



Fascículo 4

Programación en diversos lenguajes

Cuaderno 3: Saberes y tareas del programador

Perfil del programador

Hicimos un recorrido por los principales conceptos a tener en cuenta a la hora de introducirnos en el mundo de la programación y sus distintos lenguajes. Conozcamos ahora cuál es el perfil profesional de un programador, cuáles son sus saberes y qué actividades realiza.

El **programador** es quien, escribe, depura y mantiene el código fuente de un lenguaje informático, es decir, el conjunto de instrucciones que ejecuta una computadora para realizar una tarea determinada. Los programadores también reciben el nombre de desarrolladores de software.

Como hemos visto, el programador escribe un programa en un lenguaje de alto nivel que es interpretado y traducido a bytes –código binario– que la computadora puede comprender.

En la jerga técnica se dice que el **programador genera el código fuente y el programa intérprete genera el código objeto**, que también se denomina código binario o código máquina. Un programador se encarga de implementar algoritmos mediante un lenguaje de programación. Actualmente el término se asocia también al **analista**, porque a nivel práctico es complicado diferenciar las tareas de ambos actores en el desarrollo de un software.

El programador desarrolla los programas pensando en el **usuario**, en las personas que lo van a utilizar. No es lo mismo desarrollar un programa sencillo con funciones básicas que un programa complejo destinado, por ejemplo, a ingenieros aeroespaciales.

El usuario accede en general al programa a través de un **archivo ejecutable** que se instala en la PC y puede ser ejecutado repetidas veces para realizar una tarea determinada. Por ejemplo: los usuarios utilizan el programa Word para escribir textos.

El programador en cambio, accede a un programa desde un **archivo de texto** con instrucciones a la computadora escritas en un determinado lenguaje de programación, que luego podrá convertirse en un ejecutable.

Pero para que un programa pueda llevarse a cabo, independientemente del lenguaje en que se desarrolle, se requiere un equipo profesional, tiempo de investigación, pruebas y recursos.





Además de tener en cuenta al usuario o destinatario del programa que se desarrolla, se debe considerar el lenguaje –en la jerga de programación es también llamado plataforma– en la que será ejecutado el programa. Se denomina **plataforma** porque es precisamente la base, el principio sobre el cual se constituye un hardware, y sobre el cual un programa puede ejecutarse o desarrollarse.

A lo largo de sus carreras, los programadores se van especializando en distintos lenguajes –por ejemplo PHP, Java, Python, etc.– y en distintas plataformas, y conforman equipos de trabajo, en general compuestos por diseñadores, analistas e ingenieros.

Inicialmente, la profesión de programador se formalizó desde el enfoque tayloriano –caracterizado por la división del trabajo–, donde ejercían poco poder de decisión y su aporte era específicamente técnico.

El proceso de producción de un programa se concebía como un conjunto de **tareas** altamente **especializadas** en el cual estaba claramente definido el papel de cada categoría profesional:

- Los **diseñadores** se especializaban en la arquitectura del programa: el diseño conceptual y la estructura operacional fundamental de un sistema de computadora. Es decir, del modelo y la descripción funcional de los requerimientos y las implementaciones de diseño para varias partes del programa.
- El **analista** estudiaba un problema y lo describía con el propósito de buscar su solución.
- El **programador** trasladaba las especificaciones del analista en código ejecutable por la computadora. Dichas especificaciones se recogían en un documento denominado “cuaderno de carga”, que oficiaba de medio de comunicación entre ambos.

Esta concepción ha ido evolucionando y este enfoque ya no resulta válido para organizar tareas de tipo intelectual como la producción de programas. Las dificultades de comunicación entre analistas y programadores –un mero documento no bastaba para describir lo que se quería hacer– dieron origen a una categoría profesional intermedia, denominada **analista-programador**.

La concepción original del programador desaparece y se sustituye por la de un profesional mucho más formado, con funciones y tareas menos mecánicas.

La profesión de analista también ha evolucionado debido a los avances de la ingeniería del software, surgiendo el concepto de diseñador de programas, con lo que se reconoce que **el análisis es una actividad distinta del diseño**. El **análisis** describe el **problema** (el qué hacer) mientras que el **diseño** describe la **solución** (el cómo hacerlo).

Con el auge de las computadoras y la diversificación de funciones que se le





requieren, la profesión de programador se ha convertido en una especialización de alta calificación. Hoy en día, un programador debe tener profundos conocimientos que incluyen matemática, electrónica, algorítmica, ingeniería de software, teoría de la computación, interacción persona-ordenador, etcétera.

Si bien la profesión de programador no cuenta con especialidades en los niveles académicos, podría decirse que las diversas ramas que existen se fueron dando a partir de las ofertas de empleo del mercado laboral.

Veamos aquí las más frecuentes:

- **Programadores de mainframe:** se orienta hacia los lenguajes de programación más antiguos, de los que aún existen muchos en funcionamiento que requieren mantenimiento y actualización. Entre los conocimientos que se requieren se cuenta COBOL, JCL (*Job Control Language*) y bases de datos jerárquicas, entre otros.
- **Programadores de nuevas tecnologías:** esta rama gira principalmente en torno a internet y sus requerimientos: la Web 2.0, comunidades virtuales o redes sociales y los negocios por medios electrónicos o e-commerce. Entre sus conocimientos se destacan lenguajes del lado del servidor como Java, ASP, .NET, JSP, PHP, Ruby, Python, o Perl, y lenguajes del lado de cliente como HTML, XHTML, CSS, Javascript ó AYAX (conjunto de tecnologías existentes como XML y Javascript).
- **Programadores de firmware y videojuegos:** con conocimientos de hardware, microprocesadores, ensamblador y C.
- **Programadores de sistemas abiertos:** rama asociada a programas compatibles con distintas estructuras. Requiere conocimientos de C, Pascal, etc.

Además de estas áreas de interés y desarrollo para el programador existen ciertas **normas éticas** implícitas en la profesión. La más importante establece que el programador debe ante todo perseguir la obtención de **programas de calidad**. Para ello se establecen una serie de factores que determinan la calidad de un programa.

Algunos de los factores de calidad más importantes son los siguientes:

- **Corrección**
Un programa es correcto si hace lo que debe hacer tal y como se estableció en las fases previas a su desarrollo. Para determinar si un programa hace lo que debe es muy importante especificar claramente qué debe hacer el programa antes de desarrollarlo y, una vez acabado, compararlo con lo que realmente hace.
- **Claridad**
Es muy importante que el programa sea lo más claro y legible posible para facilitar así su desarrollo y posterior mantenimiento. Al elaborar un programa debe intentarse que su estructura sea sencilla y coherente, así como cuidar el estilo en la edición: de esta forma se ve facilitado el trabajo del programador, tanto en la fase de creación como en las posteriores de corrección de errores, ampliaciones, modificaciones, etc. Estas fases pueden ser incluso realizadas por





otro programador, con lo cual la claridad es aún más necesaria para que pueda continuar el trabajo fácilmente.

■ **Eficiencia**

Se trata de que el programa, además de realizar aquello para lo que fue creado, lo haga gestionando de la mejor forma posible los recursos que utiliza. Normalmente, al hablar de eficiencia de un programa se suele hacer referencia al tiempo que tarda en realizar la tarea para la que ha sido creado y a la cantidad de memoria que necesita. Sin embargo, hay otros recursos que también pueden ser de consideración para evaluar la eficiencia de un programa, dependiendo de su naturaleza (espacio en disco que utiliza, tráfico de red que genera, etc.).

■ **Portabilidad**

Un programa es portable cuando tiene la capacidad de ejecutarse en una plataforma, ya sea hardware o software, diferente de aquella en la que se elaboró. Por ejemplo, permite a un programa que se ha desarrollado para sistemas GNU/Linux ejecutarse también en la familia de sistemas operativos Windows. Esto hace que el programa pueda llegar a una mayor cantidad de usuarios más fácilmente.

En síntesis, el programador tiene como función especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias.

Todo esto a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. Una característica relevante de los lenguajes de programación es precisamente que más de un programador pueda usar un conjunto común de instrucciones que sean comprendidas entre ellos. Esto facilita los trabajos en equipo y la **programación colaborativa**.

Programación colaborativa y software libre

La **programación en equipo** es el esfuerzo coordinado de programadores que se dividen tareas. La **programación colaborativa**, en cambio, significa que dos o más programadores trabajan en forma conjunta sobre un mismo código o algoritmo.

Esta colaboración es potenciada por el auge de internet, que permite que varios programadores conecten sus estaciones de trabajo a una red de datos y trabajen de manera participativa.

Entre los años 60 y 70 del siglo xx, el software era un añadido que los vendedores de las grandes computadoras de la época aportaban a sus clientes para que estos pudieran usarlas. En dicha cultura, era común que los programadores y desarrolladores compartieran libremente sus programas. Pero a finales de los 70, las compañías iniciaron el hábito de imponer restricciones a los usuarios, con acuerdos de licencia. Con la llegada de los años 80 la situación empezó a cambiar. Las computadoras más modernas comenzaban a utilizar **sistemas operativos**





privativos, forzando a los usuarios a aceptar condiciones restrictivas que impedían realizar modificaciones al software.

Con este antecedente, Richard Stallman –principal referente del **software libre**– fundó en 1985 la Free Software Foundation (FSF) e introdujo por primera vez el término *free software* (programa libre) y el concepto de *copyleft* (libre de derechos), que desarrolló para otorgar libertad a los usuarios y para restringir las posibilidades de apropiación del software.

Para que un software sea libre debe permitirse no solo **usarlo**, también **adaptarlo**, acceder al código fuente, hacer **copias** y **compartirlo**. Se basa en la **colaboración** y no en la competencia.

Ethercodes es un proyecto de programación colaborativa que permite, sin necesidad de registro, escribir código en varios lenguajes de programación y compartirlo en tiempo real con cualquier persona. No es necesario instalar nada, solo hay que acceder a su panel de trabajo para poder elegir entre C/C++, Java, Javascript, PHP, HTML, Python, Ruby y hasta Mysql, con posibilidad de acceder a una línea de tiempo que mostrará, en forma de video, las alteraciones realizadas por quienes están programando.

Si les interesa profundizar sobre software libre, pueden hacerlo en estas páginas sugeridas:

- [Asociación Civil Software Libre Argentina](#)
- [Hacia la sociedad del conocimiento libre](#)
- [Información sobre software libre y nuevas tecnologías](#)

Si les parece interesante todo lo que vimos hasta aquí y quieren saber en qué lugares de la Argentina se puede estudiar la carrera de programador, pueden consultar el siguiente cuadro sobre las distintas carreras relacionadas con la programación, su tiempo de duración y las universidades en las que se dictan.

Fuente Universidades.org

Universidad	Carrera	Duración
Universidad Católica de Santiago del Estero	Analista de Sistemas	4 años
Universidad de Buenos Aires	Licenciatura en Análisis de Sistemas	3 años y medio
Universidad Tecnológica Nacional	Ingeniería en Sistemas	5 años
Universidad de Palermo	Licenciatura en Redes y Comunicación de Datos	4 años
Universidad de Palermo	Licenciatura en Sistemas y Gestión	4 años





Universidad	Carrera	Duración
<u>Universidad Nacional de la Patagonia Austral</u>	Analista de Sistemas	3 años
<u>Universidad Nacional de la Patagonia Austral</u>	Analista de Sistemas	3 años
<u>Universidad Nacional de la Patagonia Austral</u>	Analista de Sistemas	3 años
<u>Universidad Nacional de la Patagonia Austral</u>	Analista de Sistemas	3 años
<u>Universidad Nacional de la Patagonia San Juan Bosco</u>	Analista Programador Universitario	3 años
<u>Universidad Nacional de la Patagonia San Juan Bosco</u>	Analista Programador Universitario	3 años
<u>Universidad Nacional de la Patagonia San Juan Bosco</u>	Analista Programador Universitario	3 años
<u>Universidad Nacional de la Plata</u>	Analista en Computación	
<u>Universidad Nacional de la Plata</u>	Licenciatura en Informática	
<u>Universidad Nacional de la Rioja</u>	Analista Universitario en Sistemas.	4 años
<u>Universidad Nacional de la Rioja</u>	Programador Universitario en Computación	3 años
<u>Universidad Nacional de la Rioja</u>	Licenciatura en Análisis de Sistemas	5 años
<u>Universidad Nacional de Comahue</u>	Analista en Comunicación	3 años
<u>Universitas</u>	Analista de Sistemas	3 años
<u>Universitas</u>	Analista de Sistemas	3 años
<u>Institutos de Tecnología ORT</u>	Analista de Sistemas de Computación	3 años
<u>Universidad Nacional de Misiones</u>	Analista en Sistemas de Computación.	3 años
<u>Centro de Altos Estudios de Informática</u>	Analista de Sistemas	
<u>Universidad Nacional de San Juan</u>	Programador	3 años
<u>Ateneo de Estudios Superiores</u>	Sistemas	3 años

Fuentes

- www.wikipedia.org



Autora: María Lorena Suárez
Coordinación editorial: Mara Mobilia

