

Guía docente de la asignatura

## Fundamentos de Programación (2961115)



Fecha de aprobación: 23/06/2023

<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.



## COMPETENCIAS TRANSVERSALES

- CT06 - Motivación por la calidad y la mejora continua, actuando con rigor, responsabilidad y ética profesional.

## 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 registros, vectores y matrices, 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

1. Fundamentos
2. Diseño de funciones

### Tema 4. Registros, Vectores y Matrices

1. Registros
2. Vectores. Algoritmos de búsqueda y ordenación sobre vectores
3. Matrices

### Tema 5. Clases

1. Encapsulación. Datos y métodos.
2. Ocultación de información. Ámbito público y privado.
3. Constructores.
4. Copias de objetos.
5. Datos y métodos constantes
6. Colecciones de datos: clases tipo secuencia y tabla

### Tema 6. Recursividad

1. Diseño de algoritmos recursivos
2. Funciones recursivas
3. Clases con métodos recursivos

## 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 registros, vectores y matrices.

Práctica 5. Realización de Problemas sobre:

- Colecciones de datos (vectores y matrices) dentro de una clase.
- Acceso, búsqueda, ordenación y otras operaciones de gestión sobre colecciones de datos encapsulados en clases.

Práctica 6. Realización de Problemas sobre Recursividad  
Seminarios

1. Seminario 1. Test y depuración de programas mediante el uso del entorno de programación.
2. 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. Lafore. Object-oriented Programming in C++. Sams Publishing 2005.

### 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.
- B. Stroustrup. The C++ Programming Language. 4ª Edición. Addison-Wesley, 2015.
- R. Mercer. Computing Fundamentals with C++. Franklin, Beedle and Associates, 2018.
- T. Gaddis, J. Walters, G. Muganda. Starting out with C++: early objects. 10ª Edición. Pearson, 2019.
- W. Savitch. Resolución de problemas con C++. Pearson, 2017.

## ENLACES RECOMENDADOS

- [cplusplus.com](http://cplusplus.com)
- [cppreference.com](http://cppreference.com)
- [C++ con clase](#)
- [TIC C++](#)

## 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.](#)

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 realizará una o varias 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 4 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 10)

### 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 4 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 10).

### EVALUACIÓN ÚNICA FINAL

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 4 (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 10).

Para más información sobre la evaluación única final, véase la [Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada.](#)

