

INTRODUCCIÓN A LA PROGRAMACIÓN CON

# PYTHON

Algoritmos y lógica de programación para principiantes

**Nilo Ney Coutinho Menezes**

Edición original en portugués del libro *Introdução à Programação com Python 2ª edição*, ISBN 9788575224083, publicada por Novatec Editora Ltda. © 2014 Novatec Editora Ltda.

Traducción al Español realizada por Novatec Editora. © 2016 Novatec Editora Ltda.  
Todos los derechos reservados.

Esta traducción es impresa y vendida con el permiso de Novatec Editora, poseedora de todos los derechos para publicación y venta de esta obra. Está prohibida la reproducción de esta obra, aun parcial, por cualquier proceso, sin previa autorización, por escrito, del autor y de la Editorial.

Editor: Rubens Prates  
Traducción al español: Ricardo Pérez Banega  
Revisión gramatical: Pilar Domingo  
Edición electrónica: Camila Kuwabata  
Tapa: Victor Bittow

ISBN: 978-85-7522-513-4

Historia de impresiones:  
Julio/2016                      Primera edición

Novatec Editora Ltda.  
Rua Luís Antônio dos Santos 110  
02460-000 – São Paulo, SP – Brasil  
Tel.: +55 11 2959-6529  
E-mail: [novatec@novatec.com.br](mailto:novatec@novatec.com.br)  
Sitio: [novatec.com.br](http://novatec.com.br)  
Twitter: [twitter.com/novateceditora](https://twitter.com/novateceditora)  
Facebook: [facebook.com/novatec](https://facebook.com/novatec)  
LinkedIn: [linkedin.com/in/novatec](https://linkedin.com/in/novatec)

Edición en español impresa y distribuida por:  
Coutinho Menezes Nilo - LogiKraft  
Rue de la Grande Campagne, 40  
7340 Wasmès  
Belgium  
+32 485 251460  
[libros@logikraft.be](mailto:libros@logikraft.be)

# CAPÍTULO 1

## Motivación

*Entonces, ¿Ud. quiere aprender a programar?*

Programar computadoras es una tarea cuyo aprendizaje cabal exige tiempo y dedicación. Muchas veces no basta solo con estudiar y hacer los ejercicios, sino que también es necesario dejar que la mente se acostumbre a la nueva forma de pensar. Para muchas personas lo más difícil es mantener el entusiasmo por programar. Desisten ante las primeras dificultades y no vuelven más a estudiar. Otras personas son más pacientes, aprenden a no irritarse con la máquina y a asumir sus errores.

Para no sufrir de los males del que no aprendió a programar, es necesario que responda algunas preguntas antes de empezar:

1. ¿Ud. quiere aprender a programar?
2. ¿Cuál es su nivel de paciencia?
3. ¿Cuánto tiempo pretende estudiar?
4. ¿Cuál es su objetivo al programar?

### **1.1 ¿Ud. quiere aprender a programar?**

Responda esta pregunta, pero piense un poco antes de llegar a la respuesta final. La manera más difícil de aprender a programar es no querer programar. El deseo debe venir de usted y no de un profesor o de un amigo. Programar es un arte y necesita dedicación para ser dominado. Como todo lo desconocido, es muy difícil cuando no lo entendemos, pero se vuelve más simple a medida que lo aprendemos.

Si ya decidió aprender a programar, pase a la próxima parte. Si aún no se convenció, continúe leyendo. Programar puede volverse un nuevo *hobby* y hasta una profesión. Si estudia computación, necesita saber programar. Eso no significa que será un programador toda la vida, o que la programación limitará su crecimiento dentro del área de la informática. Una excusa que ya oí muchas veces es “*yo sé programar, pero no me gusta*”. Varios alumnos de computación terminan sus cursos sin saber programar; o sea, sin saber programar realmente. Programar es como andar en bicicleta, no se olvida, pero sólo se aprende haciéndolo. Al cambiar de un lenguaje de programación a otro, si usted realmente aprendió a programar tendrá poca dificultad para aprender el nuevo lenguaje. A diferencia de saber programar, la sintaxis de un lenguaje de programación se olvida muy fácilmente. No piense que saber programar es memorizar todos esos comandos, parámetros y nombres extraños. Programar es saber utilizar un lenguaje de programación para resolver problemas, o sea, saber expresar una solución por medio de un lenguaje de programación.

## 1.2 ¿Cuál es su nivel de paciencia?

Sea paciente.

Otro error de quien estudia programación es querer hacer cosas difíciles desde el comienzo.

¿Cuál será su primer programa? ¿Un editor de textos? ¿Una planilla electrónica? ¿Una calculadora?

¡No! Será algo mucho más simple... Cómo sumar dos números.

Es eso mismo: ¡sumar dos números!

Con el tiempo, la complejidad y el tamaño de los programas aumentarán.

Sea paciente.

Programar exige mucha paciencia y, principalmente, atención a los detalles. Una simple coma en lugar de un punto, u olvidar unas comillas, puede arruinar su programa. Al comienzo es común perder la calma o aún desesperarse hasta aprender a leer lo que realmente escribimos en nuestros programas. En esa etapa, la paciencia nunca es demasiada. Lea nuevamente el mensaje de error o deténgase para entender lo que no está funcionando correctamente. Nunca piense que la computadora está contra usted, ni le eche la culpa al día o al destino.

Sea paciente.

## 1.3 ¿Cuánto tiempo pretende estudiar?

Se puede aprender a programar en pocas horas. Si usted es el tipo de persona que programa el microondas de la tía; que para abrir los frascos de remedios lee las instrucciones de la tapa o que jugó con el Lego, entonces “programar” es su segundo nombre.

Todos nosotros ya programamos algo en la vida, aunque sea ir al cine el sábado. La cuestión es: ¿cuánto tiempo le va a dedicar a aprender a programar computadoras?

Como todo en la vida, nada de exageraciones. En realidad, tanto el tiempo como la forma de estudiar varían mucho de persona a persona. Algunas rinden más estudiando en grupo. A otras les gusta asistir a clase.

Lo importante es organizar el estudio de la programación de acuerdo a su estilo preferido. No trate de aprender todo o de entender todo rápidamente. Si eso sucede, felicitaciones, hay mucho por delante. En caso contrario, relájese. Si no entiende en el segundo intento, deje y vuelva a intentarlo mañana.

Cuando encuentre un problema, tenga calma. Vea qué escribió. Verifique si entiende lo que está escrito. Un error común es querer programar sin saber escribir las instrucciones. Es como querer escribir sin saber hablar.

Inicie el estudio con sesiones de una o dos horas por día como máximo. Después ajuste ese tiempo a su ritmo.

## 1.4 ¿Para qué programar?

Si Ud. no necesita programar para su trabajo o estudio, veamos algunas otras razones:

### 1.4.1 Escribir páginas web

Hoy, todos están expuestos a la web, la Internet y a sus miles de programas. La web sólo funciona porque permite la publicación de páginas y más páginas de textos e imágenes usando tan solo un editor de textos. La página más compleja que visitó es un conjunto de líneas de texto reunidas para instruir a un programa, el navegador (*browser*), sobre cómo presentar su contenido.

### 1.4.2 Poner en hora su reloj

¿Ud. conoce algunas personas que nunca aprendieron a poner en hora sus relojes? Yo recuerdo varias...

Seguir instrucciones es muy importante para tareas tan simples como esas. La secuencia de pasos para ajustar las horas, minutos y hasta la fecha de su reloj puede ser encarada como un programa. Normalmente se aprieta el botón de ajuste hasta que un número empieza a parpadear. Después puede usar un botón para cambiar la hora, o ir directamente al ajuste de los minutos. Eso se repite hasta que ha ajustado todos los valores tales como segundos, día, mes y, a veces, el año.

### 1.4.3 Aprender a usar mapas

¿Ya se perdió en una ciudad extraña? ¿Ya hizo una lista de pasos para llegar a algún lugar? Entonces Ud. ya programó. Sólo por buscar un mapa Ud. ya merecería un premio. Al trazar un camino desde donde está hasta donde desea llegar, relaciona una lista de calles o referencias de ese camino. Normalmente es algo así como “pasar tres calles a la izquierda”, “doblar a la derecha”, “doblar a la izquierda”... O algo como “seguir derecho hasta encontrar una señal de tránsito o un río”. O sea, programar. Seguir su programa es la mejor forma de saber si lo que escribió es correcto o si ahora está realmente perdido.

### 1.4.4 Mostrarle a sus amigos que sabe programar

Ésta puede ser la razón más complicada. Vamos a verla como un subproducto del aprendizaje y no como su objetivo final. Si esa es su razón para aprender a programar, es mejor continuar leyendo y conseguir otra.

Programar es un esfuerzo para realizar algo. Es una tarea que exige dedicación y que trae mucha satisfacción personal. Sus programas pueden ser buenos, pero ahí ya serán sus programas y no solamente su persona.

### 1.4.5 Parecer extraño

Entender qué significan miles de líneas de programa puede hacer que usted gane fama de “raro” entre los legos. Si ese es su objetivo, sepa que hay maneras más fáciles de lograrlo, tales como dejar de bañarse, dejarse crecer las uñas, tener el pelo naranja o violeta, parecer un roquero sin haber tocado nunca en una banda, etc.

Aunque buena parte de los programadores que conozco no sea exactamente lo que yo considero 100% normal, nadie lo es.

Saber programar no significa que Ud. sea loco o muy inteligente. Saber programar tampoco significa que Ud. no sea loco o que no sea muy inteligente. Imagine que aprender a programar es como cualquier otra cosa que ya aprendió.

De todos modos, puede llegar un día en que empiece a tener pensamientos extraños, aún sin ser programador, pero no se preocupe porque cuando llegue ese día usted seguramente se dará cuenta...

### 1.4.6 Entender mejor cómo funciona su computadora

Programando puede empezar a entender porqué aquella operación falló o porqué el programa simplemente se cerró de improviso.

Programar también puede ayudarlo a utilizar mejor su planilla o editor de textos. El tipo de razonamiento que se aprende programando le servirá no sólo para hacer programas, sino también para usarlos.

### 1.4.7 Cocinar

Una vez necesité preparar una comida, pero las instrucciones estaban escritas en alemán. No sé nada de alemán. Tomé el primer diccionario que encontré y empecé a traducir las palabras principales. Con las palabras traducidas, traté de entender lo que debería hacer.

Aquella noche la cena fue solo una sopa instantánea. Una receta puede ser vista como un programa. Y como todo programa, sólo es posible seguirla si usted entiende lo que está escrito.

La simple secuencia de instrucciones no ayuda a una persona que no está en condiciones de entender sus efectos.

Para algunas personas programar es más fácil que aprender alemán (o cualquier otro idioma extranjero). Y como cualquier otra lengua, no se aprende sólo con un diccionario.

Los idiomas humanos son ricos en contextos, y cada palabra suele tener múltiples significados. La buena noticia: los lenguajes de programación están hechos para que las máquinas puedan entender lo que allí está representado. Eso significa

que entender un programa es muy fácil, casi como consultar un diccionario. Otra buena noticia es que la mayoría de los lenguajes contienen conjuntos pequeños de “palabras”.

### 1.4.8 Salvar el mundo

Una buena razón para aprender a programar es salvar el mundo. ¡Eso mismo!

Todos los días, miles de kilos de alimento son desperdiciados o no llegan adonde deberían llegar por falta de organización. Programando usted puede ayudar a crear sistemas y aun programas que ayuden a otros a organizarse.

Otra buena acción es ayudar en un proyecto de software libre. Eso permitirá que muchas personas que no pueden pagar por programas para computadoras se beneficien de ellos sin cometer ningún delito.

### 1.4.9 Software libre

Por otra parte, ¿usted tiene licencia de uso para todos sus programas?

Si la respuesta es no, sepa que los programadores aprenden Linux y otros sistemas operativos mucho más rápidamente. También logran sacar mayor provecho de esos sistemas porque consiguen programarlos.

Si algo no existe, créelo. Si es malo, mejórelo.

Poner la programación en un mundo aparte puede ser la primera idea errónea que muchos tienen. La posibilidad de crear mundos dentro de las computadoras y de los programas, puede ser la segunda.

## 1.5 ¿Por qué Python?

El lenguaje de programación Python es muy interesante como primer lenguaje de programación debido a su simpleza y claridad. Aunque simple, es también un lenguaje poderoso que puede ser usado para administrar sistemas y desarrollar grandes proyectos. Es un lenguaje claro y objetivo, pues va directamente al punto, sin rodeos.



Python es software libre, o sea, puede ser utilizado gratuitamente gracias al trabajo de la Python Foundation<sup>1</sup> y de innumerables colaboradores. Puede utilizar Python prácticamente en cualquier arquitectura de computadoras o sistema operativo, como Linux<sup>2</sup>, FreeBSD<sup>3</sup>, Microsoft Windows o Mac OS X<sup>4</sup>.

Python viene creciendo en varias áreas de la computación, como inteligencia artificial; bancos de datos; biotecnología; animación 3D; aplicaciones para móviles (celulares), juegos y aun como plataforma web. Eso explica porqué Python es famoso por tener “*batteries included*”, o sea, baterías incluidas, expresión que hace referencia a un producto completo que puede ser usado inmediatamente (¿Quién no recibió alguna vez un regalo de Navidad que vino sin pilas?). Hoy es difícil encontrar una biblioteca que no tenga *bindings* (enlaces) en Python. Ese hecho vuelve el aprendizaje del lenguaje mucho más interesante ya que aprender a programar en Python permite continuar utilizando los conocimientos adquiridos también para resolver problemas reales.

Una gran ventaja de Python es la legibilidad de los programas escritos en ese lenguaje. Otros lenguajes de programación utilizan innumerables marcaciones, como punto (.) o punto y coma (;), en el final de cada línea, además de los marcadores de comienzo y fin de bloque como llaves { } o palabras especiales (begin/end). Esos marcadores vuelven a los programas un tanto más difíciles de leer y por suerte no son usados en Python. Veremos más sobre bloques y marcaciones en los capítulos siguientes.

Otro buen motivo para aprender Python es poder obtener resultados en poco tiempo. Como Python es un lenguaje completo, contando con bibliotecas para acceder a bancos de datos, procesar archivos XML, construir interfaces gráficas y aun juegos; podemos utilizar muchas funciones ya existentes escribiendo pocas líneas de código. Eso aumenta la productividad del programador, pues al utilizar bibliotecas usamos programas desarrollados y probados por otras personas, lo cual reduce el número de errores y permite concentrarse realmente en el problema que se quiere resolver.

---

1 <http://www.python.org>

2 <http://www.kernel.org> o <http://www.ubuntu.com> para obtener el paquete completo

3 <http://www.freebsd.org>

4 <http://www.apple.com/macosx>

Veamos un pequeño programa escrito en Python en la lista 1.1.

► **Lista 1.1 – Programa Hola Mundo**

```
print("¡Hola!")
```

La lista del programa 1.1 tiene solo una línea de código. La palabra `print` es una función utilizada para enviar datos a la pantalla de la computadora. Al escribir `print("Hola")`, le ordenamos a la computadora que exhiba el texto “¡Hola!” en la pantalla. Vea lo que se mostraría en la pantalla al ejecutar este programa en la computadora:

```
¡Hola!
```

Observe que las comillas (“”) no aparecen en la pantalla. Ese es uno de los detalles de la programación: necesitamos marcar o limitar el comienzo y el fin de nuestros mensajes con un símbolo, en este caso, comillas. Como podemos exhibir prácticamente cualquier texto en la pantalla, las primeras comillas indican el comienzo del mensaje y las siguientes, el fin. Al programar no podemos olvidar las limitaciones de la computadora. Una computadora no interpreta textos como los seres humanos. La máquina no logra diferenciar qué es un programa o un mensaje. Si no utilizamos las comillas, la computadora interpretará nuestro mensaje como un comando del lenguaje Python, generando un error.

El intérprete de Python es una gran herramienta para el aprendizaje del lenguaje. El intérprete es el programa que permite digitar y probar comandos escritos en Python y verificar los resultados instantáneamente. Veremos cómo utilizar el intérprete en la sección 2.2.

El lenguaje Python fue elegido para este libro porque simplifica el trabajo de aprendizaje y provee gran poder de programación. Como es un software libre, disponible prácticamente para cualquier tipo de computadora, su utilización no implica la adquisición de licencias de uso, muchas veces a un costo prohibitivo.

¡Bienvenido al mundo de la programación!