Programación en FP

Guía de supervivencia

Luis del Moral Martínez

Contribuye al ahorro de papel. No imprimas el libro si no es necesario.

Versión 1.0 (febrero de 2021)

Índice general

Índice de figuras

Índice de cuadros

Introducción a la programación

1. Presentación

En este capítulo se introduce el propósito y el contenido de esta obra, detallando cómo está organizada cada parte y capítulo. Además, se presenta una serie de recomendaciones previas antes de iniciar el resto de capítulos. Se recomienda leer este capítulo de forma completa antes de pasar al resto de contenidos del libro.

1.1 Te doy la bienvenida

Si tienes este libro entre manos, o puede que abierto en la pantalla de tu ordenador, quiero darte la bienvenida al que espero que sea un curso muy ameno y productivo sobre programación. A lo largo de este libro espero que desarrollemos técnicas efectivas para que aprendas a programar aplicaciones informáticas. En este breve capítulo quiero explicarte, en primer lugar, el propósito de esta obra, ofrecerte una serie de consejos bastante importantes antes de entrar en materia y, por supuesto, explicarte cómo está organizado el libro. Si ya has programado antes o tienes alguna noción básica sobre programación, puedes echar un vistazo al índice y comenzar por el capítulo que prefieras, pero yo te recomiendo que empecemos desde el principio.

1.2 Propósito del libro

Este libro pretende enseñarte a programar desde cero. Es por esto por lo que sus contenidos se han pensado para abarcar todas las fases de la creación de un programa, suponiendo que no tienes ningún conocimiento previo sobre programación. Así que no te preocupes si no has programado nunca, y mucho menos si acabas de aterrizar en un ciclo formativo de informática y empiezas a sentir cierto temor sobre la asignatura de programación. Todos los contenidos que contiene este libro, así como los de las futuras mejoras y versiones, se enfocan a las asignaturas de programación del primer curso de los ciclos formativos de grado superior en **DAM** (Desarrollo de Aplicaciones Multiplataforma)

y **DAW** (Desarrollo de Aplicaciones Web). Además, pretenden cumplir con los currículos establecidos en toda la normativa vigente, teniendo siempre en cuenta el enfoque práctico que tiene la formación profesional, así como la orientación hacia las prácticas y las necesidades que tienen las empresas hoy en día.

Dicho esto, y espero que con casi todos los miedos desterrados, debo decirte que pretendo ser escueto y práctico en todos los conceptos que estudiemos, acompañándote en cada paso y recomendado siempre la mejor estrategia que debes seguir. También voy a evitar irme por las ramas con definiciones enrevesadas o proponiéndote decenas de enlaces con información adicional. Mi objetivo al escribir este libro es muy claro, y es que pretendo que puedas superar esta asignatura con los conocimientos y conceptos que vamos a desarrollar entre estas páginas. En la actualidad hay demasiadas fuentes de información y esto sobrecarga bastante a muchos estudiantes, impidiendo que puedan seleccionar la información más útil y descartar la información tediosa o redundante.

1.3 Antes de comenzar

Es importante que tengamos en cuenta una serie de recomendaciones antes de que empecemos a programar. Si bien es cierto que repito estos consejos a mis estudiantes casi todos los días, es posible que algunos te funcionen mejor que otros. Y esto se debe a que no hay dos personas iguales, desde luego. Es posible que tengas unas dotes innatas para la programación, o puede que te atranques con los primeros enunciados. Tampoco te preocupes si ese es el caso, porque los consejos que te muestro a continuación los vamos a practicar hasta la saciedad, y te garantizo que con esfuerzo serás capaz de superar esta asignatura y aprender a programar con bastante soltura:

- **Progreso, no perfección**: cuando se empieza en una nueva disciplina tenemos que evitar pensar en la perfección. Siempre se suele comenzar pensando en cómo puedo perfeccionar esta o aquella destreza o qué pasos necesito para llegar a dominar esta tecnología. En mi opinión, esto es un error. Hay que trabajar y esforzarse (sin llegar a perder de vista nuestro objetivo, desde luego). De nuevo, trabajo y esfuerzo son los ingredientes básicos para dominar cualquier destreza, y la programación no va a ser menos.
- **Programar, programar y programar**: a programar se aprende programando. Igual que a montar en bicicleta se aprende usando la bicicleta. He leído muchas entrevistas a escritores profesionales en las que les pedían que por favor dijeran las claves de su éxito. La mayoría siempre respondía: «a escribir se aprende escribiendo».
- **Leer blogs de tecnología y programación**: nos permitirán estar al día sobre las nuevas tendencias en el mundo de la programación. Este consejo es extensible más allá de la tecnología. Es importante leer, y mucho. Como programador, vas a colaborar en el desarrollo de aplicaciones que resuelvan el problema de un cliente. Y leer mucho, y de calidad, te permitirá expresarte mejor y comprender mucho mejor al cliente. Además, leer abrirá tu mente y te facilitará cualquier proceso de aprendizaje (ya sea en el mundo de la informática o en otro campo de estudio).
- **Dibujar los problemas**: este consejo está relacionado con los siguientes capítulos. Es muy importante pensar en el problema que nos plantean, entenderlo, y ser capaz de dibujarlo (esto es, como aprenderás dentro de pocas páginas, la capacidad de diseñar un algoritmo, o un conjunto de pasos, que resuelvan dicho problema). Siempre les

pido a mis estudiantes que vengan a los exámenes con folios de sobra, lápices, e incluso gomas de borrar. Resulta una técnica muy útil para analizar los ejercicios.

- **Leer código de otras personas**: es un buen método para aprender a programar y, por supuesto, para mejorar la calidad de nuestro código. De esta forma podemos aprender buenas técnicas de programación. También puede picarte el gusanillo y quizás acabes colaborando en un proyecto de **software libre**[1], lo que sería estupendo, de hecho.
- **Comentar nuestro código**: los lenguajes de programación (todos ellos) tienen herramientas que permiten documentar el código. Esto permite a los programadores, además de a cualquier persona que se tope con el código en el futuro, comprender el código y saber exactamente qué es lo que se está haciendo en cada momento. Llegado el momento te comentaré las mejores técnicas sobre comentarios, y es que hay que adoptar un término medio. No conviene documentar hasta la extenuación; pero es importante incluir comentarios, o de lo contrario correrás el riesgo de retomar tu código en el futuro y lo pasarás mal hasta recordar exactamente qué hacía o para qué sirve (créeme, me ha pasado unas cuantas veces, y al final se podría haber evitado con comentarios estratégicamente distribuidos por el código).
- **Usar un entorno de desarrollo (IDE)**[2]: los entornos de desarrollo son una de las herramientas indispensables de un programador (te imaginas a un mecánico trabajando sin herramientas). Llegado el momento te enseñaré a configurar y a utilizar los entornos de desarrollo **Visual Studio Code** y **Eclipse**.
- **Revisar y probar el código**: este consejo es importantísimo. En clase y en los exámenes siempre suelo soltar la frase «llevamos mucho tiempo programando, es hora de probar nuestro código». Resulta muy importante probar lo que estamos programando, no sólo para ver que funciona (también hablaremos sobre los **errores** y las **excepciones** y cómo tratarlos; todo a su debido tiempo), sino para comprobar que estamos cumpliendo con el enunciado y que se está resolviendo el problema que se nos pide.

A estos consejos hay que sumar siempre una predisposición a practicar y *trastear* con el lenguaje de programación. Es cierto que esta asignatura puede ser compleja al inicio, pero no hay nada como experimentar con los conceptos recién aprendidos en un lenguaje de programación para asentar nuestros conocimientos. Dicho esto, también te recomiendo, además de todo lo anterior, que procures repasar todos los días o, al menos, hacer pequeños ejercicios donde pongas a prueba todo lo aprendido (ensayos del tipo ¿qué pasaría si hago esto? o ¿qué sucede si hago el programa de esta forma?).

1.4 Organización del libro

Antes de que entremos en materia me gustaría explicarte cómo se organiza este libro y detallar algunos de los elementos claves que usaremos a lo largo de los diferentes capítulos

[1] Es un tipo de software que se distribuye con una licencia que permite que los usuarios puedan ejecutarlo, copiarlo, distribuirlo, estudiarlo, modificarlo y mejorarlo. Este libro, por ejemplo, se distribuye bajo licencia libre, lo que te permite modificarlo y contribuir a su edición, si lo deseas

[2] Un **IDE** (Integrated Development Environment) es un entorno de desarrollo integrado. Su principal objetivo es el de facilitar la labor de desarrollo de aplicaciones. Algunos de los entornos más populares son *Eclipse*, *Visual Studio*, *Visual Studio Code*, *Atom*, *Sublime* o *Notepad++*, entre otros. Muy pronto configurarás tu primer IDE para empezar a programar en el lenguaje C/C++.

de esta obra. A continuación te detallaré someramente el contenido de cada una de las partes del resto del libro (como sabes, puedes comenzar por el capítulo que prefieras, pero te aconsejo que empieces en orden):

- **Primera parte**
  - **Capítulo 2. ¿Dónde me he metido**: en este capítulo hablaremos de qué significa programar y, sobre todo, qué es lo que se espera de ti en esta asignatura y en un ciclo de desarrollo de aplicaciones (**DAM** o **DAW**).
  - **Capítulo 3. ¿Cómo funciona un ordenador?**: este capítulo te ayudará a comprender los procesos básicos que permiten que funcione un ordenador, entrando únicamente en los detalles que necesitas comprender para programar.
  - **Capítulo 4. ¿Qué es un algoritmo**: detalla los pasos que hay que llevar a cabo para analizar un problema y crear un algoritmo que lo resuelva. Este capítulo incorpora un conjunto de herramientas muy útiles para enfrentarse a cualquier problema futuro. Es recomendable que vuelvas sobre tus pasos y regreses a este capítulo cuando sea preciso.

Otro aspecto importante antes de comenzar es la notación que usaremos en el libro. Esta será bastante sencilla, puesto que vamos a destacar dos elementos fundamentales que te acompañarán a lo largo de los capítulos: los **ejemplos** y los **consejos**. Mientras que en los **ejemplos** voy a incluir ciertos códigos o figuras que te ayudarán a comprender mejor los conceptos, en los **consejos** te detallaré y detalles adicionales y buenas prácticas que no puedes pasar por alto.

Ejemplo 1.1 Esto es un ejemplo de código fuente en C/C++:

```
#include<stdio.h>
#include<iostream>

// Esto es un comentario
int main(void)
{
  printf("Imprimiendo por consola\n");
  return 0;
}
```

Consejo Las palabras reservadas de un lenguaje siempre deben escribirse en minúsculas. Ejemplo: **int**, **void**, **for**, **if**...

En las futuras versiones de este libro (y sí hablo de versión y no de *edición* porque pretendo y espero que este libro evolucione a lo largo del tiempo, como si de una aplicación informática se tratara) espero incorporar una página web o un repositorio de **GitHub**[3] donde aloje código fuente y ejemplos complementarios a los que se recogen en este documento.

[3] GitHub es una empresa que proporciona servicios de alojamiento de repositorios de código que emplean la tecnología de control de versiones de . No te preocupes ahora por este término. Lo comentaremos en su debido momento y te enseñaré todo lo que debes saber para hacer uso del mismo.

2. ¿Dónde me he metido?

En este capítulo se explica en qué consiste programar y la importancia que tiene la programación en nuestro día a día. También se relaciona la asignatura de programación con otras materias de **Desarrollo de Aplicaciones Multiplataforma** (DAM) y **Desarrollo de Aplicaciones Web** (DAW), de forma que se comprenda la importancia que tiene aprender a programar de forma correcta antes de abordar contenidos o materias más complejas de ambos ciclos formativos.

2.1 ¿En qué consiste programar?

Hay personas que dicen que la programación es un arte, y también podría darte toda serie de definiciones sobre lo que significa programar. Sin embargo, voy a ser lo más simple y directo posible. Programar consiste en usar un **lenguaje de programación** para desarrollar un **programa informático**.

Un **lenguaje de programación** es un conjunto de *reglas*, *expresiones* y *símbolos* que permiten crear una secuencia de *instrucciones* que sean entendidas por un ordenador. De esta forma, los programadores usamos un lenguaje de programación para dar solución a un determinado *problema*, con lo que se obtiene un **programa informático**. Dicho programa manipula una serie de **datos** y genera unos **resultados** (que pueden almacenarse como un texto, una imagen, un documento pdf, una página web...).

Muy pronto te enseñaré las técnicas básicas que debes seguir para resolver un problema (como por ejemplo el pseudocódigo y los algoritmos) y empezarás a escribir tus propios programas. Verás que no es tan difícil como te hayan contado, ni mucho menos. Sin embargo, por el momento te pido que reflexiones en los sistemas informáticos que usas a diario. Piensa que los programas que usas en tu teléfono móvil, por ejemplo, se han desarrollado con el objetivo de resolver un problema (tomar fotografías, enviar mensajes por internet, compartir información, realizar una videollamada). Y a su vez, todos ellos se han programado usando un lenguaje de programación.

2.2 ¿Qué importancia tiene la programación?

Mira a tu alrededor. Electrodomésticos, teléfonos inteligentes, tablets, televisores, videoconsolas, ordenadores, coches, sistemas de control del tráfico... Podría seguir, pero basta con decir que los sistemas informáticos están por todas partes. Aunque aún no hemos explicado cómo funciona un sistema informático (trataremos este tema en el siguiente capítulo), si te puedo adelantar que todos funcionan gracias a la programación.

Nuestro día a día, tal y como lo conocemos, no puede existir sin la informática y sin los programas informáticos. De hecho, la programación se ha convertido en una disciplina muy importante y en una aptitud deseable para casi cualquier puesto de trabajo.

En la actualidad se aventura que muchos puestos de trabajo futuros se verán destruidos debido a la programación y la **inteligencia artificial**. También es posible que el trabajo al que te dediques en el futuro aún no exista, debido al constante cambio tecnológico en el que nos vemos inmersos. Por todas estas razones es muy importante que aprendas a programar. Y sin duda, debes hacerlo de la forma más eficaz y correcta posible.

2.3 Relación con otras materias

La programación de aplicaciones está relacionada casi con todas las materias de los ciclos formativos de **Desarrollo de Aplicaciones Multiplataforma** (DAM) y **Desarrollo de Aplicaciones Web** (DAW), puesto que el propósito de ambos ciclos formativos es el de capacitar a los estudiantes para que desarrollen aplicaciones software (ya sean multiplataforma[1] u orientadas a la web).

Es por esto por lo que saber programar, como adelantábamos en la sección anterior, es de vital importancia en la actualidad y, de hecho, muchos expertos sostienen que el aprendizaje de técnicas de programación debería comenzar mucho antes, incluso en bachillerato o puede que en la ESO. Y es que crear una aplicación informática no sólo consiste en «picar código», o en crear una **interfaz de usuario** muy bonita. Consiste en un proceso que abarca desde la comprensión de los requisitos del cliente hasta la especificación y el diseño de una solución, su implementación, testeo y posterior despliegue.

Por otra parte, cuando los estudiantes llegan al clase siempre creen que van a empezar a desarrollar videojuegos y aplicaciones alucinantes desde el primer día. Pero siento decirte que esto no es posible sin sentar unas bases sólidas sobre la programación estructurada y la programación orientada a objetos, proceso que comenzaremos en la segunda y tercera parte de este libro, respectivamente. En los siguientes capítulos hablaremos sobre cómo funciona un ordenador y el concepto de **algoritmo**. Te recomiendo que continúes leyendo antes de pasar a la acción en la parte 2, con los lenguajes de programación C y C++, puesto que aprenderás los conceptos básicos sobre el funcionamiento de un ordenador y técnicas de resolución de problemas usando algoritmos, lo que te aportará el conocimiento suficiente para empezar a escribir tus primeros programas informáticos.

2.4 Ejercicios propuestos

Completa tu aprendizaje realizando los siguientes ejercicios propuestos:

[1] Que pueden ejecutarse de forma compatible en diferentes sistemas hardware

Ejercicio 2.1 Visualiza los siguientes videos de **Youtube** y reflexiona sobre las necesidades de programación que existen en la actualidad:

- Todo el mundo debería saber programar
- ¿Por qué hay que aprender a programar?

■

Ejercicio 2.2 Investiga en los siguientes portales de búsqueda de empleo acerca de la cantidad y número de ofertas de trabajo que existen en la actualidad relacionadas con la programación (deberás investigar también acerca de los diferentes roles o puestos que se ofertan relacionados con la programación de aplicaciones):

- Tecnoempleo
- Infojobs
- Linkedin

■

3. ¿Cómo funciona un ordenador?

3.1 ¿Qué es un ordenador?

3.2 Estructura de un ordenador

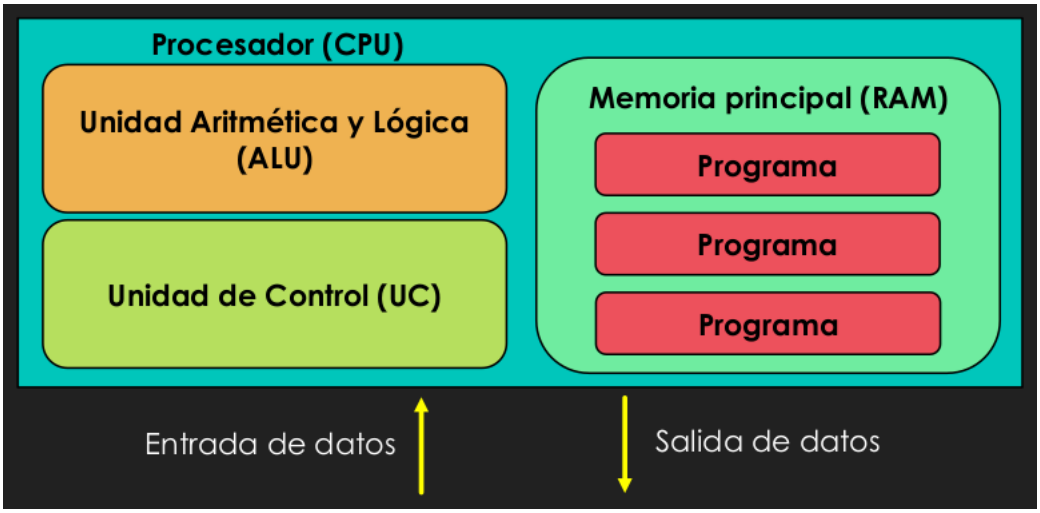

Figura 3.1: Diagrama de un ordenador

1 TB = 1024 GB = 1024 x 1024 Mb = 1.048.576 KB = 1.073.741.824 Bytes

3.3 Representación de la información

3.4 Ejercicios propuestos

Completa tu aprendizaje realizando los siguientes ejercicios propuestos:

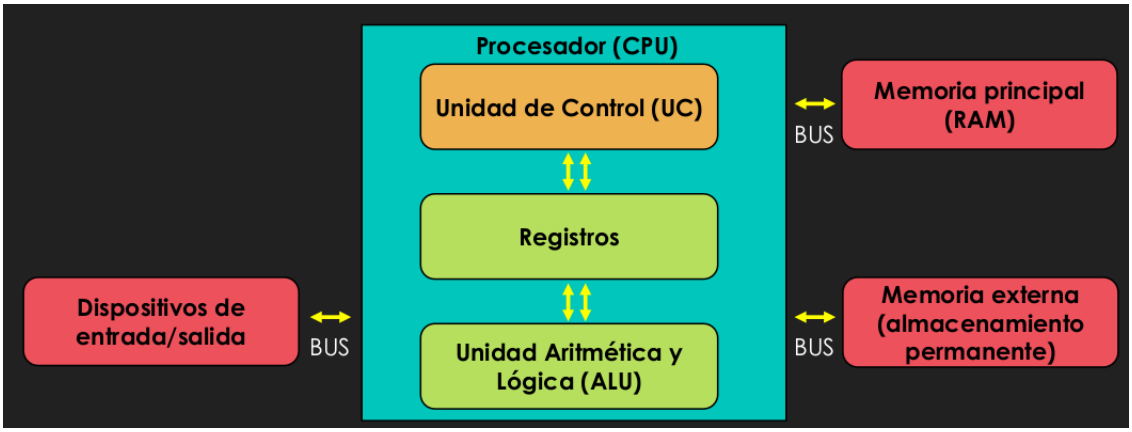

Figura 3.2: Esquema de la CPU y los buses del sistema

Unidad	Equivalencia
Byte (B)	8 bits
Kilobyte (KB)	1024 bytes
Megabyte (MB)	1024 Kbytes
Gigabyte (GB)	1024 Mbytes
Terabyte (TB)	1024 Gbytes
Petabyte (PB)	1024 Tbytes

Cuadro 3.1: Unidades de medida de almacenamiento

4. ¿Qué es un algoritmo?

Completa tu aprendizaje realizando los siguientes ejercicios propuestos:

Ejercicio 4.1 Bla bla bla ■

Ejercicio 4.2 Bla bla bla ■

Ejercicio 4.3 Bla bla bla ■

Ejercicio 4.4 Bla bla bla ■

II

Programación en C/C++

III

POO en Java

Bibliografia e índice alfabético

Bibliografía

Libros de referencia

Índice alfabético