

# Prefacio

En este libro se combinan contenidos que habitualmente se estudian por separado en asignaturas específicas dentro de algunas titulaciones de informática. En particular es el caso de la teoría de los Lenguajes de Programación, que en algunas titulaciones corresponde a una asignatura de la que hay numerosa bibliografía específica, y también es el caso del estudio de los Procesadores de Lenguajes, que también suele corresponder a una asignatura con su correspondiente bibliografía específica. Normalmente entre estas dos materias hay cierto solapamiento de temas aunque no se abordan con igual profundidad. Por ejemplo, en la bibliografía de teoría de los lenguajes de programación se tratan aspectos de la sintaxis, pero no se detallan los algoritmos de análisis sintáctico que se utilizan, que es una materia propia de los procesadores. De la misma forma, en la bibliografía de procesadores de lenguajes se tratan aspectos de la teoría de lenguajes, como tipos de datos, ámbito de las variables, etc. aunque no se entre en detalle en cuanto a los diferentes paradigmas. Por otra parte, hay libros que también tratan estas dos materias aunque no profundizan en los lenguajes de *script* ni tratan otros tipos de lenguajes como los de marcado.

Sin embargo, en algunos nuevos grados en informática propuestos dentro del marco de la EEES<sup>1</sup> se ha adoptado otro enfoque a la hora de estructurar estos contenidos. Es el caso de algunos grados relacionados con las Tecnologías de la Información que, siguiendo las recomendaciones del Computing Curricula de la ACM y el IEEE (*Curriculum Guidelines for Undergraduate Degree Programs in Information Technology*<sup>2</sup>), han dedicado menos créditos a estas materias de manera que ahora se abordan unificadas junto con otras relacionadas en una única asignatura. Este libro ofrece una respuesta a este nuevo enfoque combinando contenidos de teoría de los lenguajes de programación, procesadores de lenguajes, paradigmas de los lenguajes de programación, aspectos pragmáticos de los lenguajes de programación y lenguajes de marcado. En particular este libro está pensado como texto básico para la asignatura de "Lenguajes de Programación y Procesadores" del Grado en Ingeniería en Tecnologías de la Información de la UNED.

---

<sup>1</sup>Espacio Europeo Educación Superior (<http://www.eees.es/>)

<sup>2</sup>Information Technology 2008. Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. Association for Computing Machinery (ACM) and IEEE Computer Society

## Planificación del libro

El contenido del libro se puede cubrir en un semestre académico. Este libro está pensado como texto básico para la enseñanza de la teoría de los lenguajes de programación y sus paradigmas, así como de la teoría de los procesadores de lenguajes. Su contenido se centra en los aspectos introductorios fundamentales de las materias anteriores y los complementa con el estudio de los lenguajes de *script* y de los lenguajes de marcado. La inclusión de éstos se justifica porque su uso resulta muy relevante en aplicaciones relacionadas con las Tecnologías de la Información.

Cada capítulo se estructura siguiendo el siguiente esquema:

- Presentación del tema al comienzo de cada capítulo para que el lector tenga una visión panorámica de los contenidos que va a encontrar y pueda conocer las dificultades que implica su estudio.
- Desarrollo del tema en cuestión proporcionando ejemplos para facilitar la comprensión de los aspectos teóricos presentados.
- Ejercicios resueltos representativos de los aspectos teóricos y prácticos más destacados del capítulo.
- Ejercicios propuestos que no están solucionados en el libro y que se presentan al lector como retos abiertos.
- Notas bibliográficas que comentan la bibliografía más destacada de cada capítulo y proporcionan referencias para complementarla.

## Prerrequisitos

En cuanto a programación de ordenadores se refiere, se le suponen al lector conocimientos de programación estructurada imperativa y nociones de orientación a objetos. También se le presuponen conocimientos de las estructuras de datos básicas como arrays, listas, colas, árboles y sus operaciones básicas. Por otro lado, conocer más de un lenguaje de programación puede facilitar la comprensión de algunos conceptos relacionados con la teoría de los lenguajes de programación.

También se le presuponen al lector conocimientos de los conceptos relacionados con la teoría de lenguajes formales: autómatas finitos, expresiones regulares y gramáticas. Aunque en el desarrollo de los contenidos de este libro se repasan los conceptos de autómatas finitos, lenguajes regulares, expresiones regulares, gramáticas regulares, autómatas a pila, y lenguajes y gramáticas independientes del contexto, si el lector ya está familiarizado con ellos, la lectura y comprensión de los capítulos dedicados a los lenguajes de programación y los procesadores de lenguajes se verá facilitada.

## Organización del contenido

El contenido del libro se estructura en dos partes:

**Parte 1.** Teoría de los lenguajes de programación y procesadores de lenguajes.

**Parte 2.** Lenguajes de computadora.

La primera parte consta de dos capítulos que introducen respectivamente los aspectos fundamentales de la teoría de los lenguajes de programación y los procesadores de lenguajes:

**Capítulo 1.** Lenguajes de programación.

**Capítulo 2.** Procesadores de lenguajes.

En la segunda parte se presentan los paradigmas y modelos de programación, los lenguajes de marcado, los lenguajes de *script* y los principales aspectos pragmáticos relacionados con los lenguajes de programación. Esta segunda parte consta a su vez de cuatro capítulos:

**Capítulo 3.** Paradigmas y modelos de programación.

**Capítulo 4.** Lenguajes de marcado. XML.

**Capítulo 5.** Lenguajes de *script*.

**Capítulo 6.** Aspectos pragmáticos de los lenguajes de programación.

El contenido más detallado de cada capítulo junto con las recomendaciones para su mejor aprovechamiento es el siguiente:

- **Capítulo 1. Lenguajes de programación.** Este capítulo expone las nociones básicas necesarias para entender los lenguajes de programación. Se presentan los conceptos de sintaxis y semántica de un lenguaje de programación, así como la notación BNF para la descripción de la sintaxis de estos lenguajes. Posteriormente se profundiza en los diferentes elementos que los componen: expresiones y enunciados, tipos de datos, ámbito de variables, etc. Cada uno de estos elementos es presentado enumerando sus principales características e indicando con ejemplos la forma en que se han incluido en diferentes lenguajes de programación. Los ejercicios de este capítulo hacen hincapié en la sintaxis de los lenguajes de programación y en el aprendizaje de los diferentes conceptos presentados a lo largo del tema.

- **Capítulo 2. Procesadores de lenguajes.** Este capítulo ofrece una visión general de las diferentes etapas de transformación de un programa, desde un código fuente escrito por un programador, hasta un fichero ejecutable. De entre los diferentes tipos de procesadores de lenguajes se destacan los compiladores, de los que se ofrece una explicación más en detalle como ejemplo paradigmático de un procesador de lenguajes. La descripción de las fases de un compilador se centra fundamentalmente en las etapas de análisis más que en las de síntesis. Los ejercicios de este capítulo se centran en aspectos relacionados con la fase de análisis de un compilador, así como en la posterior etapa de traducción dirigida por sintaxis.
- **Capítulo 3. Paradigmas y modelos de programación.** Este capítulo define el concepto de paradigma de programación y describe brevemente algunos de los paradigmas existentes: imperativo, orientado a objetos, concurrente, funcional y lógico. Para introducir cada paradigma se utiliza un lenguaje representativo del mismo del cual se explican los conceptos sintácticos fundamentales para comprender el paradigma y poder realizar ejercicios sencillos. El capítulo también presenta otros lenguajes de computadora que a pesar de que no representan un paradigma concreto, son de un uso tan generalizado que merecen al menos una breve introducción: los lenguajes dinámicos y los lenguajes de marcado. Los ejercicios de este capítulo buscan reforzar el conocimiento del lector sobre cada uno de los paradigmas y lenguajes presentados.
- **Capítulo 4. Lenguajes de marcado. XML.** Este capítulo introduce al lector en los fundamentos del lenguaje de marcado XML. Se presentan las principales tecnologías asociadas para que el lector conozca sus posibilidades y principales características. Además se presentan los procesadores de documentos XML en términos de los tipos de análisis que realizan. Los ejemplos, ejercicios resueltos, junto con el uso de las herramientas recomendadas en el propio capítulo permitirán al lector familiarizarse con este lenguaje de marcado.
- **Capítulo 5. Lenguajes de *script*.** Este capítulo introduce los lenguajes de *script* comenzando con una breve descripción de sus orígenes. La presentación de los dominios en los que normalmente se utilizan sirve de marco para presentar y comentar algunas de las herramientas y lenguajes de *script* más populares. En este capítulo tienen especial relevancia los ejercicios resueltos ya que son los que van a mostrar la potencia y diferencias de este tipo de lenguajes con respecto a otros lenguajes de programación. Se tratan dos lenguajes de *script* con un poco más de detalle, Perl y PHP, y en particular este último se relaciona con XML y algunos procesadores de documentos.
- **Capítulo 6. Aspectos pragmáticos de los lenguajes de programación.** Este capítulo ofrece al lector algunos aspectos que pueden resultar clave en la elección de un determinado lenguaje frente a otro, a la hora de crear un programa infor-

mático, o de estudiar la interoperabilidad entre aplicaciones escritas en diferentes lenguajes de programación.

## Dependencias entre capítulos

Para el lector interesado en todos los contenidos abordados en el libro se recomienda una lectura secuencial siguiendo el orden de aparición de los capítulos. En el caso de que se quieran abordar sólo algunos de los temas aquí presentados o hacerlo en un orden diferente del aquí expuesto, se detallan a continuación las dependencias entre los distintos capítulos:

- El capítulo 3 depende del capítulo 1, dado que en éste se introducen conceptos que son utilizados a la hora de presentar los diferentes paradigmas, como el ámbito de variables, el concepto de subprograma, tipos de datos, etc.
- El capítulo 2 depende también del capítulo 1, puesto que se centra en los procesadores de lenguajes y se utiliza la notación BNF para explicar los conceptos relativos al análisis sintáctico.
- El capítulo 6 hace referencia al Capítulo 1 al presentar los aspectos pragmáticos de los lenguajes de programación. Aunque el capítulo 6 podría verse después del 3, su lectura después de los capítulos 4 y 5 proporciona un panorama más amplio de los lenguajes de programación.
- El capítulo 4, dedicado a los lenguajes de marcado, se podría abordar de manera independiente del resto de capítulos, sin embargo su estudio una vez leída la primera parte del libro (capítulos 1 y 2) facilita la comprensión en profundidad de las posibilidades de XML y de sus tecnologías.
- Se podría decir que el capítulo 5 profundiza en algunos de los conceptos presentados en el capítulo 3, y por lo tanto se recomienda la lectura previa de éste.

Al lector interesado solo en la teoría de los lenguajes de programación y procesadores de lenguajes se le recomienda la lectura de los capítulos 1, 2, 3 y 6 por ese orden. Si además tiene interés en los lenguajes de *script* puede incluir el capítulo 5 antes del 6.

## Ejercicios

En cada capítulo hay un apartado de **ejercicios resueltos** y otro de **ejercicios propuestos**. El primero proporciona al lector ejemplos de cómo resolver cuestiones teóricas, problemas, utilizar ciertos lenguajes o tecnologías que cubren los aspectos fundamentales desarrollados en el capítulo.

Además, los ejercicios resueltos constituyen una herramienta fundamental para la auto-evaluación en el caso de que el lector quiera comprobar su grado de comprensión de los contenidos expuestos en cada capítulo.

Los ejercicios propuestos también cubren los aspectos fundamentales del capítulo. Estos ejercicios se dejan abiertos como un reto para el lector para que los intente solucionar. Los autores animamos al lector a la lectura y resolución de ambos tipos de ejercicios.