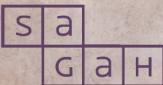


# Programação Back End I

Maurício de Oliveira Saraiva



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS

# Python e ambientes de desenvolvimento

## Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever a evolução histórica do Python.
- Instalar o Python e ferramentas para desenvolvimento.
- Diferenciar as ferramentas de implementação do Python.

## Introdução

A tecnologia tem avançado muito nos últimos anos e, com ela, as linguagens de programação têm se tornado mais populares. O avanço dessas linguagens permitiu, inclusive, que usuários leigos pudessem desenvolver aplicações e sistemas.

Nesse contexto, surgiu a linguagem de programação Python, com o propósito de oferecer aos desenvolvedores uma linguagem simples e fácil de manusear que fosse, ao mesmo tempo, elegante e poderosa. A linguagem evoluiu muito e, atualmente, ela conta com muitas ferramentas, entre editores de texto, IDEs e ferramentas de refatoração, que oferecem inúmeros recursos para tornar o trabalho dos desenvolvedores cada vez menos complexo.

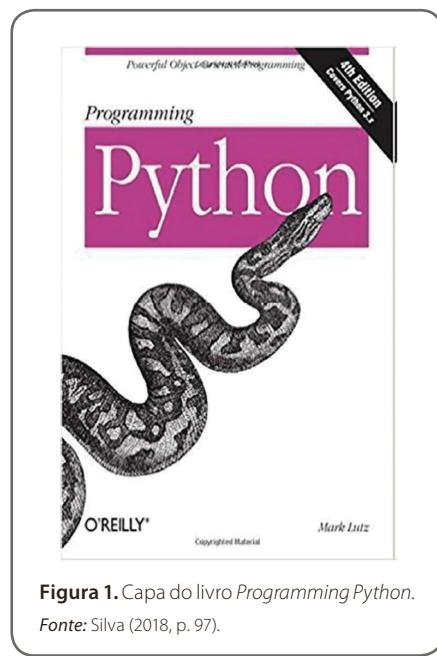
Neste capítulo, você conhecerá a evolução histórica do Python. Além disso, você aprenderá a fazer a instalação do Python e de ferramentas de desenvolvimento, assim como saberá diferenciar as ferramentas para a implementação do sistema.

## A evolução histórica do Python

A linguagem de programação Python foi criada pelo matemático Guido van Rossum, entre o final da década de 1980 e o início da década de 1990. Ela foi elaborada como uma linguagem de *scripts* para o sistema operacional Amoeba, sendo uma evolução da linguagem de programação ABC, que era voltada para físicos, engenheiros e outros (SILVA, 2018).

Van Rossum deu este nome para a nova linguagem porque gostava muito dos filmes e programas humorísticos da série *Monty Python*. Sua ideia foi criar uma linguagem que tornasse o ato de programar mais simples e leve, como o programa televisivo. Este também foi um meio que Guido encontrou para se encaixar em um padrão estabelecido na época: dar nomes de pessoas famosas para as linguagens de programação, como o exemplo das linguagens Ada e Pascal, mas, no seu caso, homenageando um programa de televisão famoso.

Por muitos anos, van Rossum tentou evitar que as pessoas vinculassem o nome de sua linguagem de programação à cobra píton, que é uma espécie de réptil. Ele desistiu quando a editora O'Reilly sugeriu que ele colocasse a foto de uma cobra píton na capa do primeiro livro de programação em Python, de 1996, que se chamaria *Programming Python*. Veja a imagem na Figura 1.



**Figura 1.** Capa do livro *Programming Python*.

Fonte: Silva (2018, p. 97).

O Python foi criado para ser uma linguagem de programação utilizada para o ensino da programação, trabalhada por leigos no assunto, com o objetivo de ser produtiva e legível para desenvolvedores, ou seja, o propósito de sua criação era o desenvolvimento de *software* com um código de fácil leitura, que proporcionasse uma manutenção igualmente fácil e rápida (PYSCIENCE-BRASIL, 2019).

A linguagem Python é de alto nível, gratuita e orientada a objetos. Possui sintaxe simples, e, ao mesmo tempo, muito limpa, legível e elegante, configurando-se, assim, em uma linguagem ideal para desenvolver aplicações de forma rápida (PYTHON SOFTWARE FOUNDATION, 2019a).

Algumas características do Python expressam melhor seus objetivos de ser menos complexa e mais compreensível (SILVA, 2018), conforme as listadas a seguir.

- A utilização de caracteres especiais é bem reduzida, o que faz com que o código escrito em Python fique muito parecido com um código escrito em português estruturado, ou um pseudocódigo executável.
- A estruturação dos blocos de código é completamente feita por meio de indentação.
- As palavras-chave que servem para a compilação também são desnecessárias.
- Existe uma espécie de coletor de lixo, que gerencia, de maneira automática, a utilização da memória durante a programação.
- Suporte a técnicas e estratégias de programação mais complexas, como a orientação a objetos.
- Por ser um *software* livre, o Python é uma linguagem sobre a qual os programadores podem aprofundar seu conhecimento e, ainda, modificar o próprio código-fonte da linguagem.
- Não possui um propósito específico, podendo ser utilizada para desenvolver os mais diversos tipos e tamanhos de aplicações.
- É multiplataforma, funcionando em vários sistemas operacionais e dispositivos.
- O pacote de instalação contém uma biblioteca com diversas classes, métodos e funções.

Além dessas características, é importante citar que existe uma filosofia por trás da linguagem de programação Python, chamada *Zen of Python*, que possui algumas frases que auxiliam o programador ao longo do seu trabalho (SILVA, 2018):

- Bonito é melhor que feio.
- Explícito é melhor que implícito.
- Simples é melhor que complexo.
- Complexo é melhor que complicado.
- Linear é melhor que aninhado.
- Esparsos é melhor que denso.
- Legibilidade conta.
- Casos especiais não são especiais o bastante para quebrar as regras.
- Erros nunca devem passar silenciosamente, a menos que sejam explicitamente silenciados.
- Diante da ambiguidade, recuse a tentação de adivinhar.
- Deveria haver um e, preferencialmente, apenas um modo óbvio para fazer algo.
- Agora é melhor que nunca.
- Se a implementação é difícil de explicar, é uma má ideia.
- Se a implementação é fácil de explicar, pode ser uma boa ideia.
- *Namespaces* são uma grande ideia.

A primeira versão do Python, mínima, mas operacional, chamava-se v0.9.0 e foi disponibilizada no início dos anos 1990. Esta versão já era uma linguagem mais utilizada que sua antecessora, a ABC. Continha 21 partes que formavam um pacote, e nesta versão o Python contava com classes, herança, funções, módulos, formas de tratar exceções e, ainda, tipos de dados nativos da linguagem, como o *list*, *dict*, *str*, entre outros (MAGNUN, 2014).

Após a v0.9.0, foram lançadas outras 14 versões do Python, entre 1991 e 1995, todas no *Centrum wiskunde & informatica* (CWI), em português, Centro de matemática e computação. A numeração das versões passou da v0.9.0, de 1991, para a versão 1.2, em abril de 1995. Apesar de a linguagem Python ter sido idealizada por Guido van Rossum, dentro do CWI, este órgão nunca financiou seu projeto de maneira oficial.

Em meados de 1994, a comunidade de desenvolvedores Python entendeu que o fato de Guido ser o principal desenvolvedor e tomar praticamente todas as decisões a respeito da linguagem poderia diminuir a possibilidade de alguém

ou alguma organização investirem nesta, o que poderia extinguí-la. Neste contexto, surgiu a necessidade de que uma organização se responsabilizasse pela linguagem e sua evolução.

Ainda em 1994, foi realizado o primeiro encontro sobre Python, contando com a presença de aproximadamente 20 profissionais desenvolvedores da linguagem. Nesse encontro, Guido recebeu uma proposta para trabalhar na Corporation for National Research Initiatives (CNRI) e o Python passou a receber investimento da Defense Advanced Research Projects Agency (DARPA).

Foi criada, então, a primeira organização dedicada somente ao Python, que se chamava *Python software activity*, que auxiliou no crescimento e na divulgação da comunidade Python, principalmente por meio do site oficial da linguagem e dos *special interest groups* (SIGS), grupos de interesse específico na linguagem. Entre 1995 e 2000, foram criadas seis versões do Python, de 1.3 a 1.6.



### Link

O site oficial da linguagem de programação Python está disponível em inglês, mas pode ser traduzido para o português. Para acessá-lo, clique no *link* a seguir.

[www.python.org](http://www.python.org)

No início dos anos 2000, os principais idealizadores do Python receberam um convite para fazer parte da BeOpen.com, uma empresa de desenvolvedores de *software open source*. Naquela época, o Python era conhecido como uma linguagem com licença em camadas, uma vez que misturava sua base do CWI, mas tinha uma parte trabalhada no CNRI, além de outra parte já trabalhada na BeOpen.com.

Em 2000, foi criada a versão 2.0 do Python pelo PythonLabs, dentro da BeOpen.com, que era basicamente a versão 1.6, alterando apenas o responsável para a BeOpen.com. Logo depois, a BeOpen.com entrou em falência e os integrantes do PythonLabs foram contratados pela empresa Digital Creations. A Digital Creations, por sua vez, teve seu nome alterado para Zope Corporation, para prestar homenagem ao Web CMS Zope, seu produto mais conhecido para gerenciamento de conteúdo.

Como o objetivo dos desenvolvedores e da comunidade Python era que a linguagem não sofresse influência direta apenas de seus idealizadores, em 2001 foi criada a Python Software Foundation (PSF), cuja logomarca mostra duas cobras entrelaçadas. Por esse motivo e pela rapidez da mudança, nenhuma versão do Python foi atribuída à Zope Corporation, apesar de a Zope patrocinar a PSF até os dias de hoje.

A PSF era uma organização sem fins lucrativos, formada por desenvolvedores Python, que tinha como propósito manter a propriedade intelectual da linguagem, a fim de promover sua divulgação e seu avanço, além de auxiliar no surgimento e no crescimento das comunidades de programadores Python.

Dessa forma, todas as versões posteriores do Python foram feitas pela *PSF license agreement*, licença que atribui à PSF todos os direitos sobre a marca Python (MAGNUN, 2014). Quaisquer alterações avaliadas e aprovadas pela maioria dos membros da PSF são implementadas em uma nova versão da linguagem. Atualmente, a PSF é patrocinada por empresas como Microsoft, Google e Globo.com.

A versão 3.0 do Python, conhecida como Python 3000 ou Python3, foi lançada em 2008. Esta versão apresentou uma quebra de compatibilidade com as versões anteriores, que eram conhecidas como a família 2.x de versões. Essa incompatibilidade veio para corrigir diversas falhas existentes e limpar excessos e recursos mais complicados das versões anteriores.

Entre as modificações realizadas para o Python3, pode-se citar a alteração da palavra reservada *print*, que passou a ser uma função e, assim, facilitou a utilização dessa rotina, segundo os desenvolvedores.

Atualmente, o Python é componente embutido de diversos sistemas operacionais, como o Linux, NetBSD, OpenBSD, OS X e AmigaOS 4. Além disso, muitas empresas famosas utilizam o Python em seus projetos, como Google, Youtube, Nasa, Disney, Globo.com, Magazine Luiza, entre outras.

## A instalação do Python e ferramentas para desenvolvimento

### Instalando o Python

A instalação do Python para Windows é simples e sua explicação em detalhes pode ser encontrada no *site* oficial da linguagem. Para iniciar, deve-se baixar o instalador do Python.



## Link

O instalador do Python está disponível no *link* a seguir.

<https://qrgo.page.link/2otU8>

No topo do *site* é apresentada a última versão disponível para instalação. Para este exemplo, a versão para *download* é a 3.7.4, como mostra a Figura 2.

The screenshot shows the Python Software Foundation website's download section for Windows. At the top, there is a navigation bar with links for About, Downloads, Documentation, Community, Success Stories, News, and Events. Below the navigation bar, a large yellow button says "Download Python 3.7.4". To the right of the button is a graphic of two boxes descending from the sky on parachutes. Text on the page includes: "Download the latest version for Windows", "Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other", "Want to help test development versions of Python? Pre-releases, Docker Images", "Looking for Python 2.7? See below for specific releases", "Looking for a specific release?", and "Python releases by version number". A small copyright notice at the bottom right reads "© Python Software Foundation".

**Figura 2.** Tela de *download* do Python.

*Fonte:* Python Software Foundation (2019b, documento *on-line*).

Após baixar o arquivo de instalação e executá-lo, a tela, conforme ilustra a Figura 3, será mostrada. Ela oferece a opção de adicionar o Python na variável de ambiente PATH. Se ela estiver habilitada, isso significa que o Python poderá ser executado pela linha de comandos do Windows. Esta é uma boa opção para não precisar configurar, posteriormente, essa variável de ambiente de maneira manual.

É possível, ainda, escolher a opção de customizar a instalação, na qual será possível escolher quais ferramentas serão instaladas; ou, simplesmente, clicar em *Install Now*, para instalar o Python no endereço sugerido, com todas as ferramentas, incluindo a IDE IDLE.

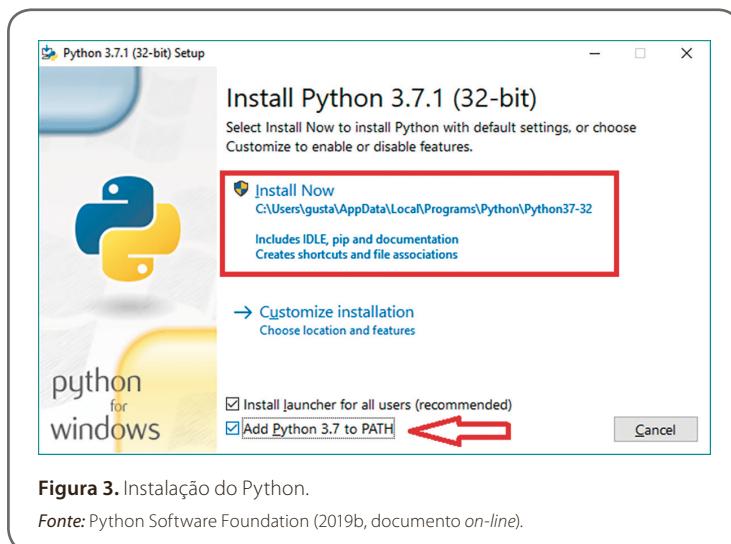


Figura 3. Instalação do Python.

Fonte: Python Software Foundation (2019b, documento on-line).

Clicando em *Install Now*, uma tela com o progresso da instalação será apresentada e é preciso aguardar a finalização da instalação. É importante lembrar que para instalar o Python é necessário possuir permissões de administrador no computador. Quando a instalação for concluída, basta clicar em *Close*, conforme mostra a Figura 4.

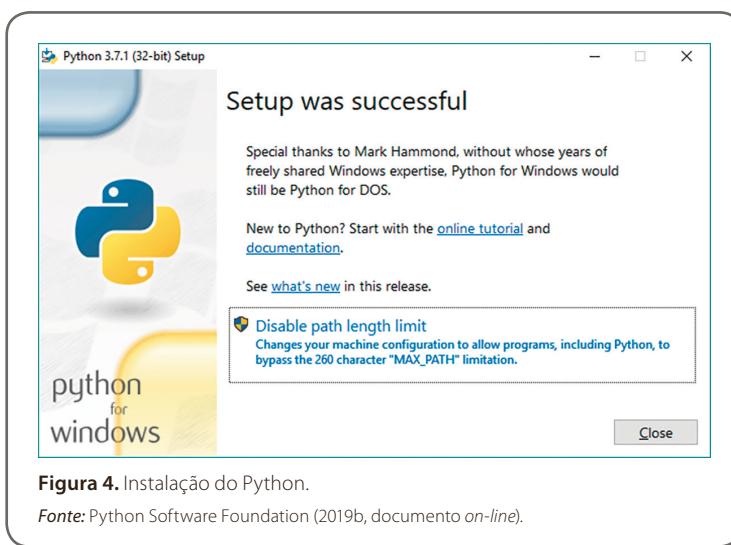
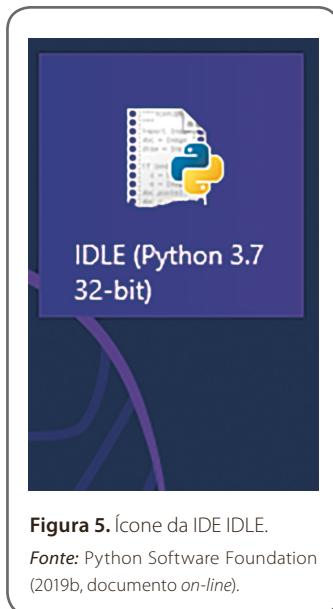


Figura 4. Instalação do Python.

Fonte: Python Software Foundation (2019b, documento on-line).

## Instalando a IDE IDLE

A IDE IDLE é um ambiente integrado de desenvolvimento que já está instalado, por padrão, na instalação do Python, ou seja, está presente nos pacotes de instalação disponíveis no site oficial da linguagem. Depois de instalá-la, basta dar duplo clique no ícone da IDLE para que esta seja executada, conforme ilustrado na Figura 5.



**Figura 5.** Ícone da IDE IDLE.

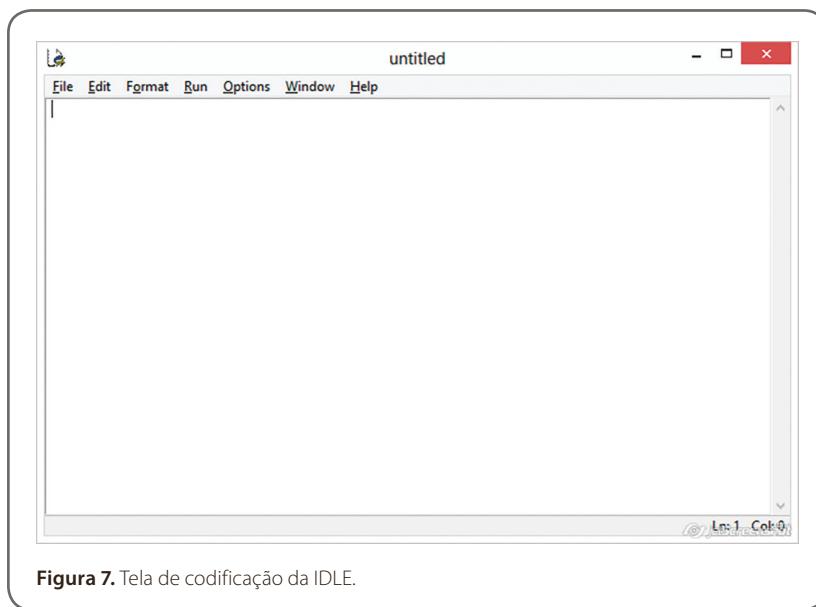
*Fonte:* Python Software Foundation (2019b, documento *on-line*).

Em seguida, a janela ilustrada na Figura 6 aparecerá e será possível realizar alguns testes.



**Figura 6.** Tela principal da IDLE.

Para digitar o código de um programa, basta clicar na barra de ferramentas em *File*, *New File*, e uma tela em branco irá aparecer, conforme mostra a Figura 7. Essa tela permite, além de codificar seus programas, utilizar todos os recursos da IDE e colocar o programa em execução.



**Figura 7.** Tela de codificação da IDLE.

## Instalando a IDE PyCharm

A IDE PyCharm é um dos ambientes de desenvolvimento Python mais conhecidos e fáceis de utilizar, além de ser gratuito. Foi desenvolvido pela empresa JetBrains, que é especialista em desenvolvimento de *software* utilizados para desenvolver outros *software*. AS IDEs desenvolvidas pela JetBrains estão entre as melhores e mais intuitivas do mercado.

A PyCharm apresenta um módulo de acesso a banco de dados, permitindo que o desenvolvedor teste seus códigos SQL, administre o banco de dados, altere, insira e exclua os registros das tabelas, sem a necessidade de utilizar outro programa.

É importante ressaltar que para instalar a IDE PyCharm é preciso ter o Java pré-instalado, além de permissões de administrador no computador. É preciso, também, ter instalado o Python previamente, pois a PyCharm irá realizar a configuração do ambiente e, se não houver o Python, será preciso configurar todos os recursos posteriormente, de forma manual.



### Link

Para fazer *download* da IDE PyCharm, acesse o *link* a seguir.

<https://qrgo.page.link/Euy8u>

Após acessar o *link* para *download* da PyCharm, basta clicar no botão *Download*, do PyCharm Community, como mostra a Figura 8, uma vez que essa é a versão gratuita.



**Figura 8.** Tela de download da PyCharm.

*Fonte:* JetBrains (2019, documento on-line).

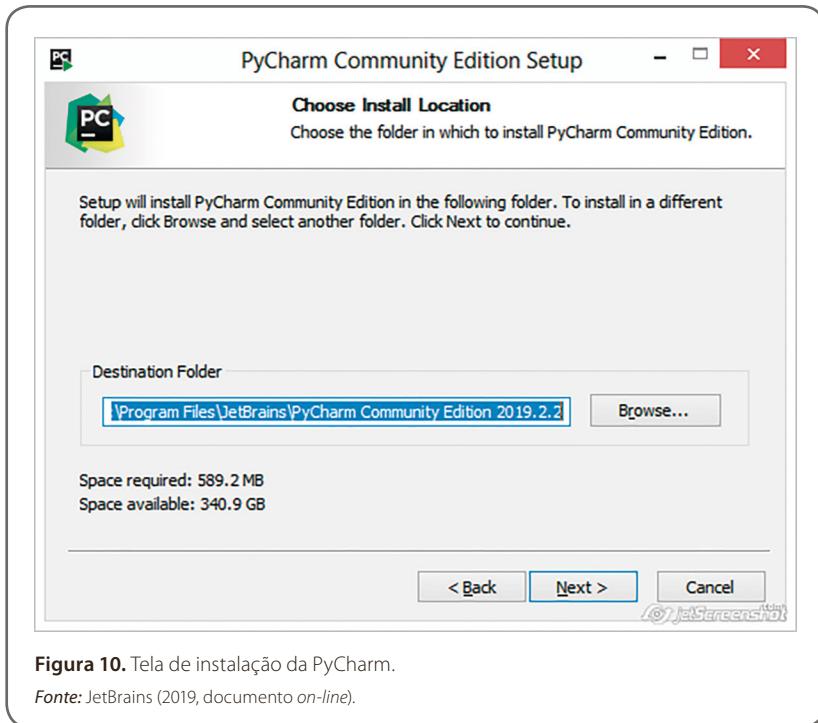
Basta escolher um local para salvar o arquivo de instalação e, após baixar o arquivo, clicar sobre ele duas vezes sobre ele. A tela mostrada na Figura 9 aparecerá e será preciso clicar em *Next*.



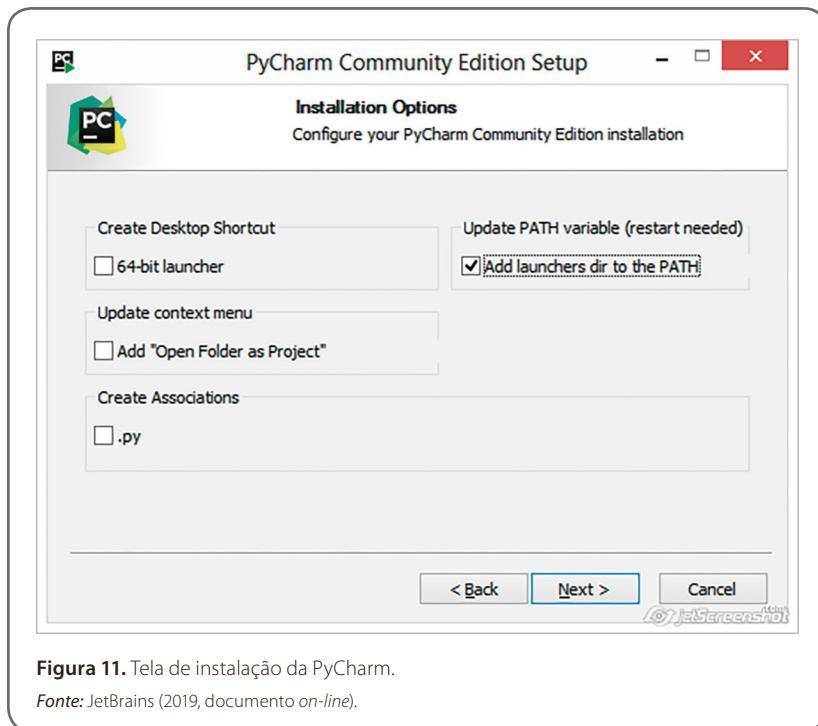
**Figura 9.** Tela de instalação da PyCharm.

*Fonte:* JetBrains (2019, documento on-line).

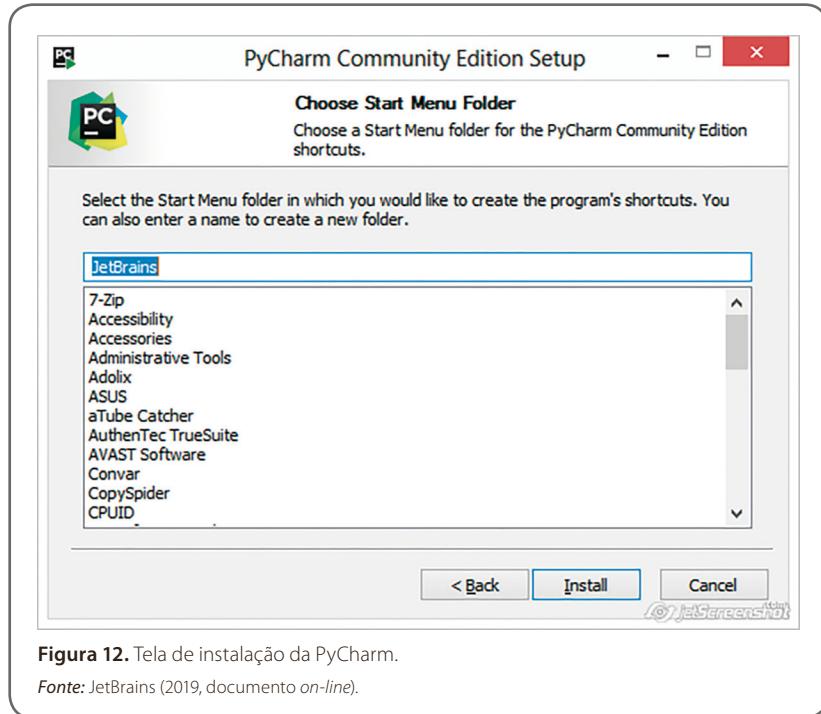
Em seguida, o instalador pedirá confirmação sobre o local do disco rígido em que a IDE será instalada. Esse endereço pode ser modificado. Depois de pronto, basta clicar em *Next*, como mostra a Figura 10.



Na próxima tela será possível criar um atalho para executar a PyCharm e alterar a variável de ambiente PATH. Escolhendo essa opção, será preciso reiniciar o computador. Basta clicar em *Next*, como mostra a Figura 11.



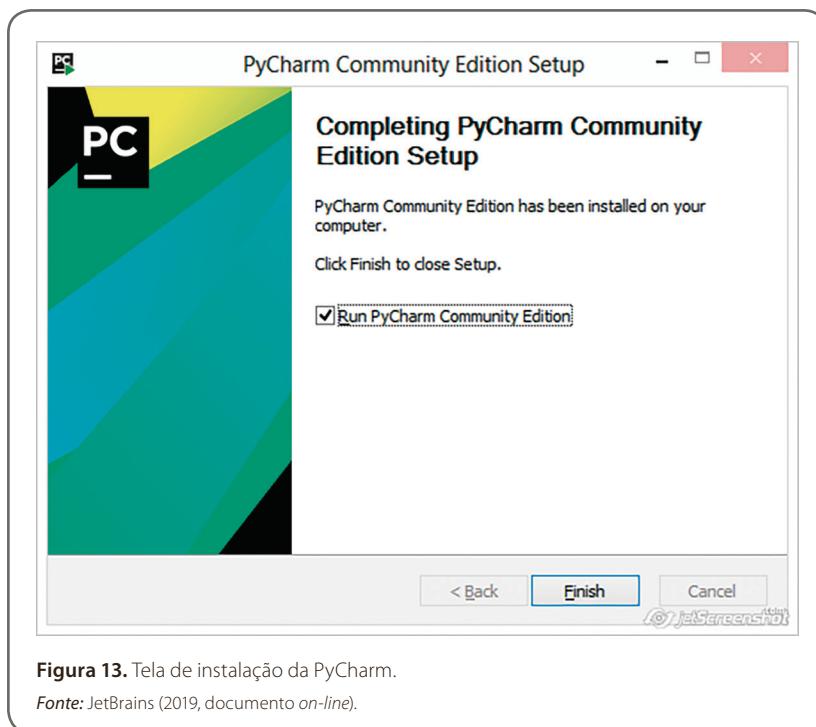
Na próxima tela, é preciso clicar em *Install* e aguardar a finalização da instalação, como mostra a Figura 12.



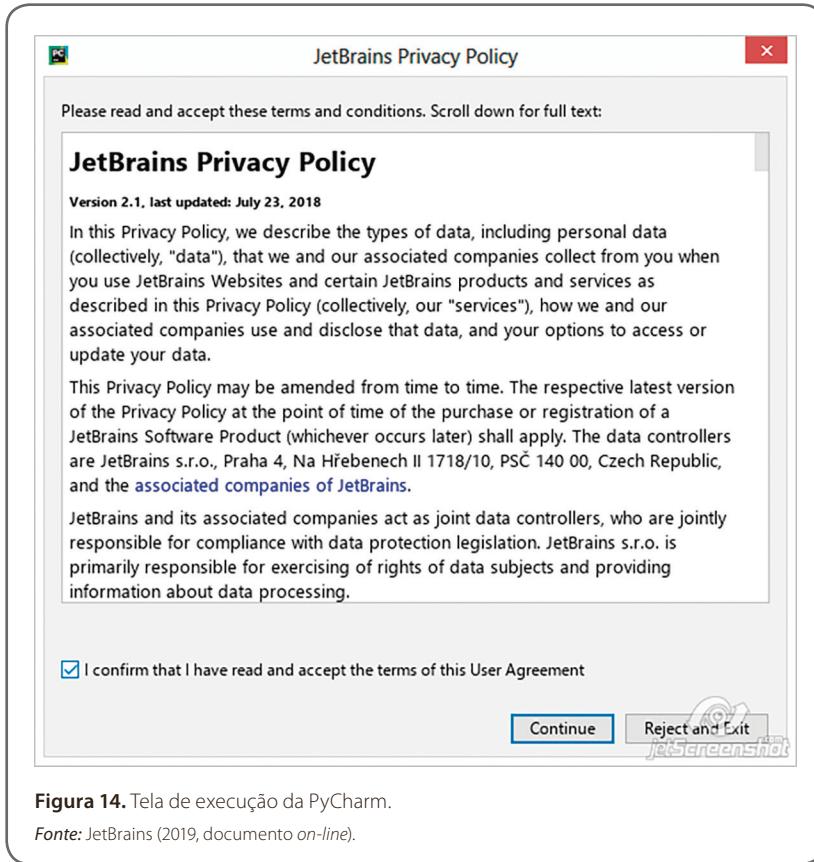
**Figura 12.** Tela de instalação da PyCharm.

*Fonte:* JetBrains (2019, documento on-line).

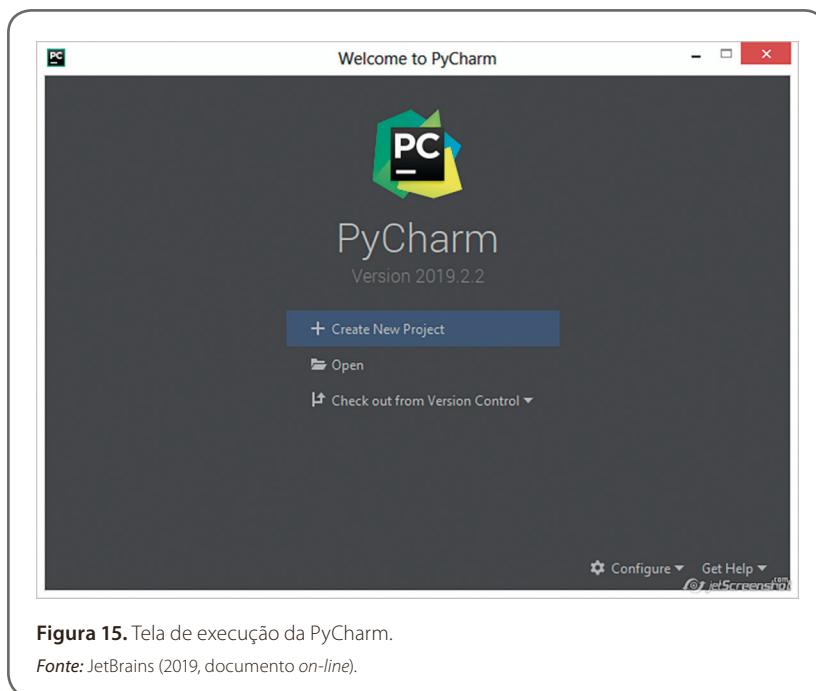
Quando a instalação terminar, a tela mostrada na Figura 13 irá aparecer. Clicando em *Finish*, com a opção *Run PyCharm Community Edition* habilitada, a IDE PyCharm será executada.



Para prosseguir, será preciso concordar com a política da fabricante da PyCharm, a JetBrains, conforme mostra a Figura 14.



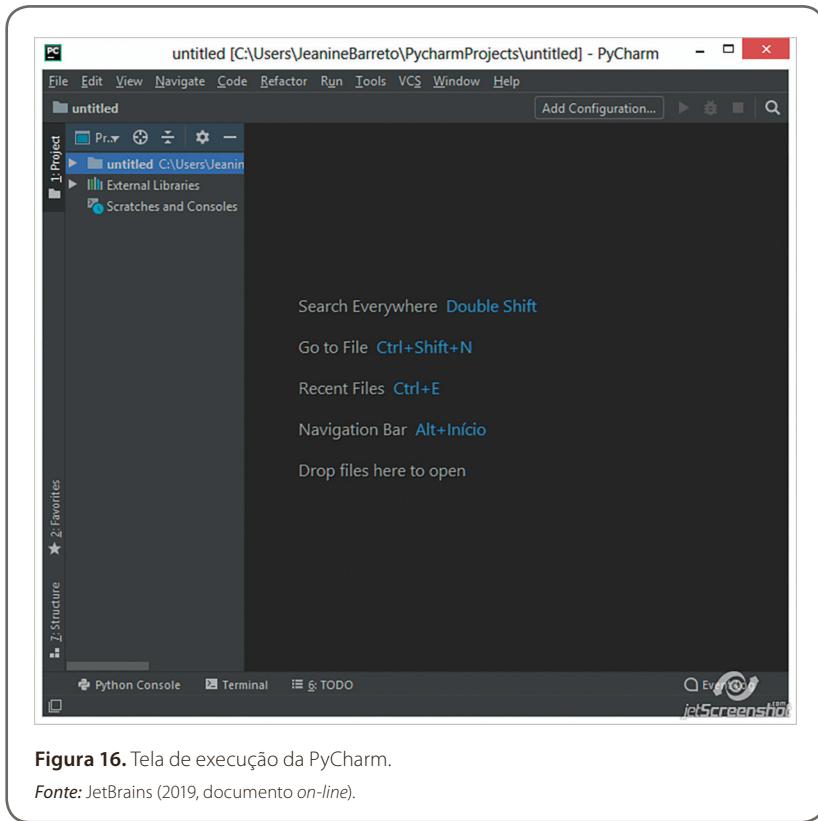
Será exibida, então, a tela de opções da PyCharm, que oferece a possibilidade de iniciar um novo projeto ou abrir um já existente, como mostra a Figura 15.



**Figura 15.** Tela de execução da PyCharm.

*Fonte:* JetBrains (2019, documento *on-line*).

Iniciando um novo projeto, será exibida a tela principal da PyCharm, conforme ilustra a Figura 16. Essa tela fornece acesso a todos os recursos dessa IDE.



**Figura 16.** Tela de execução da PyCharm.

*Fonte:* JetBrains (2019, documento on-line).

## As ferramentas de implementação do Python

O site oficial do Python apresenta suas principais ferramentas de implementação, que podem ser utilizadas para auxiliar os desenvolvedores no uso desta linguagem.



## Link

Os detalhes a respeito das ferramentas de desenvolvimento em Python estão disponíveis no *link* a seguir.

<https://qrgo.page.link/FoJU2>

Estas ferramentas podem ser classificadas em editores de texto, IDEs gratuitas, IDEs pagas e, ainda, ferramentas de refatoração, conforme serão apresentadas a seguir (PYTHON SOFTWARE FOUNDATION, 2019a).

### Editores de texto

**Atom:** excelente para iniciantes, esta ferramenta tem código aberto, é feita pelo Github e apresenta suporte para várias linguagens, sendo o Python uma delas. É possível trabalhar com o Git e o Github pela interface do editor de texto.

**Emacs:** essa ferramenta de edição de texto é poderosa e extensível ao eLisp. Depois de algum tempo de utilização e, tendo o desenvolvedor dominado a Emacs, é possível torná-la uma IDE Python muito completa.

**jext:** é um excelente editor de texto que apresenta suporte para várias linguagens de programação e possui *plugins* para a linguagem Python.

**joe:** essa ferramenta leva os desenvolvedores ao tempo dos antigos editores de texto, pois seus comandos são os mesmos do WordStar, Sidekick e outros editores que foram populares há muitos anos.

**PS Pad:** é uma ferramenta grátis para o sistema operacional Windows. Tem como característica colorir o código em Python e apresenta suporte para vários tipos de codificação. Possui utilitários para HTML/XML, tabela ASCII e conversão entre DOS e UNIX.

**SciTE:** essa ferramenta é um ótimo editor de texto, desenvolvido para auxiliar no desenvolvimento de código de programação. Apresenta suporte para várias linguagens de programação, roda e debuga programas, é simples de usar e sua configuração é facilitada. Está disponível para Windows e X, que é o ambiente gráfico do UNIX.

**sublime:** essa ferramenta consiste em um editor de texto proprietário. Ela é paga, mas apresenta uma versão beta que permite experimentá-la sem custo por um período.

**Vim:** é uma ferramenta com grande potencial de configuração e pode ser transformada em IDE Python.



### Link

No link a seguir, você pode acessar um tutorial que ensina a transformar o Vim em uma IDE Python.

<https://qrgo.page.link/2dCTA>

**Visual Studio Code:** mais conhecida como VSCode, essa ferramenta é gratuita e tem código aberto. Foi desenvolvida pela Microsoft e apresenta suporte para várias outras linguagens de programação, além do Python.

## IDEs gratuitas

**EasyEclipse:** essa ferramenta tem código aberto, é hospedada pela Sourceforge e apresenta pacotes do Eclipse, que são pré-configurados com *plugins* para codificação em Python, Ruby e outras linguagens.

**Eclipse:** o Eclipse é a IDE mais pesada e poderosa que suporta o Python. Esta é uma ferramenta feita em Java, perfeita para o desenvolvimento de código de programação nesta linguagem, mas apresenta *plugins* que permitem a codificação em Python.

**Komodo-Edit:** apresenta um ótimo editor de texto, dotado de muitos recursos importantes, como o autocompletar, dicas, suporte a arquivos de diversas linguagens, coloração de sintaxe, checagem de sintaxe, entre outros.

**IDLE:** o nome dessa ferramenta significa *integrated development and learning environment*, ou ambiente integrado de desenvolvimento e aprendizagem. A IDLE consiste em um ambiente de desenvolvimento completo e integrado para o Python. Ela é lançada e embarcada em todas as versões dessa linguagem, desde a 2.3, e é totalmente escrita em Python. Dentre as características da IDLE, destaca-se que esta conta com um editor de textos de várias janelas, com vantagens como a sintaxe, o autocomplemento, a indentação rápida, entre outras. Além disso, possui um depurador integrado com um passo a passo, pontos de parada de execução persistentes e visibilidade para as chamadas de pilha.

**NetBeans:** assim como o Eclipse, é uma IDE que oferece suporte à linguagem de programação Python por meio de *plugins*.

**NINJA-IDE:** envolve uma IDE multiplataforma de desenvolvimento integrado, podendo ser executada em sistemas Linux, Mac OS e Windows, proporcionando aos desenvolvedores a criação de aplicações que podem ser utilizadas para diversos fins. Os recursos dessa IDE têm como propósito transformar a experiência de codificar *software* em algo agradável e simples.

**Pida:** essa ferramenta tem como propósito fazer a integração com o editor Vim, apresentando recursos como gerenciamento de projetos, gerenciamento de código-fonte, refatoração, depurador, entre outros.

**PyCharm community:** essa IDE foi desenvolvida pela companhia JetBrains, sob licença da Apache. Ela é multiplataforma e fornece análise de código, depurador gráfico, ferramenta de teste de unidade integrado, integração com sistemas de controle de versão, além suportar desenvolvimento para Web com o Django.

**PyPE:** consiste em um editor para Python, desenvolvido em Python, que é simples e funcional, mas apresenta muitos recursos importantes, além de ser multiplataforma.

**PyScripter:** fornece suporte para recursos como debugador, autocompletar, navegação no código e outros.

**PythonCard:** essa é uma IDE utilizada para construir aplicações multi-plataforma no sistema Windows, Mac OS X e Linux, utilizando o Python. Essa ferramenta auxilia os programadores que pretendem desenvolver aplicações gráficas de modo fácil e rápido, com código simples, mas poderoso.

**PythonWin:** essa é a IDE que vem acompanhando as extensões PyWin32, que são as extensões Win32 para Python. Como recursos, ela oferece o autocompletar e o debugador, além de recursos interessantes, voltados à programação com sistema Windows.

**Spyder:** essa ferramenta, também chamada de Pydee, consiste em um ambiente de desenvolvimento completo e interativo para Python, que possui recursos de edição avançada, testes interativos embutidos, recursos de depuração e introspecção.

## IDEs pagas

**Komodo:** essa IDE suporta várias linguagens de programação, incluindo o Python, e é multiplataforma, pois roda em Linux e Windows. Dispõe de uma licença gratuita, que pode ser utilizada pelos desenvolvedores durante seu aprendizado.

**PyCharm:** essa é uma das IDEs mais ricas em recursos, que propicia um desenvolvimento eficiente. Apresenta recursos importantes para o desenvolvimento profissional para Web, suporte para diversos sistemas de controle de versão e integração com Github por meio de *plugins*. Além disso, possui gerador de diagrama de classe e suporte ao interpretador Python remoto.

**Wing:** essa IDE é voltada para o ensino de Python, apresentando menos recursos na versão wingide101 do que nas versões profissional e pessoal. No entanto, contém recursos importantes, como o depurador gráfico.

## Ferramentas de refatoração

A refatoração é o processo que envolve a modificação de um *software*, com a finalidade de aperfeiçoar a estrutura interna do código-fonte sem que isso ocasione modificações no comportamento externo do sistema. Entre essas ferramentas, estão as seguintes.

- **Bcycle Repair Man:** essa ferramenta faz a automatização de operações básicas de refatoração para a linguagem de programação Python, apresentando integração com vários editores, como o Emacs e o PyDev.
- **Rope:** essa ferramenta possui biblioteca de refatoração para Python e pode ser utilizada em vários editores e IDEs. Proporciona ao desenvolvedor a possibilidade de renomear e reestruturar o código, mover classes, funções, módulos e métodos. Além disso, os programadores podem visualizar as modificações, refazê-las ou desfazê-las.



## Referências

JETBRAINS. *Download PyCharm*. [2019]. Disponível em: <https://www.jetbrains.com/pycharm/download/#section=windows>. Acesso em: 10 out. 2019.

MAGNUM. *A história do Python*. 2014. Disponível em: <http://mindbending.org/pt/a-historia-do-python>. Acesso em: 10 out. 2019.

PYSCIENCE-BRASIL. *Python: o que é? Por que usar?* [2019]. Disponível em: <http://pyscience-brasil.wikidot.com/python:oq-e-pq>. Acesso em: 10 out. 2019.

PYTHON SOFTWARE FOUNDATION. *Download the latest version for Windows*. 2019b. Disponível em: <https://www.python.org/downloads/>. Acesso em: 10 out. 2019.

PYTHON SOFTWARE FOUNDATION. [Site Oficial]. 2019a. Disponível em: <https://www.python.org/>. Acesso em: 10 out. 2019.

SILVA, D. M. *Python: história e ascendência*. Revista Programar, Lisboa, ed. 59, p. 96–99, 2018.

## Leituras recomendadas

VELINO. *VIM a IDE para programadores Python*. 2011. Disponível em: <https://avelino.run/vim-a-ide-para-programadores-python/>. Acesso em: 10 out. 2019.

PYTHON BRASIL. *Ferramentas de desenvolvimento*. [2019]. Disponível em: <https://python.org.br/ferramentas/>. Acesso em: 10 out. 2019.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES  
EDUCACIONAIS  
INTEGRADAS