

# CAPÍTULO 1

---

## Ambientes de desenvolvimento

---

### 1.1 Tutorial de uso do PyCharm e do Jupyter

Neste tutorial, segue as instruções básicas sobre o uso do PyCharm e o Project Jupyter Lab. Ambas as ferramentas são poderosas para desenvolvimento de software e análise de dados em Python e oferecem recursos úteis em diferentes cenários.

### 1.2 PyCharm

#### 1.2.1 Instalação

- Baixe e instale o PyCharm Community Edition a partir do site oficial (<https://www.jetbrains.com/pycharm/>).
- Siga as instruções do instalador para concluir a instalação.

#### 1.2.2 Criando um projeto

- Abra o PyCharm e clique em *Create New Project* (Criar novo projeto).
- Escolha um diretório para o projeto.

- Defina o interpretador Python desejado.
- Clique em *Create* (Criar) para criar o projeto.

### 1.2.3 Configurando um ambiente virtual

- Recomenda-se configurar um ambiente virtual para isolar as dependências do projeto.
- No menu, clique em *File* (Arquivo) – > *Settings* (Configurações) – > *Python Interpreter* (Interpretador Python).
- Clique no ícone de engrenagem e selecione *Add* (Adicionar).
- Escolha *Virtualenv Environment* (Ambiente Virtualenv) e siga as instruções para configurar o ambiente.

### 1.2.4 Editor de código

- O PyCharm oferece um editor de código e repleto de recursos.
- Crie um novo arquivo Python clicando com o botão direito do mouse na pasta do projeto e selecionando *New* (Novo) – > *Python File* (Arquivo Python).
- Comece a escrever seu código no editor.

### 1.2.5 Executando código

- De forma simples, para executar um arquivo Python no PyCharm, basta clicar com o botão direito do mouse no arquivo e selecionar *Run* (Executar);
- Se estiver familiarizado, pode executar o arquivo em python pelo terminal com o comando *python nome\_do\_arquivo.py*.

### 1.2.6 Debug

- O PyCharm oferece recursos avançados de depuração.
- Coloque pontos de interrupção no seu código clicando na margem esquerda do editor.

- Clique em *Debug* (Depurar) para executar o código em modo de depuração.

## 1.3 Project Jupyter

### 1.3.1 Instalação

- O Jupyter pode ser instalado utilizando o **pip** com o comando `pip install jupyterlab`.
- O Jupyter Notebook geralmente é instalado junto com a distribuição do Anaconda (<https://www.anaconda.com/>) ou o Miniconda (<https://docs.conda.io/en/latest/miniconda.html>).

### 1.3.2 Editor de código

- Abra o terminal ou prompt de comando e digite o comando `jupyter-lab` ou `jupyter lab`, que abrirá uma interface web no seu navegador padrão.
- No Jupyter Notebook, clique em *New* (Novo) – > *Python 3* para criar um novo notebook Python.
- Uma nova aba será aberta com o notebook vazio.

### 1.3.3 Células de código

- O Jupyter Notebook utiliza células para escrever e executar o código.
- Cada célula pode conter código Python que pode ser executado independentemente.
- Digite seu código em uma célula e pressione **Shift + Enter** para executá-la.

### 1.3.4 Markdown

- Além das células de código, você também pode adicionar células de texto formatado usando a sintaxe Markdown.
- Selecione **Markdown** no menu dropdown na barra de ferramentas superior para criar uma célula de texto.

### 1.3.5 Executando células

- As células de código podem ser executadas individualmente ou em sequência.
- Para executar uma célula, selecione-a e pressione **Shift + Enter**.
- O resultado da célula será exibido logo abaixo dela.

### 1.3.6 Considerações finais

Espero que este tutorial básico tenha fornecido uma introdução útil ao uso do PyCharm e do Jupyter. Ambas as ferramentas têm suas próprias vantagens e podem ser úteis em diferentes contextos de desenvolvimento e análise de dados em Python.