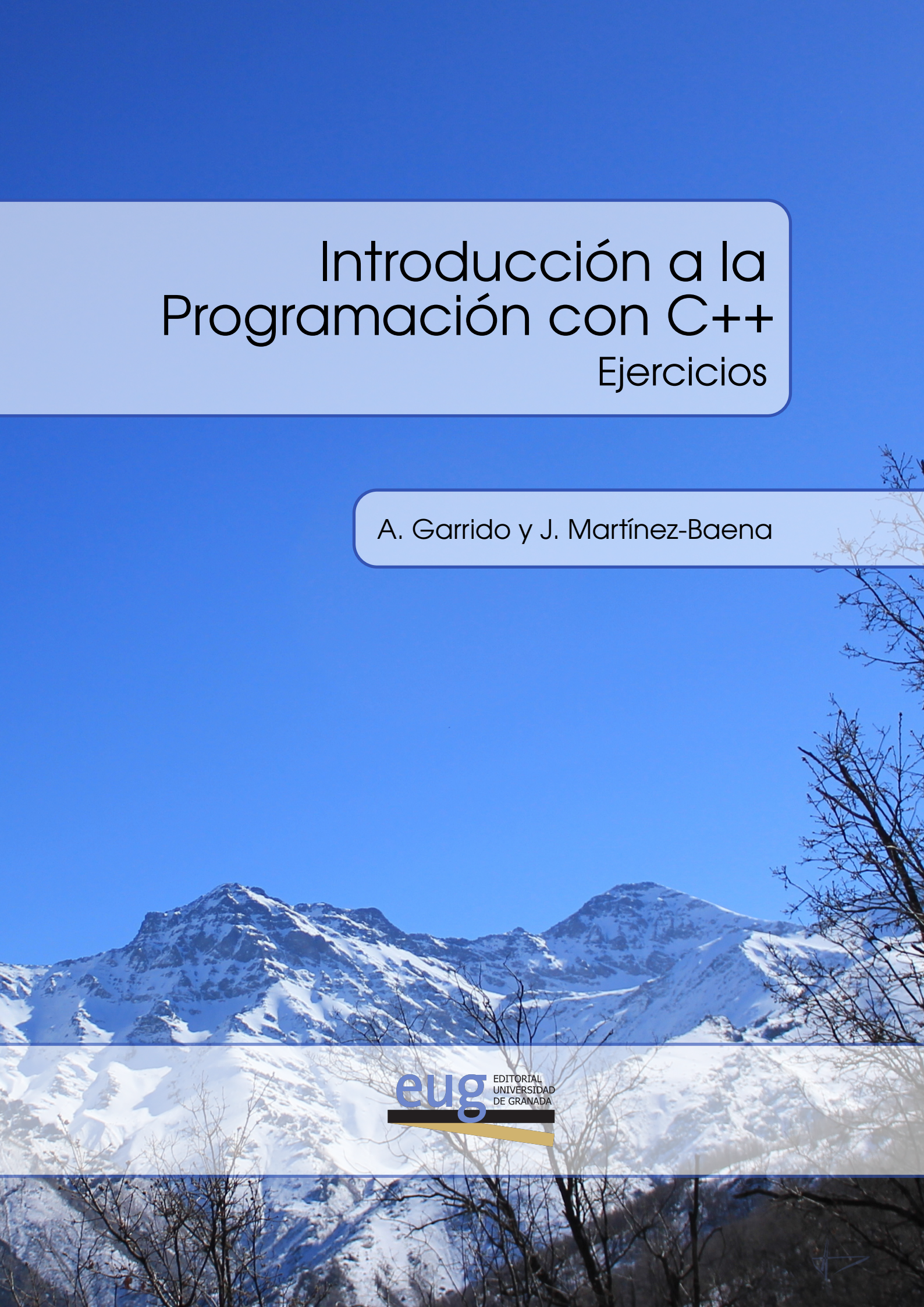
Introducción a la Programación con C++

Ejercicios

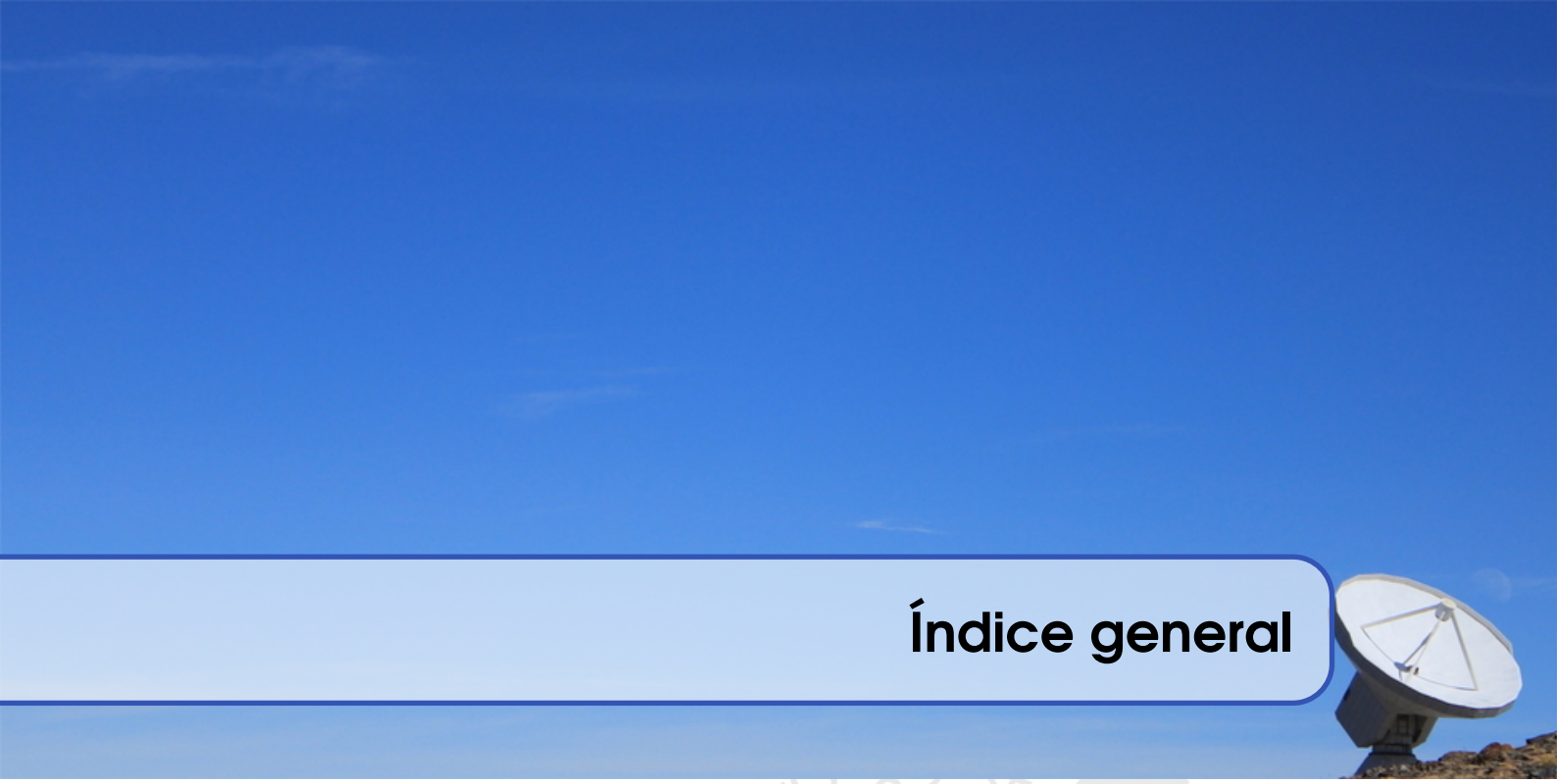
A. Garrido y J. Martínez-Baena



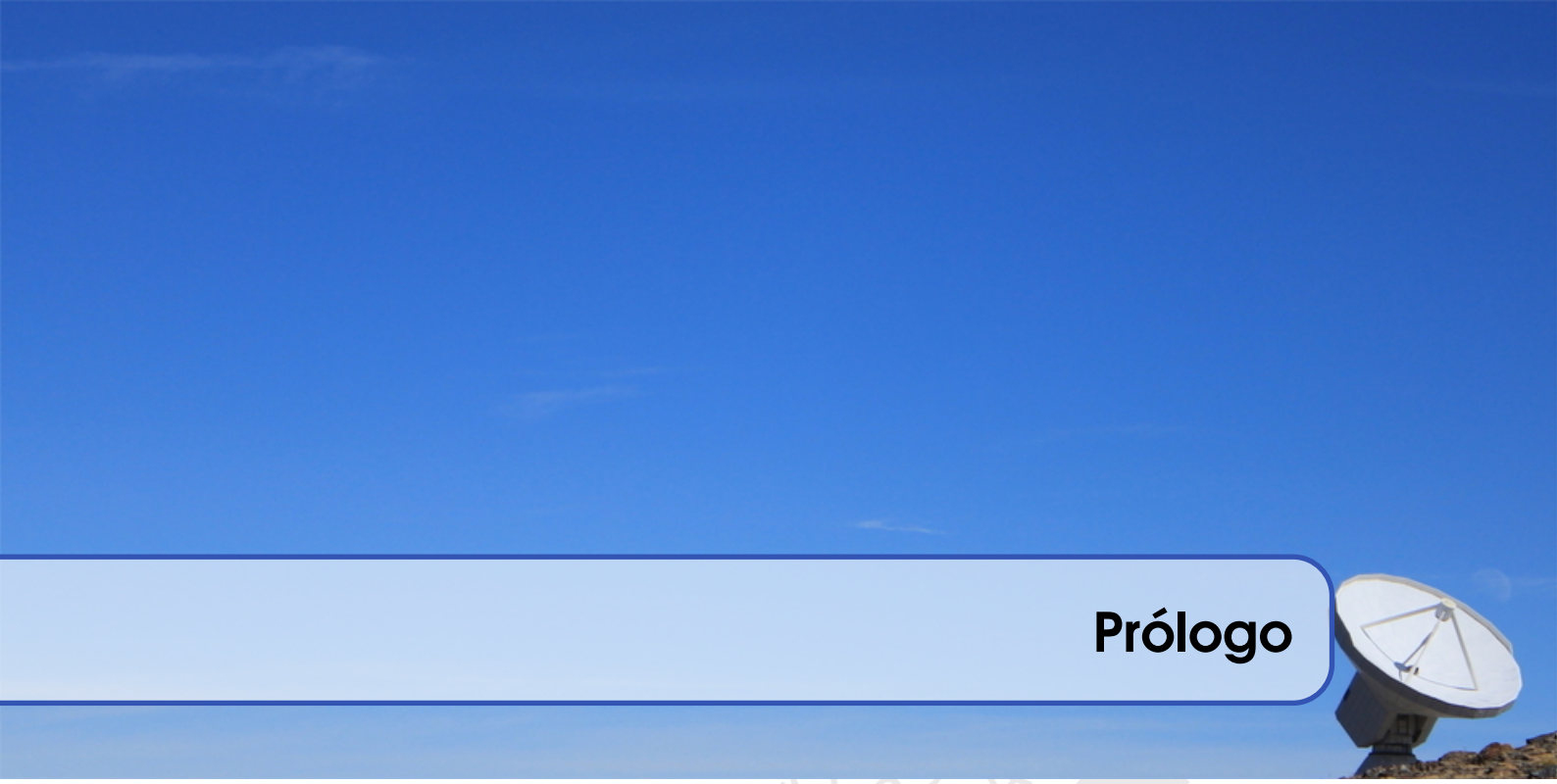
eug EDITORIAL UNIVERSIDAD DE GRANADA

INTRODUCCIÓN A LA PROGRAMACIÓN CON C++: Ejercicios
ISBN: 978-84-338-5924-2.
Edita: Editorial Universidad de Granada.
Campus Universitario de Cartuja. Granada.

Printed in Spain *Impreso en España*.



Índice general



Prólogo

La enseñanza de la programación de ordenadores es un tema ampliamente discutido, no sólo por la dificultad de enseñar una serie de conceptos y habilidades que pueden resultar bastante complicados, sino también por la rapidez con que cambia la tecnología, dando lugar no sólo a nuevos lenguajes de programación, sino también a nuevas formas de abordarla.

El rápido desarrollo de las tecnologías disponibles para programar ordenadores ha tenido un efecto directo sobre la enseñanza, ya que ha incrementado el conjunto de habilidades que se supone que debe poseer un programador. Aun así, no podemos olvidar que los fundamentos de la programación son los mismos.

Un curso de fundamentos de programación presenta conceptos relativamente simples. Sin embargo, para un estudiante que comienza en la programación, los conceptos que se discuten pueden resultar bastante abstractos y difíciles de asimilar. Esta dificultad puede verse compensada en gran medida con una metodología orientada a la práctica. Por ello, tan importante como disponer de un buen manual de referencia sobre el lenguaje de estudio —C++ en nuestro caso— lo es disponer de un buen libro de apoyo en la realización de prácticas.

Precisamente, ese es el objetivo de este libro: ofrecer al estudiante un documento de trabajo para facilitar el aprendizaje aplicando de forma práctica los conceptos aprendidos en clase o en otros libros más teóricos sobre C++ y la *STL*. En concreto, nos permite ofrecer un guión de prácticas relacionadas con el libro *Fundamentos de Programación con la STL* (Garrido[5]).

Organización del cuaderno de prácticas

Para compensar la falta de experiencia del estudiante, los ejercicios del cuaderno se diseñan para que incidan especialmente en situaciones habituales en la práctica de programación. De hecho, un programador con experiencia consideraría simples muchos de los problemas que se proponen. Esta simplicidad es consecuencia de que tiene asimilados muchos esquemas de solución —patrones de diseño— que ha resuelto repetidamente.

Los temas se plantean con una serie de ejercicios que hagan explícitas estas formas de solución. Adicionalmente, en algunos casos se incluye un pequeño resumen de aspectos teóricos importantes o una discusión sobre la forma en que cierto diseño facilita la solución de problemas. Cada sección suele estar diseñada para practicar algún aspecto concreto. Además, parte de los ejercicios permiten profundizar en algunos puntos de especial relevancia y que suelen pasar desapercibidos por los estudiantes.

El libro está diseñado para comenzar desde cero el aprendizaje de la programación. Como se puede comprobar con un simple vistazo al índice de capítulos, se trata de programación estructurada y modular. Aunque C++ es un lenguaje multiparadigma, este cuaderno se ha creado para un curso de fundamentos; el objetivo es que el estudiante acabe con una base sólida que permita el abordaje de curso más avanzados con garantías de éxito. Más concretamente, se llevan a cabo prácticas que incorporan contenidos sobre:

- Programas simples con expresiones en C++.
- Estructuras de control, incluyendo la selección y la iteración.
- Diseño de funciones.
- Tipos compuestos homogéneos: vectores, matrices y cadenas.
- Tipos compuestos heterogéneos: estructuras y pares.
- Problemas simples de E/S con ficheros de texto.

Finalmente, el documento incluye apéndices con información relevante para realizar los guiones. En concreto, podrá usarlos para consultar:

- La *generación de números aleatorios*. Es un tema que generalmente no se aborda directamente en las clases de teoría, sino que se supone se practicará cuando se desarrollen programas que generan valores aleatorios. Sin embargo, es un tema cuyo contenido teórico es muy relevante para poderlo usar adecuadamente. En lugar de dar una breve especificación de las funciones que ofrece el lenguaje, se incluye una exposición más detallada con el fin de que el estudiante no sólo lo use, sino de que entienda por qué funciona.
- La *redirección de E/S*. Aunque inicialmente se plantea como un tema relacionado con la E/S, se ha dejado para un apéndice puesto que es un contenido transversal para todo el curso. Desde los primeros ejercicios ya podemos aprovechar este redireccionamiento para ejecuciones rápidas desde la línea de órdenes.
- *Tablas* relacionadas con el lenguaje. En la práctica, es muy útil disponer de tablas que incluyen detalles sobre palabras reservadas, operadores del lenguaje, código *ASCII*, etc.

Entorno de programación

El curso está basado en el estándar más extendido de C++ —el del 98/03— aunque sigue siendo eficaz si está interesado en programar con el último estándar, ya que los ejercicios y las discusiones siguen siendo igualmente válidos. Desde este punto de vista, puede usar cualquier versión, aunque se recomienda al menos la del 98. En la fecha que estamos, seguro que cualquier compilador cumple con esta condición.

No es necesario usar ningún compilador o entorno de programación concreto. Se ha desarrollado para que el estudiante sea libre de optar por las herramientas que le sean más cómodas, siendo casi cualquier compilador estándar una buena opción para practicar. Sólo es necesario un editor de texto y un compilador.

Sin embargo, siendo un primer contacto con la programación, es recomendable que el estudiante pueda centrarse en los detalles del lenguaje y deje la dificultad de la gestión de proyectos para más adelante. Por ello, se recomienda algún entorno que facilite la compilación y ejecución. En nuestro caso, hemos usado *Code::Blocks*, que se ajusta perfectamente a los objetivos del curso.

Agradecimientos

Los autores desean agradecer a todos sus alumnos por haber contribuido —aun sin saberlo— a la creación de este cuaderno, especialmente a los que *"se equivocan"* porque precisamente ellos permiten encontrar los puntos débiles en el proceso de enseñar a programar.

Esta sección no puede terminar sin un agradecimiento a todos los que con su trabajo desinteresado han contribuido al software libre, creando sistemas y herramientas gratuitas que nos permiten por un lado crear este documento —íntegramente desarrollado con software libre— y por otro, mucho más interesante, ofrecer una solución totalmente gratuita para realizar este curso.

A. Garrido y J. Martínez-Baena
Mayo de 2016.

1 Expresiones en C++

1.1 Introducción

En este capítulo se presenta el estudio de las expresiones básicas en C++. En concreto se estudiarán los siguientes aspectos:

- Tipos de datos básicos y sus operaciones.
- Expresiones.
- Funciones de biblioteca.

A pesar de la amplia gama de tipos de datos que ofrece el lenguaje, no se estudiarán en profundidad, ya que por un lado prácticamente todos los programas que se van a crear pueden resolverse con los tipos básicos, y por otro no aportan un valor añadido en un curso introductorio. Más adelante, cuando ya se posea una base de programación y se requieran soluciones más específicas y optimizadas, será fácil incorporar todos esos conocimientos.

1.1.1 Tipos de datos y operaciones

Antes de empezar, es importante que el alumno entienda que un programa puede manejar una amplia variedad de información y, para ello, incluye la posibilidad de almacenar y operar con distintos tipos de datos. Un tipo de dato implica:

- Una representación. La información la almacena y maneja un ordenador que sólo sabe trabajar con ceros y unos. Cualquier dato que quiera introducir en un ordenador deberá transformarse a ceros y unos. Es inevitable que los datos se representen siempre en ese lenguaje binario.
  Como consecuencia de esa representación, los tipos de datos que maneja el ordenador tendrán ciertas características muy concretas. Por ejemplo, debido a la limitación en la cantidad de memoria que posee el ordenador, se puede limitar el tamaño de los datos. Un caso concreto sería guardar un dato de tipo entero, que sabemos que tiene un número infinito de posibilidades, pero que en nuestros programas estará limitado a un rango finito de valores.
- Un conjunto de operaciones. El lenguaje nos ofrece, para cada tipo, un conjunto de operaciones asociado. Estas operaciones nos permiten obtener nuevos datos como resultado de operar con uno o más operandos. Por ejemplo, podemos obtener un nuevo entero como resultado de la suma de dos datos del mismo tipo.

En este capítulo vamos a introducir la creación de expresiones donde aparecen datos de distintos tipos relacionados mediante un conjunto de operadores. Para escribir nuestros programas, es fundamental entender perfectamente cómo se evalúan estas expresiones y qué resultados se obtienen.

1.2 Tipos de datos numéricos

En esta sección vamos a introducir algunos ejemplos de programas simples en C++ con la intención de que el lector asimile más fácilmente las consecuencias prácticas de los conceptos y reglas que se estudian de forma más detallada en clase de teoría.

Aunque durante la exposición se incluirán algunos contenidos para recordar y enfatizar algunos detalles clave, es recomendable que revise los contenidos teóricos antes trabajar este tema. Los ejercicios se han diseñado para hacer que el lector infiera los conceptos fundamentales del tema, evitando memorizar reglas e invitando a razonar sobre los contenidos que se presentan.