

# Python No Muerde

## Capítulo: Python no muerde, yo sí



Este libro está disponible bajo una licencia CC-by-nc-sa-2.5.

**Es decir que usted es libre de:**



Copiar, distribuir, exhibir, y ejecutar la obra



Hacer obras derivadas

**Bajo las siguientes condiciones:**



Atribución — Usted debe atribuir la obra en la forma especificada por el autor o el licenciante.



No Comercial — Usted no puede usar esta obra con fines comerciales.



Compartir Obras Derivadas Igual — Si usted altera, transforma, o crea sobre esta obra, sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.

El texto completo de la licencia está en el sitio de [creative commons](https://creativecommons.org/licenses/by-nc-sa/2.5/).

**Autor:** Roberto Alsina <[ralcina@netmanagers.com.ar](mailto:ralcina@netmanagers.com.ar)>  
**Versión:** 8e80f80bdea9

# Introducción

## Requisitos

Éste es un libro sobre Python <sup>1</sup>. Es un libro que trata de explicar una manera posible de usarlo, una manera de tomar una idea de tu cabeza y convertirla en un programa, que puedas usar y compartir.

- 1 | ¿Por qué Python? Porque es mi lenguaje favorito. ¿De qué otro lenguaje podría escribir?

¿Qué necesitas saber para poder leer este libro?

El libro no va a explicar la sintaxis de python, sino que va a asumir que la conocés. De todas formas, la primera vez que aparezca algo nuevo, va a indicar dónde se puede aprender más sobre ello. Por ejemplo:

```
# Creamos una lista con los cuadrados de los números pares  
cuadrados = [ x**2 for x in numeros if x%2 == 0 ]
```

### Referencia

Eso es una [comprensión de lista](#)

En general esas referencias van a llevarte al [Tutorial de Python](#) en castellano. Ese libro contiene toda la información acerca del lenguaje que se necesita para poder seguir éste.

Cuando una aplicación requiera una interfaz gráfica, vamos a utilizar PyQt <sup>2</sup>. No vamos a asumir ningún conocimiento previo de PyQt pero tampoco se va a explicar en detalle, excepto cuando involucre un concepto nuevo.

Por ejemplo, no voy a explicar el significado de `setEnabled` <sup>3</sup> pero sí el concepto de signals y slots cuando haga falta.

- 2 | PyQt es software libre, es multiplataforma, y es muy potente y fácil de usar. Eso no quiere decir que las alternativas no tengan las mismas características, pero quiero enfocarme en programar, no en discutir, y **yo** prefiero PyQt. Si preferís una alternativa, este libro es libre: podés hacer una versión propia!
- 3 | PyQt tiene una excelente [documentación de referencia](#) para esas cosas.

## Convenciones

Las variables, funciones y palabras reservadas de python se mostrarán en el texto con letra monoespaciada. Por ejemplo, `for` es una palabra reservada.

Los fragmentos de código fuente se va a mostrar así:

```
# Creamos una lista con los cuadrados de los números impares  
cuadrados = [ x**2 for x in numeros if x%2 > 0 ]
```

Los listados extensos o programas completos se incluirán sin cajas, mostrarán números de líneas e indicarán el nombre del mismo:

cuadrados.py

```
1 # Creamos una lista con los cuadrados de los números impares  
2 cuadrados = [ x**2 for x in numeros if x%2 > 0 ]
```

En ese ejemplo, debería haber, en los ejemplos que acompañan al libro, un archivo `codigo/X/cuadrados.py` donde `X` es el número del capítulo en el que el listado aparece.

## Lenguaje

Las discusiones acerca de como escribir un libro técnico en castellano son eternas. Que en España se traduce todo todo todo. Que en Argentina no. Que decir “cadena de caracteres” en lugar de `string` es malo para la ecología.

Por suerte en este libro hay un único criterio superador que ojalá otros libros adopten: Está escrito como escribo yo. Ni un poquito distinto. No creo que siquiera califique como castellano, como mucho está escrito en argentino. Si a los lectores de la ex madre patria les molesta el estilo... tradúzcanlo.

## Mapa

Dentro de lo posible, voy a intentar que cada capítulo sea autocontenido, explicando un tema sin depender demasiado de los otros, y terminando con un ejemplo concreto y funcional.

Éstos son los capítulos del libro, con breves descripciones.

1. Introducción

2. Pensar en python

Programar en python, a veces, no es como programar en otros lenguajes. Acá vas a ver algunos ejemplos. Si te gustan... python es para vos. Si no te gustan... bueno, el libro es barato... capaz que Java es lo tuyo..

3. La vida es corta

Por eso, hay muchas cosas que no vale la pena hacer. Claro, yo estoy escribiendo un editor de textos así que este capítulo es pura hipocresía...

4. Las capas de una aplicación

Batman, los alfajores santafesinos, el ozono... las mejores cosas tienen capas. Cómo organizar una aplicación en capas.

5. Documentación y testing

Documentar es testear. Testear es documentar.

6. La GUI es la parte fácil

Lo difícil es saber que querés. Lamentablemente este capítulo te muestra lo fácil. Una introducción rápida a PyQt.

7. Diseño de interfaz gráfica

Visto desde la mirada del programador. Cómo hacer para no meterse en un callejón sin salida. Cómo hacerle caso a un diseñador.

8. Un programa útil

Integremos las cosas que vimos antes y usémoslas para algo.

9. Instalación, deployment y otras yerbas

Hacer que tu programa funcione en la computadora de otra gente

10. Cómo crear un proyecto de software libre

## Mapa

¿Cómo se hace? ¿Qué se necesita? ¿Me conviene? Las respuestas son “depende”, “ganas” y “a veces”. O “así”, “una idea” y “sí”. O sea, no sé. Pero veamos.

### 11. Rebelión contra el Zen

Cuándo es mejor implícito que explícito? ¿Cuándo es algo lo suficientemente especial para ser, realmente, especial?

### 12. Herramientas

Programar tiene más en común con la carpintería que con la arquitectura.

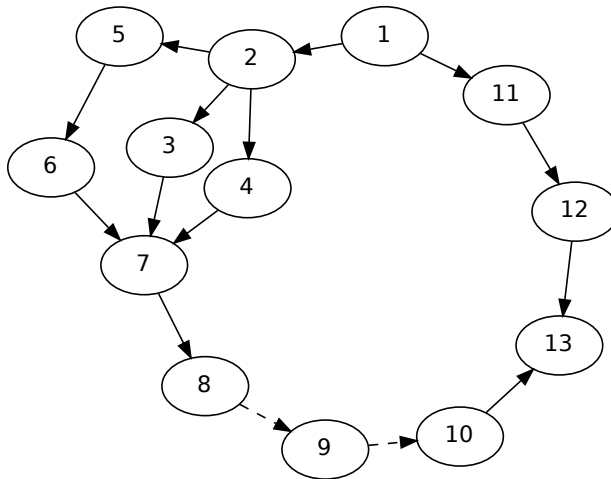
### 13. Conclusiones, caminos y rutas de escape

¿Y ahora qué?

Este es un diagrama de dependencias. Cada capítulo tiene flechas que lo conectan desde los capítulos que necesitas haber leído anteriormente.

Con suerte será un [grafo acíclico](#).

La línea de puntos significa ‘no es realmente necesario, pero...’



*Este libro se lee siguiendo las flechas.*

Acerca del Autor

Acerca del Autor

Habr  que pedirle a alguien que ponga algo no demasiado insultante.