[Aula 2] Linguagem de Programação

Análise e Desenvolvimento de Sistemas

Introdução à Programação em Python – Prof. Jean Zahn

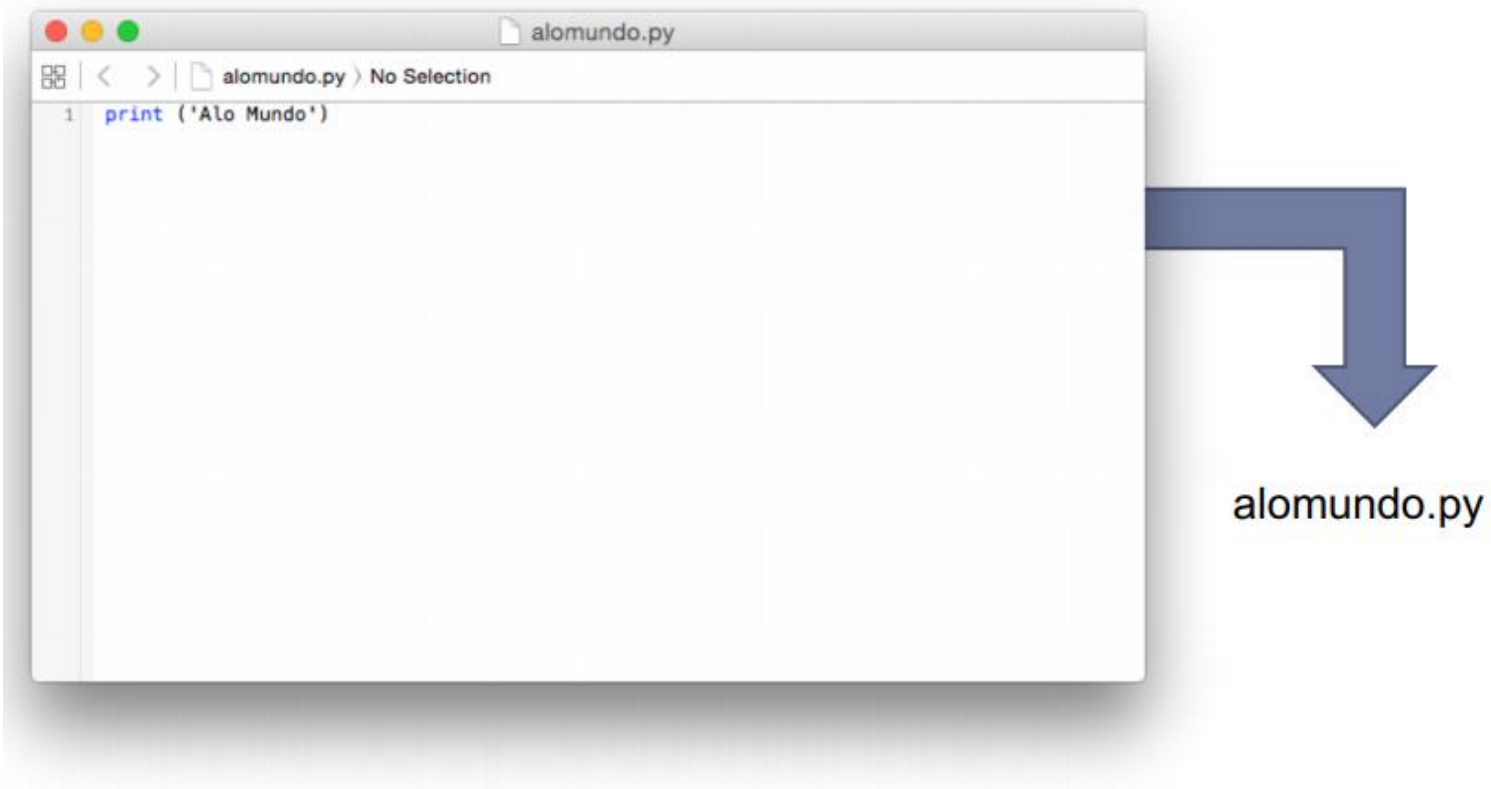
jeanozahn@gmail.com

Vamos programar em Python! Mas...

- Como um programa é organizado?
- Quais são os tipos de dados disponíveis?
- Como variáveis podem ser declaradas?
- Como atribuir valores às variáveis?
- Como entrada e saída básica de dados podem ser feitas?

Vamos começar com um exemplo...

Primeiro passo: escrever o programa!

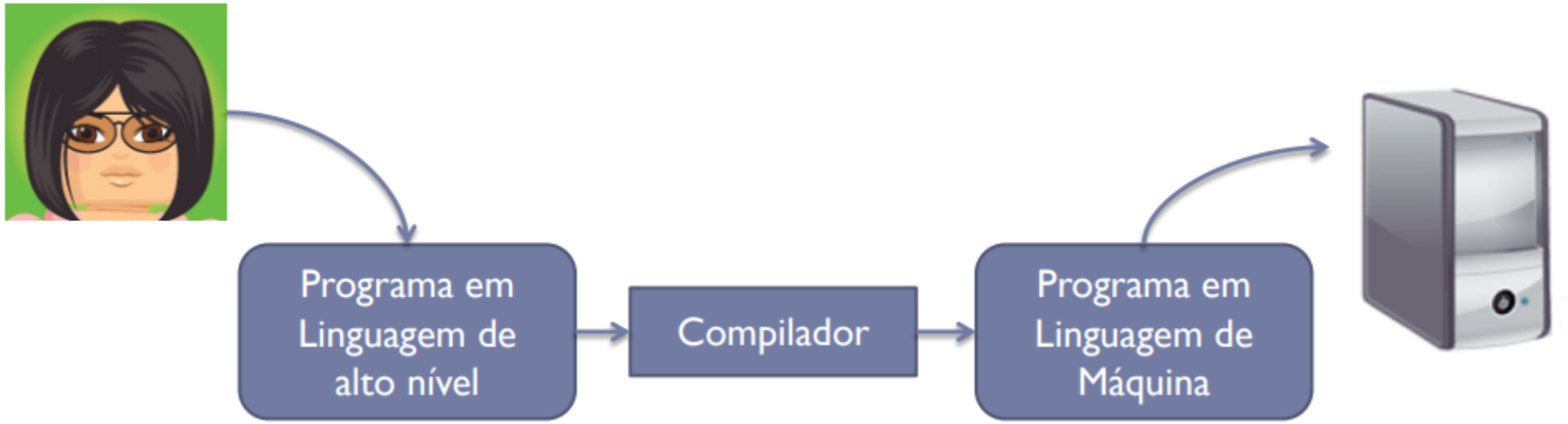


Mas o computador não conhece Python!!!

- O computador só entende binário
  - Linguagem de zeros e uns
  - 01001001110101010100101010101, entendeu?
- Precisamos traduzir o programa Python para binário

Compilação

- Na maioria das linguagens, antes de executar um programa, é necessário compilar o programa
- O compilador gera um arquivo "executável"
  - Esse novo arquivo é o que será de fato executado

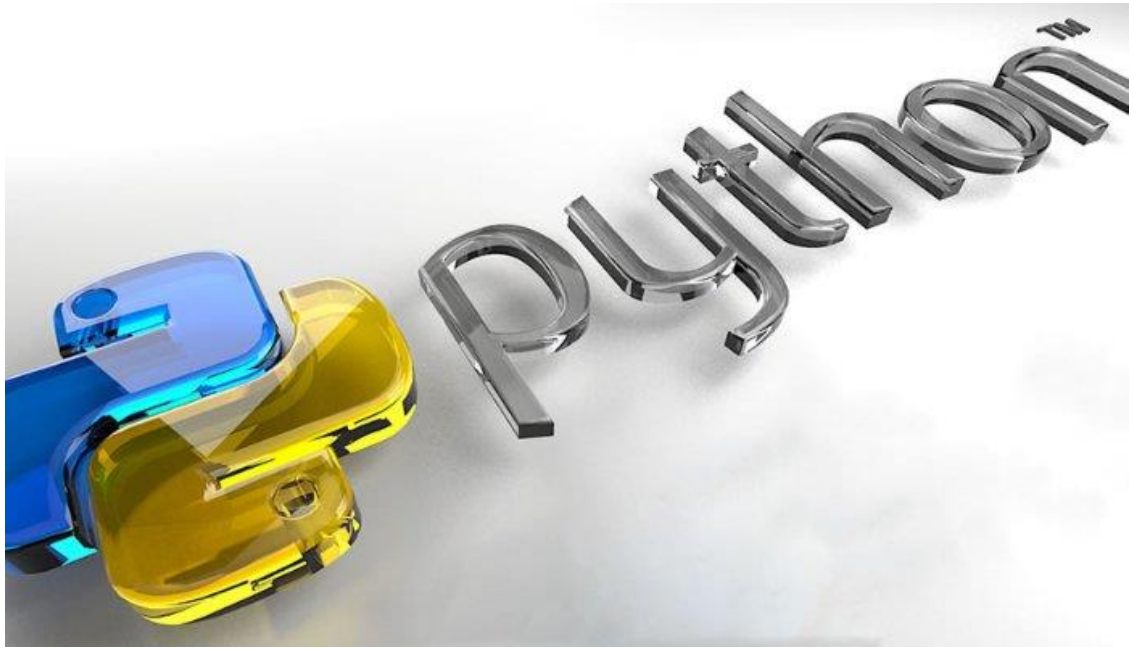


Python é uma linguagem interpretada

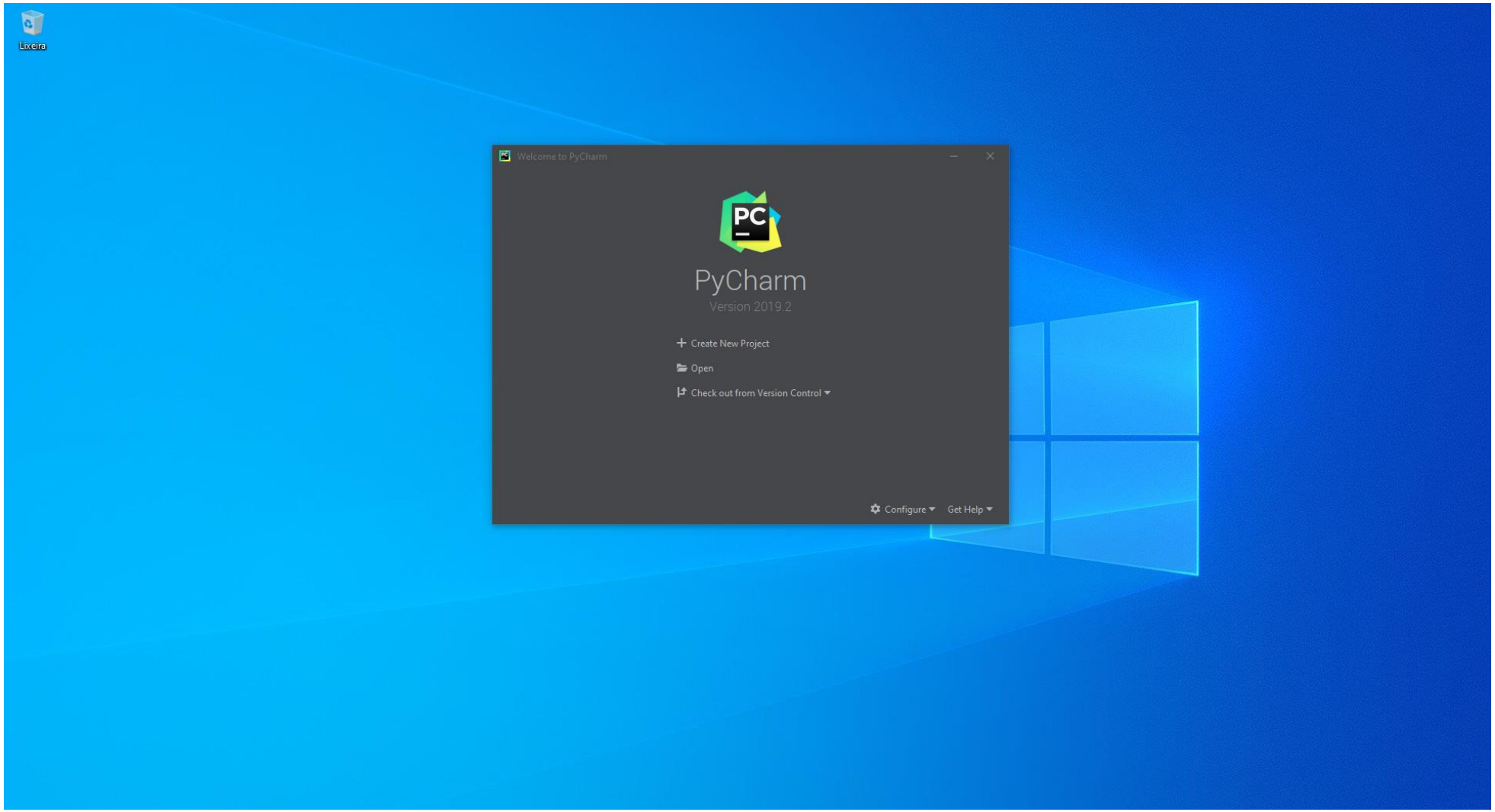
- Não é necessário compilar o código Python
- O interpretador Python vai lendo o código fonte, traduzindo para linguagem de máquina e executando ao mesmo tempo

Instalação do Interpretador Python

- Download do Python 3.7.4
  - http://www.python.org/downloads/



Vamos começar



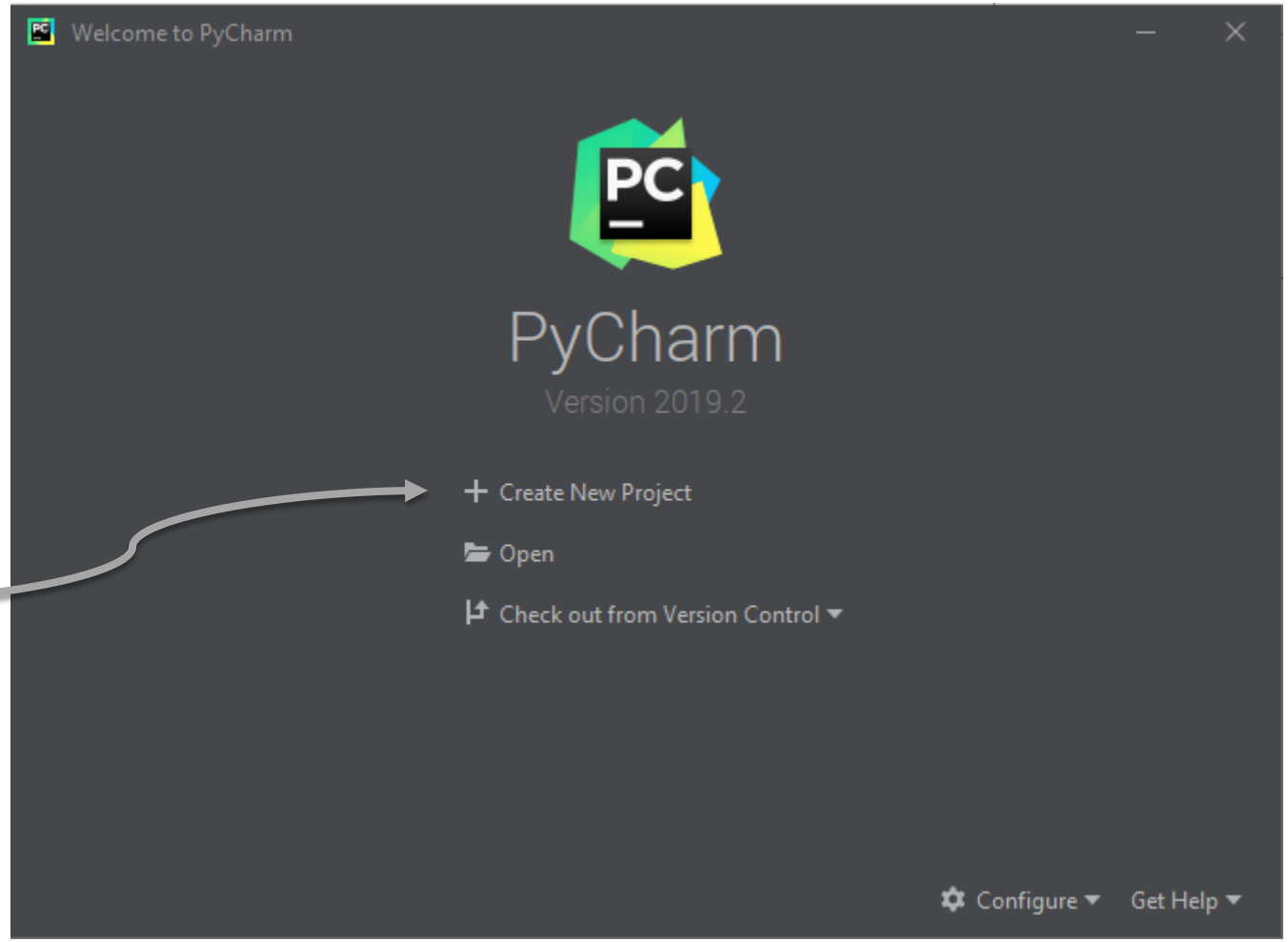
Notepad x IDE

- Dificuldades do Notepad
  - Editor básico, sem ajuda para programar
  - Execução externa
- *Integrated Development Environment* (IDE)

Instalação do PyCharm

- Usaremos o PyCharm nas aulas, mas os alunos podem optar por qualquer outra IDE ou editor
- Download do PyCharm
  - https://www.jetbrains.com/pycharm-edu/download/

Criando um projeto no PyCharm



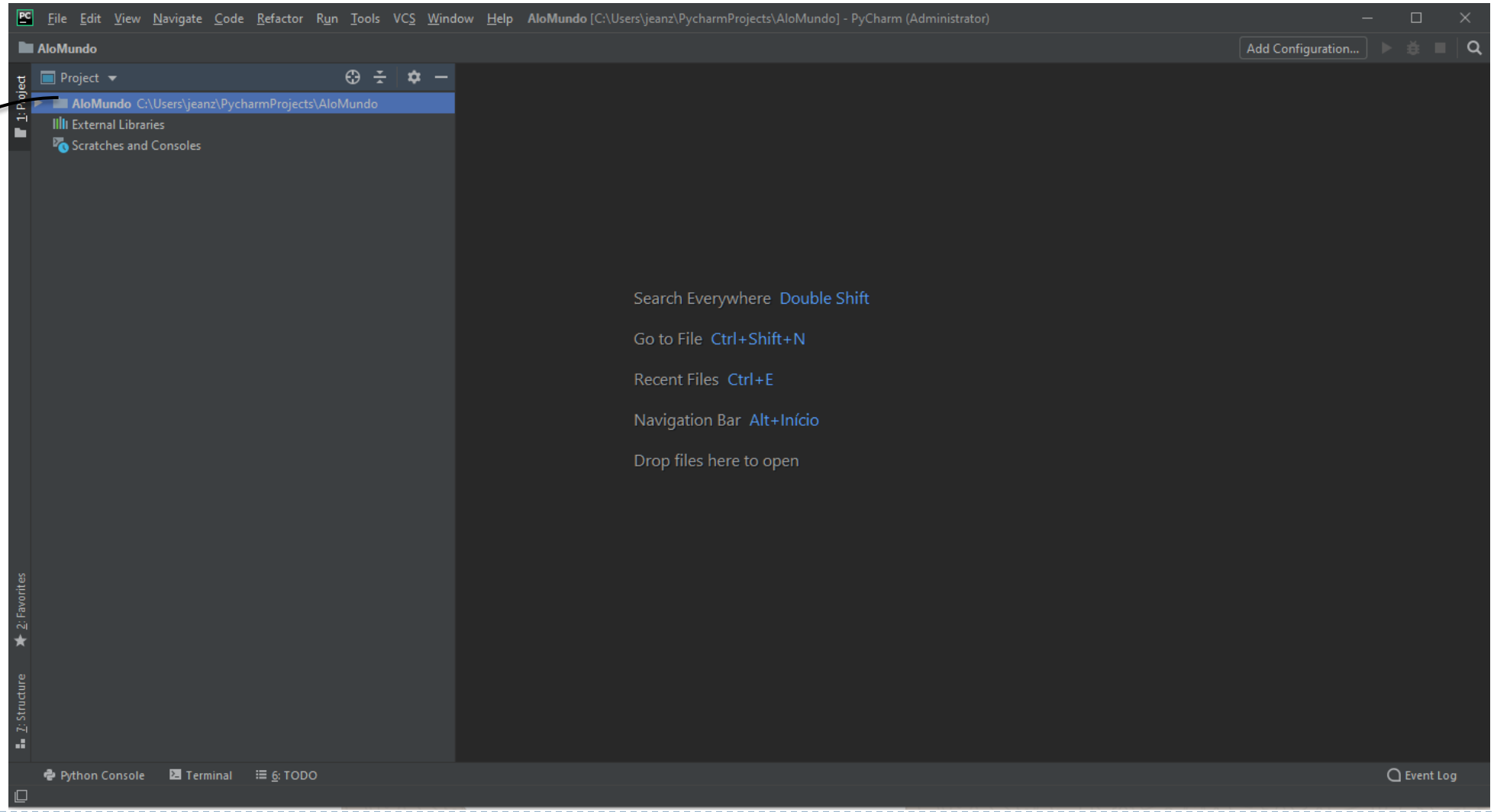
Selecionar categoria
Create New Project

Criando um projeto no PyCharm

Definir o nome e local do **projeto** e clicar e **Create** ao final

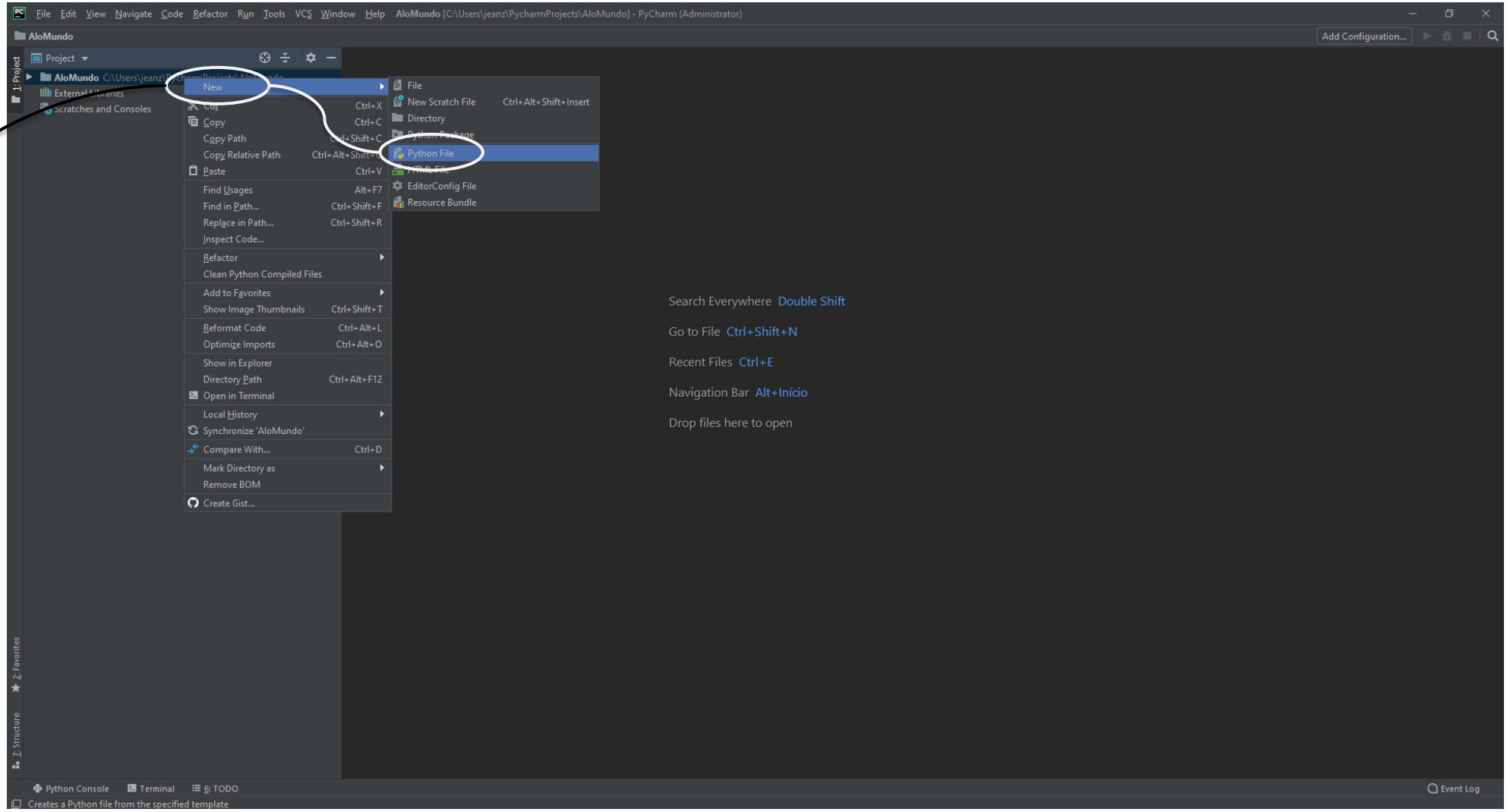
Criando um Arquivo Python no Projeto

Clicar com o **botão da direita** sobre o nome do projeto



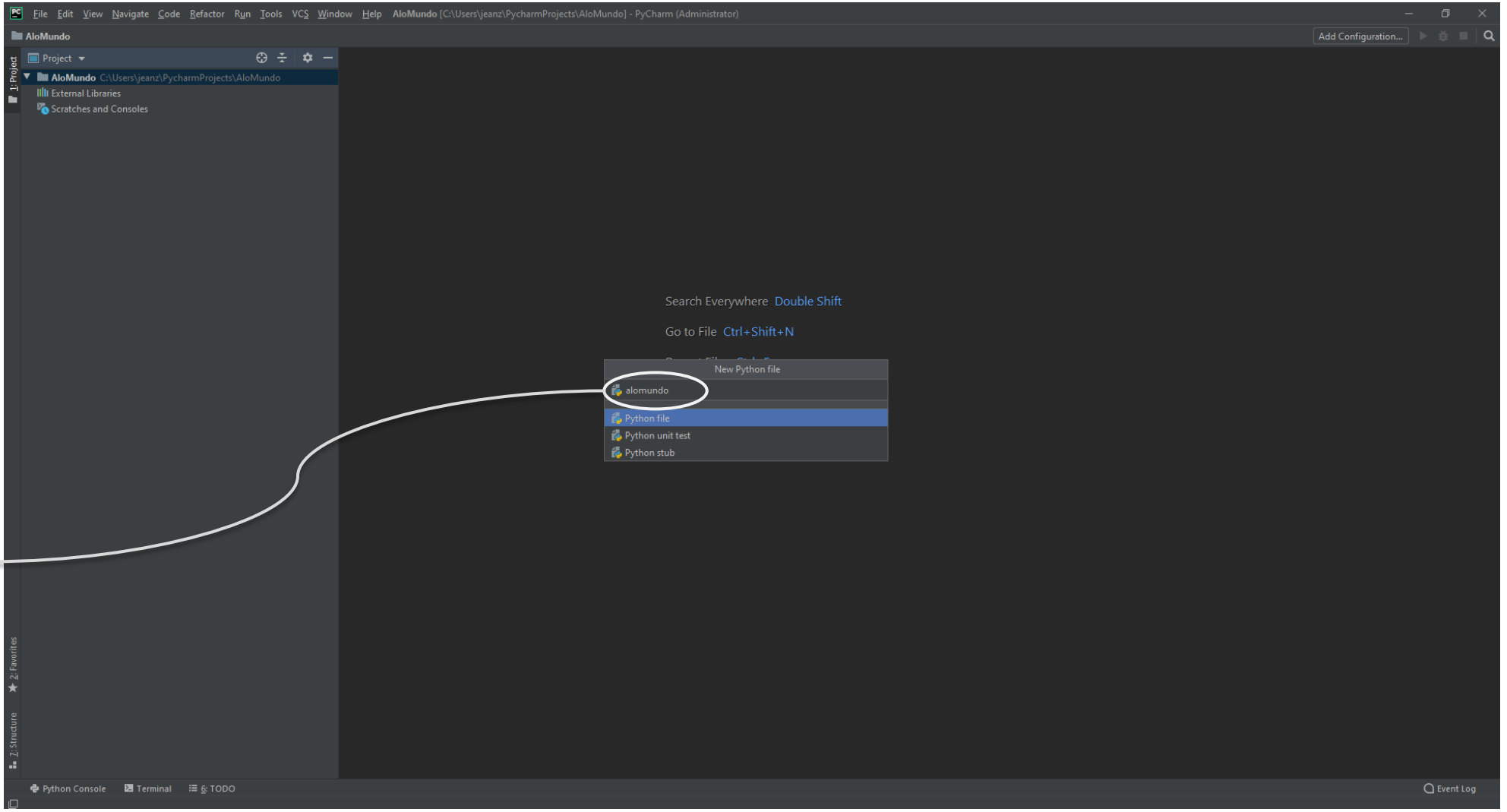
Criando um Arquivo Python no Projeto

Selecionar **New/Python File**

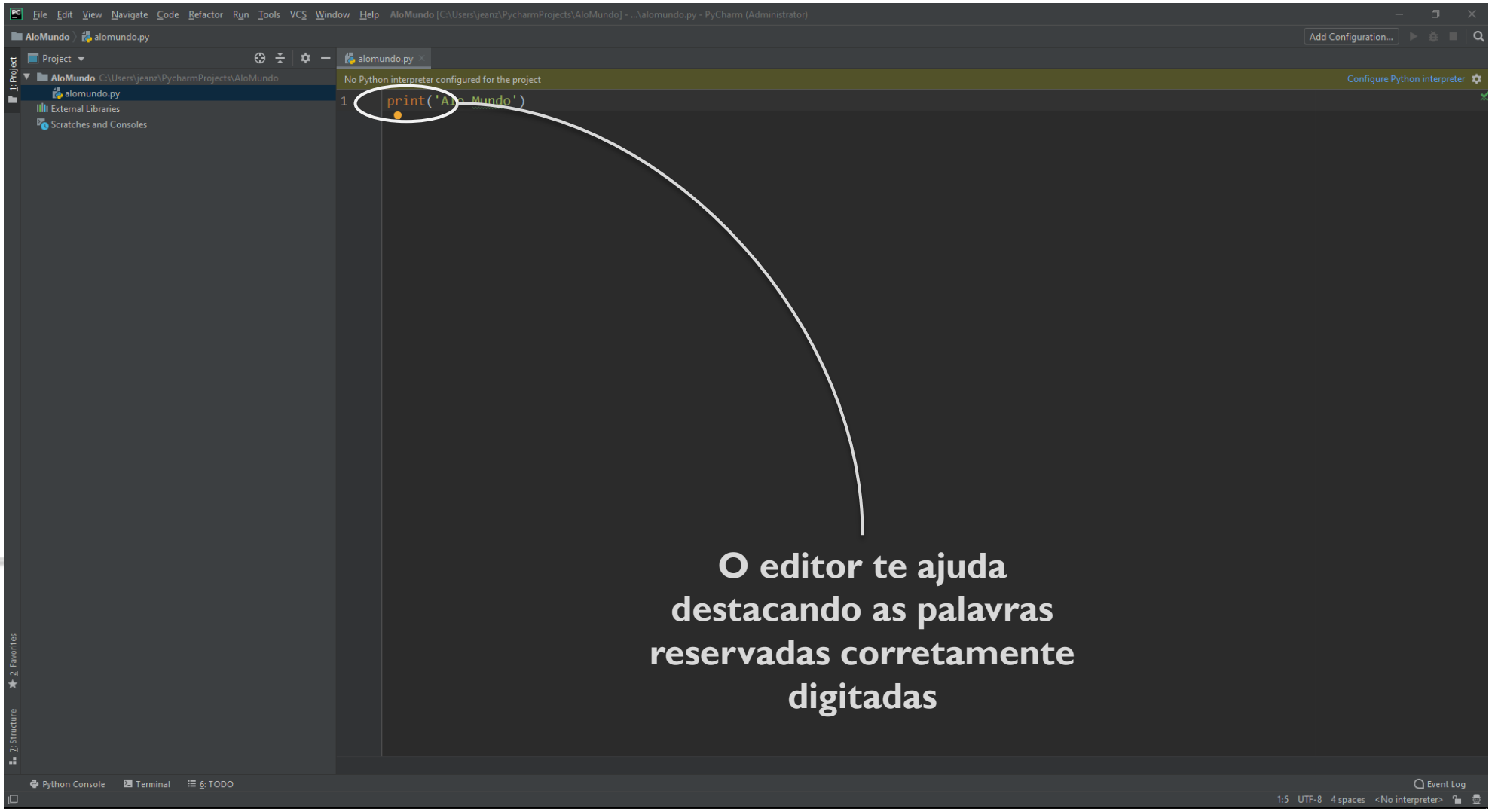


Criando um Arquivo Python no Projeto

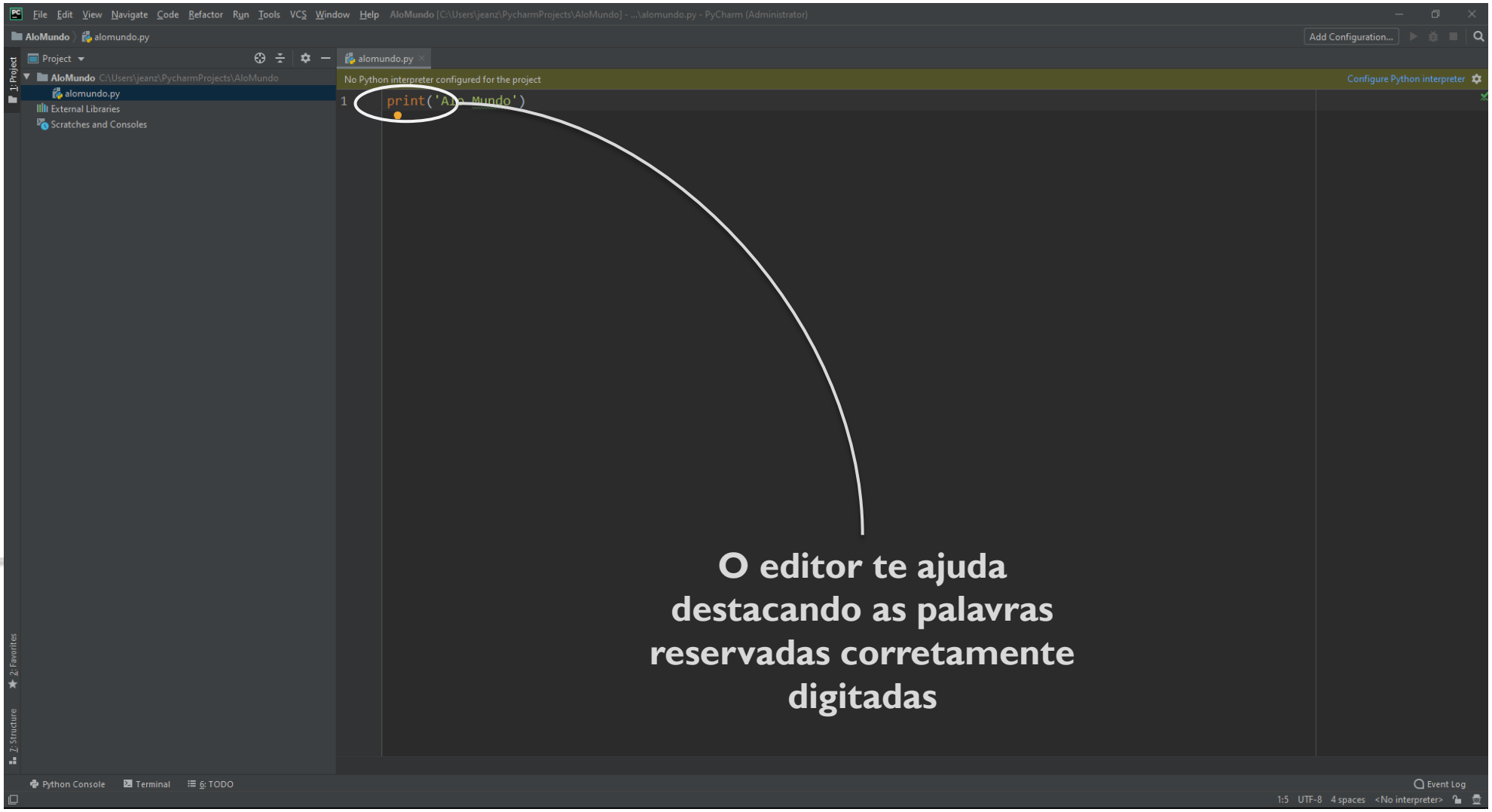
Informar o **nome** do arquivo e depois pressionar **Enter**



Criando um Arquivo Python no Projeto



Criando um Arquivo Python no Projeto



Regras básicas

- Sequência dos comandos é importante
- Blocos devem ser criados usando endentação (com espaços ou tab)

Comentários

- Comentários são trechos do programa voltados para a leitura por humanos, e ignorados pelo interpretador
- Começam com o símbolo #
  - Tudo na linha após # é ignorado pelo interpretador
- Use comentários para documentar seu código e fazer com que ele seja fácil de entender por outras pessoas

Atribuição de valores

- Em Python, o operador de igualdade (=) é usado para atribuir valores às variáveis (são os escaninhos usados pela secretária!)
- É equivalente ao símbolo de atribuição (←) que usávamos no pseudocódigo
- Sempre na forma: **variável = valor ou expressão**
  - A expressão do lado direito é processada
  - O valor gerado é atribuído à variável

Exemplo de programa em Python

```
# Este programa calcula a area de um triangulo
retangulo
altura = 15
base = 3
area = (altura * base)/2
print(area)
```

Quais são os tipos de dados disponíveis?

- Em Python, toda variável tem um tipo
- Com isso, o computador pode saber quais operações são permitidas
- Os tipos podem ser divididos em três grupos
  - Tipos numéricos (inteiro, float, ...)
  - Tipos textuais (caractere e string)
  - Tipo lógico (booleano)
- Os tipos são definidos dinamicamente, pelo próprio Python
  - Não é preciso dizer de que tipo é cada variável

Exemplo de variáveis lógicas (**boolean**)

```
x = True
y = False
```

Exemplo de variáveis textuais (**string**)

```
nome = 'Maria'
sobrenome = "Silva"
letra = 'A'
texto = 'Alo Mundo'
```

Exemplos de variáveis numéricas

```
a = -5
b = 10
c = 200
d = -12312312
e = 345092834
f = 2.5
g = 0.6023e24
h = 0.4e-3
```

Tipagem Dinâmica

```
a = -5 → inteiro
b = 10 → inteiro
c = 200 → inteiro
d = -12312312 → inteiro
e = 345092834 → inteiro
f = 2.5 → float
g = 0.6023e24 → float
h = 0.4e-3 → float
```

- Tipo é determinado **automaticamente** pelo Python no momento de criação da variável

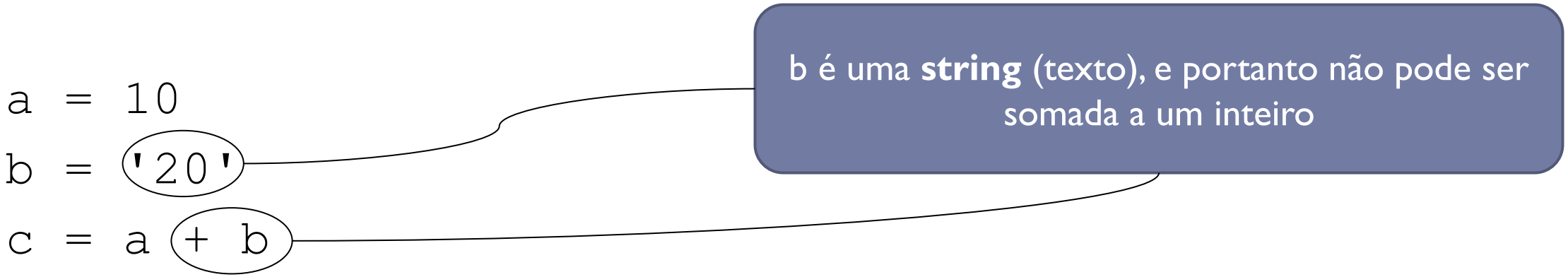
Tipagem Forte

- Uma vez que uma variável tenha um valor de um tipo, ele não pode ser usado como se fosse de outro tipo
- Exemplo:

```
a = 10
b = '20'
c = a + b
```

Tipagem Forte

- Uma vez que uma variável tenha um valor de um tipo, ele não pode ser usado como se fosse de outro tipo
- Exemplo:




```
Traceback (most recent call last):
   File "<stdin>", line 1, in <module>
 TypeError: unsupported operand type (s) for +: 'int' and 'str
```

Regras para nomes de variáveis

- Os nomes de variáveis devem respeitar algumas regras
  - São sensíveis a caixa
  - Podem ter tamanho ilimitado (mas evite abusos)
  - Devem começar com letra ou underline (_)
  - Outros caracteres podem ser letras, números ou underline
  - Não podem ter espaço nem acentos
  - Não podem ser uma palavra reservada da linguagem

Entrada de dados

- Para entrada de dados, usamos **`input`**
- É possível informar um texto que aparecerá impresso na tela para que o usuário saiba que o programa está esperando a entrada de um valor

```
nome = input('Digite o nome do aluno: ')
print(nome)
```

Input lê dados como string

- Você pode usar o comando `type` para saber o tipo que o Python atribuiu a uma variável

```
altura = input('Digite a altura do triangulo: ')
print(type(altura))
base = input('Digite a base do triangulo: ')
print(type(base))
...
```

Mudança de tipo

- Usar `int()`, `float()` ou `eval()` para fazer o Python ler variáveis de tipo numérico

```
altura = int(input('Digite a altura do triangulo: '))
print(type(altura))
base = int(input('Digite a base do triangulo: '))
print(type(base))
area = (base * altura)/2
print('A area do triangulo eh: ', area)
```

Saída de dados

- Para saída de dados, usamos `print`

Exemplo de entrada e saída de dados

```
print(‘Linguagem de Programação é muito legal')
print(123)
altura = 10
print(altura)
print('Vamos pular uma linha \n')
print('O nome do aluno eh', nome)
```

Voltando ao exemplo de programa em Python

```
altura = int(input('Digite a altura do triangulo: '))
base = int(input('Digite a base do triangulo: '))
area = (base * altura)/2
print('A área do triangulo é:', area)
```

Formatação de Números

- É possível especificar uma máscara no comando print para imprimir números com um determinado formato
- Pode-se, por exemplo, fazer com que um `float` seja impresso com apenas duas casas decimais
  - `print("%.2f" % variável)`
  - `f` é usado para números do tipo float
  - `d` é usado para números inteiros
  - `s` é usado para strings

Voltando ao exemplo de programa em Python

```
altura = int(input('Digite a altura do triangulo: '))
base = int(input('Digite a base do triangulo: '))
area = (base * altura)/2
print('Altura = %4d' % altura)
print('Base = %4d' % base)
print(‘A área do triangulo é %.2f' % area)
```

Imprimindo várias variáveis ao mesmo tempo

```
altura = int(input('Digite a altura do triangulo: '))
base = int(input('Digite a base do triangulo: '))
area = (base * altura)/2
tipo = "retangulo"
print('A área do triangulo %s de altura %.0f e base %.0f é: %.2f' % (tipo,
altura, base, area))
```

Exercícios

- Qual a saída do programa abaixo?

```
x = 1.0
y = 2.0
z = 3.0


x = -x
y = y - 1
z = z + x
z = z + x - y
print("x =", x, ", y =", y, ", z =", z)
```

Exercícios

1. Faça um programa que leia o nome, a idade, a altura, o peso e a nacionalidade do usuário e escreva essas informações na forma de um parágrafo de apresentação
2. Faça um programa que exiba o perímetro de uma circunferência a partir do seu raio
3. Faça um programa que leia dois pontos num espaço bidimensional e calcule a distância entre esses pontos

5. Faça um programa para, a partir de um valor informado em centavos, indicar a menor quantidade de moedas que representa esse valor
   - Considere moedas de 1, 5, 10, 25 e 50 centavos, e 1 real
   - Exemplo: para o valor 290 centavos, a menor quantidade de moedas é 2 moedas de 1 real, 1 moeda de 50 centavos, 1 moeda de 25 centavos, 1 moeda de 10 centavos e 1 moeda de 5 centavos

Referências

- Aula baseada no material disponibilizado por Leonardo Murta, Aline Paes e Vanessa Braganholo. Instituto de Computação – Universidade Federal Fluminense

[Aula 2] Linguagem de Programação

Análise e Desenvolvimento de Sistemas

Introdução à Programação em Python – Prof. Jean Zahn

jeanozahn@gmail.com