

Guía docente de la asignatura

## Fundamentos de Programación



Fecha última actualización: 21/06/2021

Fecha de aprobación: 21/06/2021

<b>Grado</b>	Grado en Ingeniería Informática	<b>Rama</b>	Ingeniería y Arquitectura				
<b>Módulo</b>	Formación Básica	<b>Materia</b>	Informática				
<b>Curso</b>	1º	<b>Semestre</b>	1º	<b>Créditos</b>	6	<b>Tipo</b>	Troncal

### PRERREQUISITOS Y/O RECOMENDACIONES

No se necesita ningún prerrequisito.

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

Tipos de datos básicos. Objetos, operadores y expresiones. Estructuras de control. Funciones y procedimientos. Tipos de datos compuestos: homogéneos (arrays) y heterogéneos (registros). Algoritmos básicos de ordenación y búsqueda. Recursividad.

### COMPETENCIAS ASOCIADAS A MATERIA/ASIGNATURA

#### COMPETENCIAS GENERALES

- CG08 - Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

#### 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 el funcionamiento de un computador, haciendo especial énfasis en la necesidad de desarrollo de software por parte del programador.
- Presentar la historia de la programación y de los distintos paradigmas de programación, situando en ese contexto el lenguaje de programación que se va a utilizar.
- Comprender la necesidad de un proceso de traducción de un lenguaje de alto nivel.
- Conocer y distinguir los conceptos de algoritmo y programa.
- Mostrar la necesidad de codificar la información que maneja internamente un computador, enfatizando posibilidades y limitaciones cuando se resuelve un problema.
- Conocer los tipos de datos primitivos y sus operaciones.
- Distinguir entre tipo de dato y objeto.
- Conocer las acciones básicas de E/S de datos.
- Aprender a usar las estructuras de control básicas: secuencial, condicional e iterativa.
- Comprender la necesidad de dividir la solución creando módulos (funciones o procedimientos) que implementen operaciones no primitivas.
- Comprender la necesidad de la especificación de una función o procedimiento, como método de abstracción, introduciendo los conceptos de precondition y postcondition.
- Aprender a resolver problemas aplicando una metodología de diseño modular (top-down/bottom-up).
- Manejar correctamente los mecanismos de comunicación entre módulos (interfaces), así como las distintas formas de paso de parámetros y devolución de resultados.
- Entender la gestión de llamadas a funciones mediante la pila.
- Motivar e introducir los tipos de datos compuestos, arrays y registros, así como sus operaciones.
- Conocer los algoritmos de ordenación básicos (selección, inserción, burbuja).
- Conocer los algoritmos de búsqueda básicos (lineal, dicotómica).
- Motivar y aprender a resolver problemas mediante algoritmos recursivos

## PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

### TEÓRICO

#### Tema 1. Programación en C++: Fundamentos

1. El ordenador, algoritmos y programas
2. Especificación de programas
3. Datos y tipos de datos
4. Operadores y Expresiones
5. Tipos de datos comunes en C++

#### Tema 2. Estructuras de Control

1. Estructura Condicional
2. Estructuras Repetitivas

#### Tema 3. Funciones y Clases

1. Funciones
2. Clases

#### Tema 4. Vectores y Matrices

1. Vectores



## 2. Matrices

### Tema 5. Recursividad

1. Diseño de algoritmos recursivos
2. Funciones recursivas

### PRÁCTICO

Práctica 1. Presentación y uso de un entorno de desarrollo para programar C++. Resolución de problemas sobre:

- Expresiones, sentencias y programas.
- Uso de tipos de datos para representar información.

Práctica 2. Resolución de problemas sobre:

- Estructuras condicionales.
- Estructuras repetitivas.

Práctica 3. Resolución de problemas sobre:

- Definición y uso de funciones que trabajan sobre tipos simples.
- Definición de clases. Ámbitos privado y público.
- Acceso de los métodos de una clase a sus variables de instancia.

Práctica 4. Realización de Problemas sobre:

- Construcción de clases que proporcionen acceso y operaciones sobre conjunto de datos.
- Otros métodos de búsqueda y ordenación de vectores.
- Matrices dentro de una clase.

### Seminarios

Seminario 1. Test y depuración de programas mediante el uso del entorno de programación.

Seminario 2. Documentación de funciones.

- Documentación de cabeceras.
- Descripción de algoritmos.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA FUNDAMENTAL

- A. Garrido. Fundamentos de Programación en C++. Delta Publicaciones, 2005.
- R. Mercer. Computing Fundamentals with C++. Object-oriented Programming and Design. 2ª Edición. Palgrave Macmillan, 2000.
- T. Gaddis, J. Walters, G. Muganda. Starting out with C++: early objects. 8ª Edición. Pearson, 2013.
- W. Savitch. Resolución de problemas con C++. Pearson, 2006.



## BIBLIOGRAFÍA COMPLEMENTARIA

- A. Garrido. Fundamentos de programación con la STL. Editorial Universidad de Granada, 2016.
- A. Garrido. Metodología de la Programación: de bits a objetos. Editorial Universidad de Granada, 2016.
- R. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.
- S. McConnell. Code Complete: A Practical Handbook of Software Construction. 2ª Edición. Microsoft Press, 2004.
- Object-oriented Programming in C++. Robert Lafore. Sams Publishing, 2001.
- B. Stroustrup. The C++ Programming Language. 4ª Edición. Addison-Wesley, 2013.

## ENLACES RECOMENDADOS

- C++ con clase. <http://conclase.net/c>
- C Plus Plus (en inglés) <https://www.cplusplus.com>
- C++ Reference (en inglés) <http://www.cppreference.com>
- Zator (libro programación) <http://www.zator.com/Cpp/>
- Plataforma enseñanza C++.  
<http://decsai.ugr.es/~jfv/ed1/c%2B%2B/cdrom3/TIC-CD/web/index.htm>

## 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

Todo lo relativo a la evaluación se registrará por la Normativa de evaluación y calificación de los estudiantes vigente en la Universidad de Granada, que puede consultarse en: <https://www.ugr.es/universidad/normativa/texto-consolidado-normativa-evaluacion-calificacion-estudiantes-universidad-granada>

La nota final del alumno se calculará a partir de las calificaciones que obtenga en las siguientes partes, y según la ponderación indicada:

- PARTE TEÓRICA: La ponderación de esta parte es del 60%. Para la parte escrita 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 consta de los siguientes apartados:
  - La parte de la nota de la evaluación continua se obtendrá con distintas pruebas que se realizarán durante el desarrollo de las clases prácticas, además de la asistencia y entrega de los trabajos requeridos. La ponderación de esta parte es del 10%.
  - Se realizarán pruebas de resolución de problemas a lo largo de la asignatura. La ponderación de esta parte es del 30%.

Si la nota de la parte teórica es superior o igual a 3,5 sobre 10, entonces la nota final será:

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

En otro caso, la nota final será la nota de la parte teórica (sobre 6), pudiendo el alumno mantener la nota de la parte práctica para la convocatoria extraordinaria.

### EVALUACIÓN EXTRAORDINARIA

Los alumnos podrán optar, entre conservar la nota de cada parte obtenida en la convocatoria ordinaria o volver a evaluarse.

Se realizarán las siguientes pruebas en un único acto académico:

- **PARTE TEÓRICA:** La ponderación de esta parte es del 60%. Para la parte escrita 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 en el ordenador.

Si la nota de la parte teórica es superior o igual a 3,5 sobre 10, entonces la nota final será:

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

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

### EVALUACIÓN ÚNICA FINAL

De acuerdo a lo establecido en la Normativa de evaluación y de calificación de los estudiantes de la Universidad de Granada, la evaluación será preferentemente continua. No obstante, el estudiante que no pueda acogerse a dicho sistema por motivos laborales, estado de salud, discapacidad o cualquier otra causa debidamente justificada podrá acogerse a la evaluación única final.

Para acogerse a la evaluación única final, el estudiante, en las dos primeras semanas de impartición de la asignatura, o en las dos semanas siguientes a su matriculación si ésta se ha producido con posterioridad al inicio de la asignatura, lo solicitará, a través del procedimiento electrónico, al Director del Departamento o al Coordinador del Máster, alegando y acreditando las razones que le asisten para no poder seguir el sistema de evaluación continua

Aquellos alumnos que opten por acogerse a la "Evaluación Única Final", realizarán un examen en ordenador como evaluación de la parte práctica. La parte teórica se evaluará mediante un examen escrito multi-pregunta sobre los contenidos de la materia impartida. Ambos exámenes se



realizarán en un solo acto académico.

El peso de la evaluación de las actividades formativas, se ajustará a lo indicado a continuación:

- **PARTE TEÓRICA:** La ponderación de esta parte es del 60%. Para la parte escrita 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%. Para la parte práctica se realizará un examen práctico en el ordenador.

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).

Para más información sobre la evaluación única final, véase la Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada:

<https://www.ugr.es/universidad/normativa/texto-consolidado-normativa-evaluacion-calificacion-estudiantes-universidad-granada>

