Guía docente de la asignatura

Metodología de la Programación

Fecha última actualización: **21/06/2021**
Fecha de aprobación: **21/06/2021**

Grado	Grado en Ingeniería Informática y Matemáticas	**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. Funciones inline.
5. Parámetros con valor por defecto.
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 ficheros.
4. Ficheros de texto y binarios.

PRÁCTICO

Práctica 1. Compilación separada y gestión de proyectos.

Práctica 2. Punteros y memoria dinámica.

Práctica 3. Abstracción en C++: clases.

Práctica 4. Proyecto informático de programación.

Seminario 1. Primeros programas con linux

1. Órdenes básicas: Entorno de desarrollo.
2. Compilación y enlazado en linux.
3. Depuración en linux.

Seminario 2. Compilación separada y espacios de nombres

1. Compilación separada.
2. El preprocesador.
3. Bibliotecas.
4. Espacios de nombres.

Seminario 3. Tipos aritméticos. Representación y conversiones

1. Tipos integrales y en coma flotante.
2. Características de los tipos.
3. Conversiones.
4. Operadores lógicos a nivel de bit.

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 (8th Edition)”. Addison Wesley 2013
- 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. Prentice Hall-Pearson, 2013

BIBLIOGRAFÍA COMPLEMENTARIA

- A. Garrido. "Fundamentos de programación con la STL". Editorial Universidad de Granada, 2016. ISBN 978-84-338-5917-4.
- 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.
- Walter Savitch. “Resolución de problemas con C++”. Pearson, 2006.
- Bjarne Stroustrup. “El Lenguaje de Programación C++. Edición especial”. Addison Wesley, 2002.
- Bjarne Stroustrup. “The C++ Programming Language, 4th Edition”. Addison Wesley Professional, 2013.
- Sedgewick. “Algorithms in C++”. Addison-Wesley, 2002.
- A. Garrido. “Fundamentos de Programación en C++”. Delta Publicaciones, 2005.
- A. Garrido, J. Fdez-Valdivia. “Abstracción y estructuras de datos en C++”. Delta publicaciones, 2006.

ENLACES RECOMENDADOS

Páginas para usar como referencia:

- C++ Reference (en inglés) http://www.cppreference.com
- C Plus Plus (en inglés) http://www.cplusplus.com

Cursos en internet:

- C++ con clase. http://c.conclase.net
- Zator. http://www.zator.com/Cpp/

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 **3,5** (sobre 10), entonces la nota final será:

Nota final = 0,6 * parte teórica + 0,4 * parte práctica

En otro caso, la nota final será la nota de la parte teórica (sobre 6), 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 **3,5** (sobre 10), entonces la nota final será:

Nota final = 0,6 * parte teórica + 0,4 * parte práctica

En otro caso, la nota final será la nota de la parte teórica (sobre 6).

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 **3,5** (sobre 10), entonces la nota final será:

Nota final = 0,6 * parte teórica + 0,4 * parte práctica

En otro caso, la nota final será la nota de la parte teórica (sobre 6).