así como su relación cuando se desarrolla un programa.
- Distinguir y manejar correctamente las referencias y los objetos referenciados.
- Justificar la importancia de los conceptos de encapsulamiento y ocultamiento de la información.
- Aprender a desarrollar nuevos tipos de datos, realizando una correcta separación entre interfaz e implementación.
- Saber enfrentarse a problemas de mayor tamaño considerando una división en subproblemas y una solución basada en la programación modular y la abstracción.
- Comprender cómo los mecanismos de abstracción soportan la creación de componentes software modulares y reutilizables.
- Manejar correctamente herramientas de depuración, pruebas y validación.
- Aprender a desarrollar código con una correcta gestión de condiciones de excepción.
- Entender la necesidad de un correcto diseño para obtener un software de mayor calidad (mejor preparado para su mantenimiento).
- Ser capaces de desarrollar la solución de problemas de mayor tamaño, incluyendo una correcta implementación y documentación.
- Asimilar los principios básicos de la abstracción para facilitar el estudio de la programación orientada a objetos.
- Aprender a realizar una correcta gestión de la E/S, especialmente motivada por la necesidad de manejar grandes cantidades de información almacenada en ficheros.

PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

TEÓRICO

Tema 1. Punteros y memoria dinámica

1. El tipo de dato puntero.
2. Vectores, matrices, cadenas y punteros.
3. Memoria dinámica.
4. Ejemplos de estructuras de datos simples.

Tema 2. Funciones

1. La función main.
2. La responsable de que todo funcione: La Pila.
3. Paso de parámetros y devolución de resultados.
4. Parámetros con valor por defecto.
5. Funciones inline.
6. Punteros a función.

Tema 3. Tipos de datos abstractos en C++: Clases

1. Abstracción y diseño de clases: atributos y métodos.
2. Constructores, destructor y asignación en clases que gestionan memoria dinámica.
3. Sobrecarga de operadores.



Tema 4. Gestión de E/S. Ficheros

1. Flujos de E/S.
2. Operaciones básicas con flujos.
3. Flujos asociados a string.
4. Flujos asociados a ficheros.
5. Ficheros de texto y binarios.

PRÁCTICO

Prácticas sobre los contenidos teóricos.

Los estudiantes realizarán prácticas sobre los contenidos desarrollados en el programa teórico mediante la resolución de diferentes problemas propuestos.

- Prácticas sobre el tema 1. Punteros y memoria dinámica.
- Prácticas sobre el tema 2. Funciones.
- Prácticas sobre el tema 3. Clases.
- Prácticas sobre el tema 4. Gestión de E/S. Ficheros

Proyecto informático de programación.

Los estudiantes aplicarán los conocimientos y destrezas adquiridas en la resolución de un proyecto de programación completo, incluyendo un correcto análisis, diseño, implementación y documentación.

Seminarios.

A lo largo del curso se trabajará en seminarios específicos:

- Seminario 1. El modelo de compilación en C++. El programa g++
 1. Compilación separada.
 2. El preprocesador, el compilador y el enlazador
 3. g++ : el compilador de GNU para C++
- Seminario 2. Gestión automatizada de proyectos.
 1. El programa make
 2. Ficheros makefile
 3. Entornos de programación.
- Seminario 3. Modularización. Bibliotecas. El programa ar.
 1. Bibliotecas
 2. El programa ar
- Seminario 4. Técnicas de gestión de errores y depuración
 1. Devolución de valores de error.
 2. Aserciones: errores en depuración.
 3. Excepciones.
 4. Herramientas de ayuda a la depuración.
- Seminario 5. Documentación de software
 1. Diseño e implementación.
 2. Herramientas automáticas de documentación.

BIBLIOGRAFÍA

BIBLIOGRAFÍA FUNDAMENTAL

- A. Garrido. "Metodología de la programación: de bits a objetos". Editorial Universidad de Granada, 2016. ISBN 978-84-338-5868-9.
- T. Gaddis, J. Walters, G. Muganda. "Starting Out with C++: Early Objects (10th Edition)".



Pearson, 2019

- A.Garrido. "Prácticas con C++: metodología de la programación (2ª ed.)". Editorial Universidad de Granada, 2017. Edición electrónica. ISBN 978-84-338-6032-3.
- Deitel & Deitel. "C++: How to Program (10th Edition)". Pearson, 2017

BIBLIOGRAFÍA COMPLEMENTARIA

- A. Garrido y J. Martínez-Baena. "Introducción a la programación con C++: ejercicios". Editorial Universidad de Granada, 2016. Edición electrónica. ISBN 978-84-338-5924-2.
- A. Garrido. "Fundamentos de programación con la STL". Editorial Universidad de Granada, 2016. ISBN 978-84-338-5917-4.
- Bjarne Stroustrup. "The C++ Programming Language, 4th Edition". Addison Wesley Professional, 2013.
- Bjarne Stroustrup. "Programming: Principles and Practice Using C++" Addison-Wesley, 2014.

ENLACES RECOMENDADOS

Páginas para usar como manuales de referencia:

- [C++ Reference](#) (en inglés)
- [C Plus Plus](#) (en inglés)

METODOLOGÍA DOCENTE

- MD01 - Lección Magistral (Clases Teóricas-Expositivas)
- MD02 - Actividades Prácticas (Resolución de Problemas, Resolución de Casos Prácticos, Desarrollo de Proyectos, Prácticas en Laboratorio, Taller de Programación, Aula de Informática, Prácticas de Campo).
- MD03 - Seminarios (Debates, Demos, Exposición de Trabajos Tutelados, Conferencias, Visitas Guiadas, Monografías).
- MD04 - Actividades no presenciales Individuales.
- MD05 - Actividades no presenciales Grupales.
- MD06 - Tutorías Académicas.

EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

EVALUACIÓN ORDINARIA

La nota final del estudiante se calculará a partir de las calificaciones que obtenga en las siguientes partes:

Parte teórica: La ponderación de esta parte es del 60%.

- Para la parte teórica se realizará un examen escrito multi-pregunta sobre los contenidos de la materia impartida.
- Este examen se realizará en su convocatoria oficial ordinaria.

Parte práctica: La ponderación de esta parte es del 40%.

- La evaluación del trabajo práctico se realizará, preferentemente, de forma continua y



constará de las siguientes pruebas:

1. Un examen práctico individual que tendrá lugar a mitad del cuatrimestre. (10 %)
2. La realización de un proyecto informático al final del cuatrimestre (20 %)
3. La realización de guiones de prácticas a lo largo del cuatrimestre (10 %)

Si la nota de la parte teórica es superior o igual a 4 (sobre 10), entonces la nota final será:

Nota final = $0,6 * \text{parte teórica} + 0,4 * \text{parte práctica}$

En otro caso, la nota final será la nota de la parte teórica (sobre 10), pudiendo el estudiante mantener la nota de la parte práctica para la convocatoria extraordinaria.

EVALUACIÓN EXTRAORDINARIA

Los estudiantes podrán optar entre: conservar la nota de cada parte obtenida en la convocatoria ordinaria o volver a ser evaluados.

Se realizarán las siguientes pruebas en un único acto académico:

Parte teórica: La ponderación de esta parte es del 60%.

- Para la parte teórica se realizará un examen escrito multi-pregunta sobre los contenidos de la materia impartida.

Parte práctica: La ponderación de esta parte es del 40%.

- Para la parte práctica se realizará un examen práctico multi-pregunta sobre los contenidos de la materia impartida.

Si la nota de la parte teórica es superior o igual a 4 (sobre 10), entonces la nota final será:

Nota final = $0,6 * \text{parte teórica} + 0,4 * \text{parte práctica}$

En otro caso, la nota final será la nota de la parte teórica (sobre 10).

EVALUACIÓN ÚNICA FINAL

Esta modalidad de evaluación se realizará en un único acto académico en la fecha establecida por el Centro y consistirá en un examen (evaluado de 0 a 10) que incluirá preguntas que garanticen que el alumno ha adquirido la totalidad de las competencias, tanto teóricas como prácticas, descritas en esta misma guía docente.

Parte teórica: La ponderación de esta parte es del 60%.

Parte práctica: La ponderación de esta parte es del 40%.

Si la nota de la parte teórica es superior o igual a 4 (sobre 10), entonces la nota final será:

Nota final = $0,6 * \text{parte teórica} + 0,4 * \text{parte práctica}$

En otro caso, la nota final será la nota de la parte teórica (sobre 10).

INFORMACIÓN ADICIONAL

Información de interés para estudiantado con discapacidad y/o Necesidades Específicas de Apoyo Educativo (NEAE): [Gestión de servicios y apoyos \(https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad\)](https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad).

