



Processamento da Informação

Ambientes de programação

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q1/2020



Apresentação

Apresentação

- **Professor:**

Jesús P. Mena-Chalco (CMCC)

jesus.mena@ufabc.edu.br

- **Formação:**

- Engenheiro da Computação.
- Mestre e Doutor em Ciência da Computação.
Instituto de Matemática e Estatística da USP.

- Sala 517-A, torre 2, 5º Andar.

- **Áreas de pesquisa:**

- Pattern recognition
- Graph mining
- Scientometrics/Bibliometrics

Aprender a programar: considerações

- Fall in love with mathematics (pratique matemática)
- **Be self-motivated** (trabalhe com pares)
- Never back down (seja persistente)
- Become a master (ensine aos colegas)
- Be a bookworm (seja leitor ávido)

Leia as seguintes sugestões:

<http://www.wikihow.com/Learn-a-Programming-Language>

Páginas importantes

- <http://professor.ufabc.edu.br/~jesus.mena/courses>

Procurar: “2020/Q1”

- <https://ava.ufabc.edu.br/>

Procurar pelo nome de sua turma:

- DA4-SA, DA5-SA, DA6-SA ou
- DB4-SA, DB5-SA, DB6-SA

Créditos (T-P-I)=(3-2-5)

Cada disciplina na UFABC é representada por três algarismos: T-P-I

- **T:** Número de horas semanais de aulas expositivas presenciais da disciplina (teóricas)
- **P:** Número médio de horas semanais de trabalho de laboratório, aulas práticas ou aulas de exercícios, realizadas em sala de aula (práticas)
- **I:** Estimativa de horas semanais adicionais de trabalhos necessárias para o bom aproveitamento da disciplina (estudos e trabalhos)

Calendário

Fevereiro

	Dom	Seg	Ter	Qua	Qui	Sex	Sáb
							1
	2	3	4	5	6	7	8
1	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22
3	23	24	25	26	27	28	29

Março

	Dom	Seg	Ter	Qua	Qui	Sex	Sáb
4	1	2	3	4	5	6	7
5	8	9	10	11	12	13	14
6	15	16	17	18	19	20	21
7	22	23	24	25	26	27	28
8	29	30	31				

Abril

	Dom	Seg	Ter	Qua	Qui	Sex	Sáb
8				1	2	3	4
9	5	6	7	8	9	10	11
10	12	13	14	15	16	17	18
11	19	20	21	22	23	24	25
12	26	27	28	29	30		

Maio

	Dom	Seg	Ter	Qua	Qui	Sex	Sáb
12						1	2
13	3	4	5	6	7	8	9
14	10	11	12	13	14	15	16
	17	18	19	20	21	22	23
	24	25	26	27	28	29	30
	31						

Teoria

Prática

Agenda

Aula	Semana	Dia	Modalidade	Tópicos - Alinhamento	
1	1	11/fev	Terça	Prática	Ambiente de trabalho e Programas sequenciais
2	1	12/fev	Quarta	Teoria - sem. I	Apresentação da disciplina e Algoritmos sequenciais
3	1	14/fev	Sexta	Teoria	Modularização e Estruturas de seleção - Parte 1
4	2	18/fev	Terça	Prática	Estruturas de seleção - Parte 1
5	2	21/fev	Sexta	Teoria	Estruturas de seleção - Parte 2
6	3	28/fev	Sexta	Teoria	Estruturas de repetição - Parte 1
7	4	03/mar	Terça	Prática	Estruturas de seleção - Parte 2 / repetição Parte 1
8	4	06/mar	Sexta	Teoria	Estruturas de repetição - Parte 2
9	5	10/mar	Terça	Prática	Estruturas de repetição - Parte 2
10	5	11/mar	Quarta	Teoria - sem. I	Revisão
11	5	13/mar	Sexta	Teoria	P1 - Teoria
12	6	17/mar	Terça	Prática	P1 - Prática
13	6	20/mar	Sexta	Teoria	Vetores - Parte 1
14	7	24/mar	Terça	Prática	Vetores - Parte 1
15	7	25/mar	Quarta	Teoria - sem. I	Vetores - Parte 2
16	7	27/mar	Sexta	Teoria	Vetores - Parte 3
17	8	31/mar	Terça	Prática	Vetores - Partes 2 e 3
18	8	03/abr	Sexta	Teoria	Matrizes - Parte 1
19	9	07/abr	Terça	Prática	Matrizes - Parte 1a
20	10	14/abr	Terça	Prática	Matrizes - Parte 1b
21	10	17/abr	Sexta	Teoria	Matrizes - Parte 2
22	11	22/abr	Quarta	Teoria - sem. I	Recursão - Parte 1
23	11	24/abr	Sexta	Teoria	Recursão - Parte 2
24	12	28/abr	Terça	Prática	Recursão
25	Reposição	06/mai	Quarta	Prática	P2 - Prática
26	Reposição	07/mai	Quinta	Teoria	Exercícios
27	Reposição	08/mai	Sexta	Teoria	Revisão
28	Reposição	11/mai	Segunda	Teoria	P2 - Teoria
29	Reposição	13/mai	Quarta	Prática	SUB - Prática
30	Reposição	14/mai	Quinta	Teoria	SUB - Teoria
		20/jun	Sábado	Teoria+Prática	REC - Sábado às 10h (sala a ser definida)

Avaliação

Parte de Teoria: Apresentarei amanhã.

Parte de Prática: Duas provas.

- Prova 1 (50%): 17/03/2020
 - Prova 2 (50%): 06/05/2020
 - Bônus: 2 desafios de programação (10%)
-
- Prova Substitutiva: 13/05/2020
 - Prova de Recuperação: Q3/2020 → 20/06/2020 às 10h

Avaliação

Ver o plano de ensino disponível na página da disciplina.

$$\text{MF} = 0,35 * \text{Teoria} + 0,35 * \text{Prática} + 0,3 \text{ Listas de exerc.}$$

- **A:** nota ≥ 9
- **B:** $7,5 \leq \text{nota} < 9$
- **C:** $6 \leq \text{nota} < 7,5$
- **D:** $5,0 \leq \text{nota} < 6$
- **F:** nota $< 5,0$

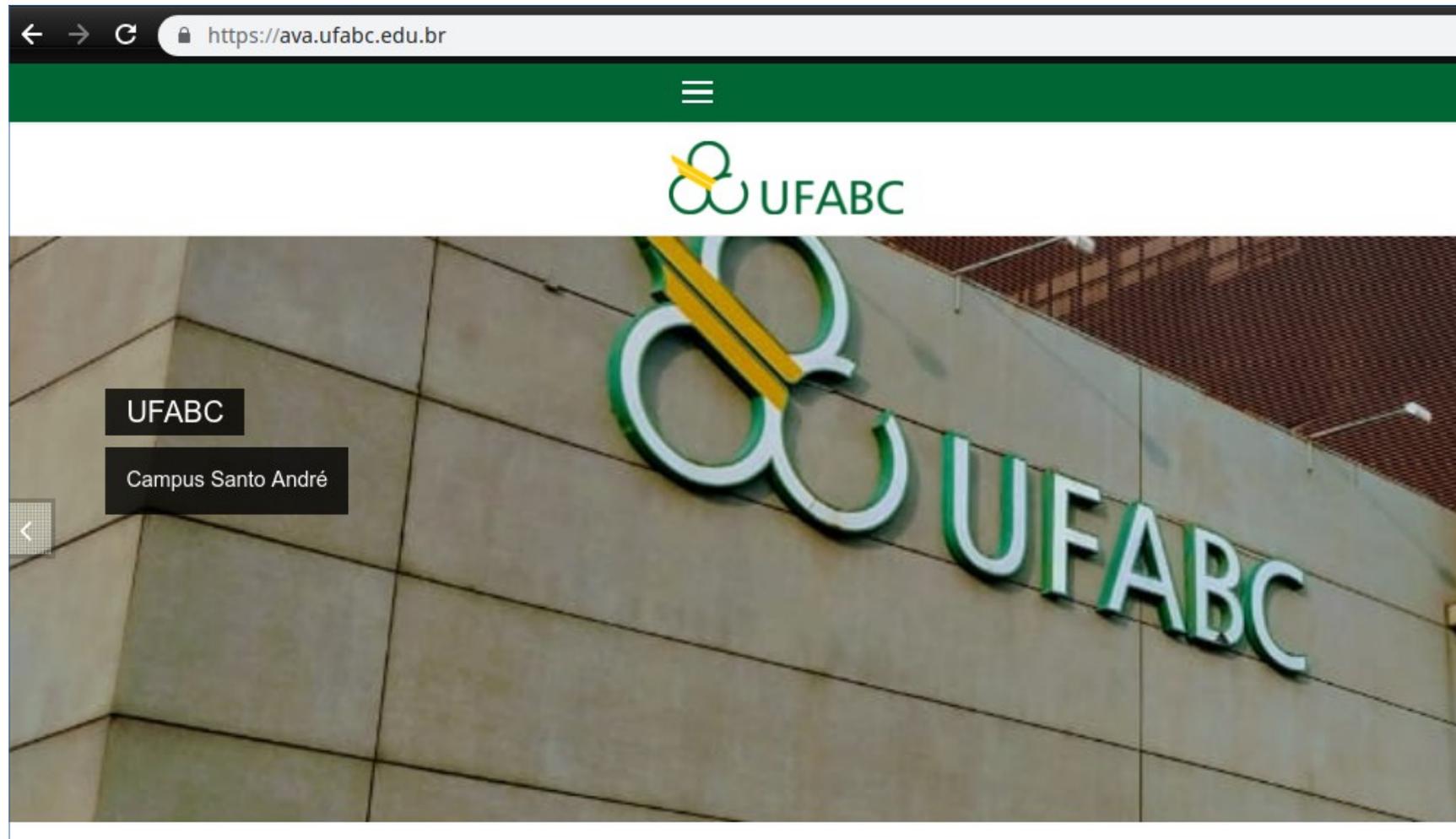
Listas de exercícios

- Lista 1: 18/fev Entrega: 03/mar
- Lista 2: 03/mar Entrega: 10/mar
- Lista 3: 10/mar Entrega: 17/mar
- Lista 4: 24/mar Entrega: 31/mar
- Lista 5: 31/mar Entrega: 07/abr
- Lista 6: 07/abr Entrega: 14/abr
- Lista 7: 14/abr Entrega: 28/mar
- Lista 8: 28/abr Entrega: 13/mai

A nota das listas é média aritmética dos exercícios.

Listas para treino (não valen nota no lab)

Temos adicionalmente Exercícios-Programa (EPs) que serão considerados para treino: <https://ava.ufabc.edu.br>



Bibliografia

Bibliografia Básica

- Forbellone, A. L. V.; Eberspächer, H. F.; **Lógica de Programação - A Construção de Algoritmos e Estruturas de Dados**; 3ª edição, Editora Pearson Prentice-Hall, 2005
- Sebesta, R. W.; **Conceitos de Linguagens de Programação**; 5ª edição, Editora Bookman, 2003

Bibliografia Complementar

- Ascensio, A.F.; Campos, E.A., **Fundamentos da Programação de Computadores**, Pearson, 3a edição, 2012.
- Puga, S., **Lógica de programação e estruturas de dados com aplicações em Java**, Pearson Prentice-Hall, 2a edição, 2009.

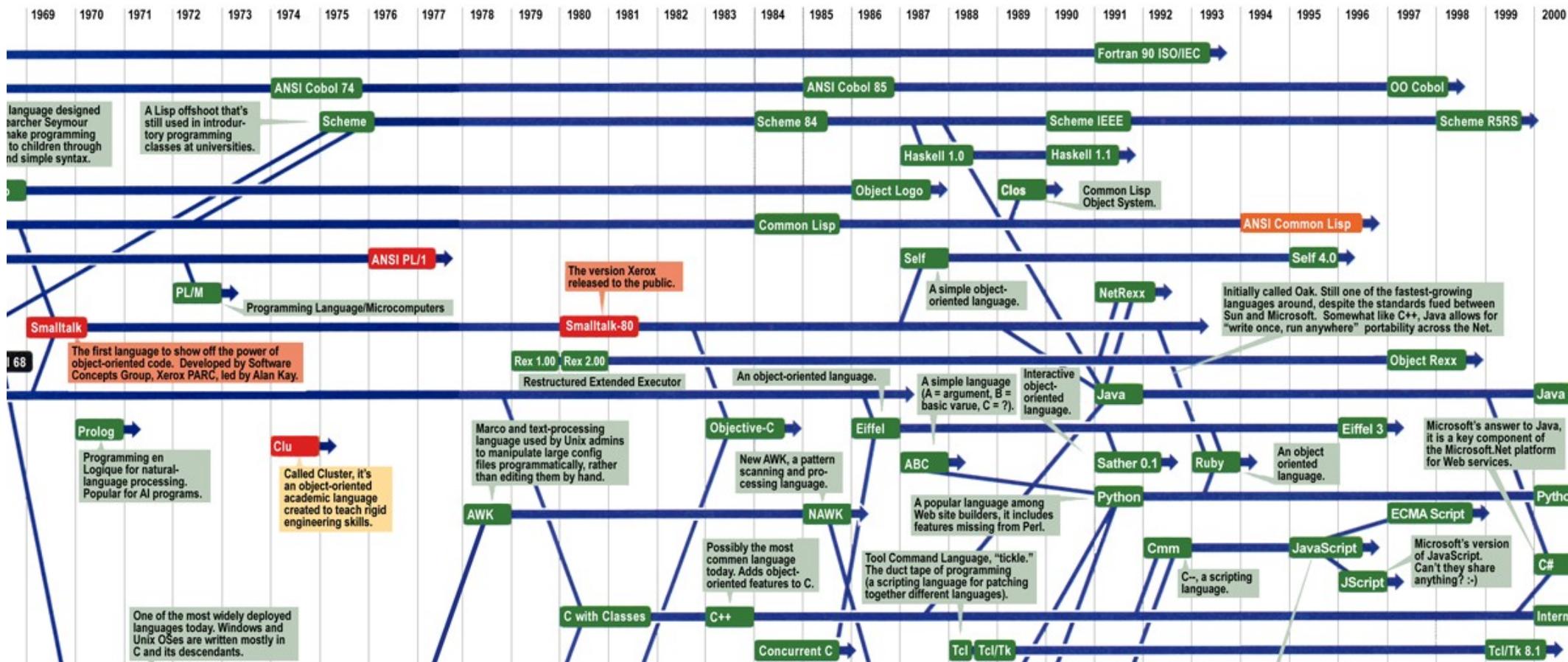


Qual linguagem usar para programar?

Linguagens de programação



Linguagens de programação



(*) <http://www.digibarn.com/collections/posters/tongues/>

Linguagens de programação

The Evolution Of Computer Programming Languages

```
00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59
60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79
80 81 82 83 84 85 86 87 88 89
90 91 92 93 94 95 96 97 98 99
```



Hex

```
0000 0000 0000 0000
0001 0000 0000 0000
0002 0000 0000 0000
0003 0000 0000 0000
```



Assembler

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



C

```
PROGRAM NAME, DATE, TIME
PROGRAM NAME, DATE, TIME
PROGRAM NAME, DATE, TIME
PROGRAM NAME, DATE, TIME
```



Fortran

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



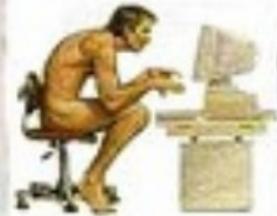
C++

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



Java

```
Module "main.c"
Module "lib.c"
Module "utils.c"
Module "main.c"
Module "lib.c"
```



Linguagens mais utilizadas

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0

Por que programar em Python?



Q why python is|

- Q why python is **so popular**
- Q why python is **so slow**
- Q why python is **used**
- Q why python is **called python**
- Q why python is **important**
- Q why python is **slow**
- Q why python is **bad**
- Q why python is **interpreted language**
- Q why python is **scripting language**
- Q why python is **better than java**



Nos buscadores web encontrarão confirmação para quase todas as hipóteses levantadas...

Quase todas sendo confirmadas...

O **importante não é a linguagem** mas a forma de resolver computacionalmente um problema.

Focaremos mais na **lógica de programação**, não usaremos nenhuma estrutura sofisticada.

Quem usa python?

- Wikipedia
- Google
- Yahoo!
- CERN
- NASA
- Facebook
- Amazon
- Instagram
- Spotify





Ambientes de programação

Ambientes de programação

Um ambiente de programação oferece **ferramentas** de gerenciamento e desenvolvimento de software.

O objetivo é **agilizar** o desenvolvimento.

Um *Integrated Development Environment* (IDE) reúne as características básicas para um desenvolvimento rápido de software.

Ferramentas comuns:

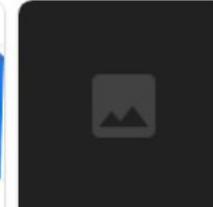
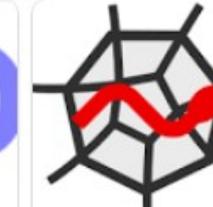
- Editor de código (editor de texto)
- Compilador
- Depurador

Qual IDE utilizar?

ide programming

All Images Videos News Shopping More Settings Tools

Integrated development environment Software

 <p>Eclipse Eclipse Publi...</p>	 <p>Microsoft Visual Studio Proprietary s...</p>	 <p>NetBeans Apache Lice...</p>	 <p>IntelliJ IDEA Apache Lice...</p>	 <p>Atom MIT License</p>	 <p>PyCharm Apache Lice...</p>	 <p>How to Get IDLE Python Soft...</p>	 <p>Xcode Freeware</p>	 <p>CODE BLOCKS GNU Gener...</p>
 <p>PyDev Eclipse Publi...</p>	 <p>Spyder MIT License</p>	 <p>RubyMine Proprietary s...</p>	 <p>Qt GNU Lesser...</p>	 <p>Wing IDE Proprietary s...</p>	 <p>MonoDevelop GNU Lesser...</p>	 <p>Cloud9 IDE Freeware</p>	 <p>Dev-C++ GNU Gener...</p>	

Ambientes de programação

IDE 1

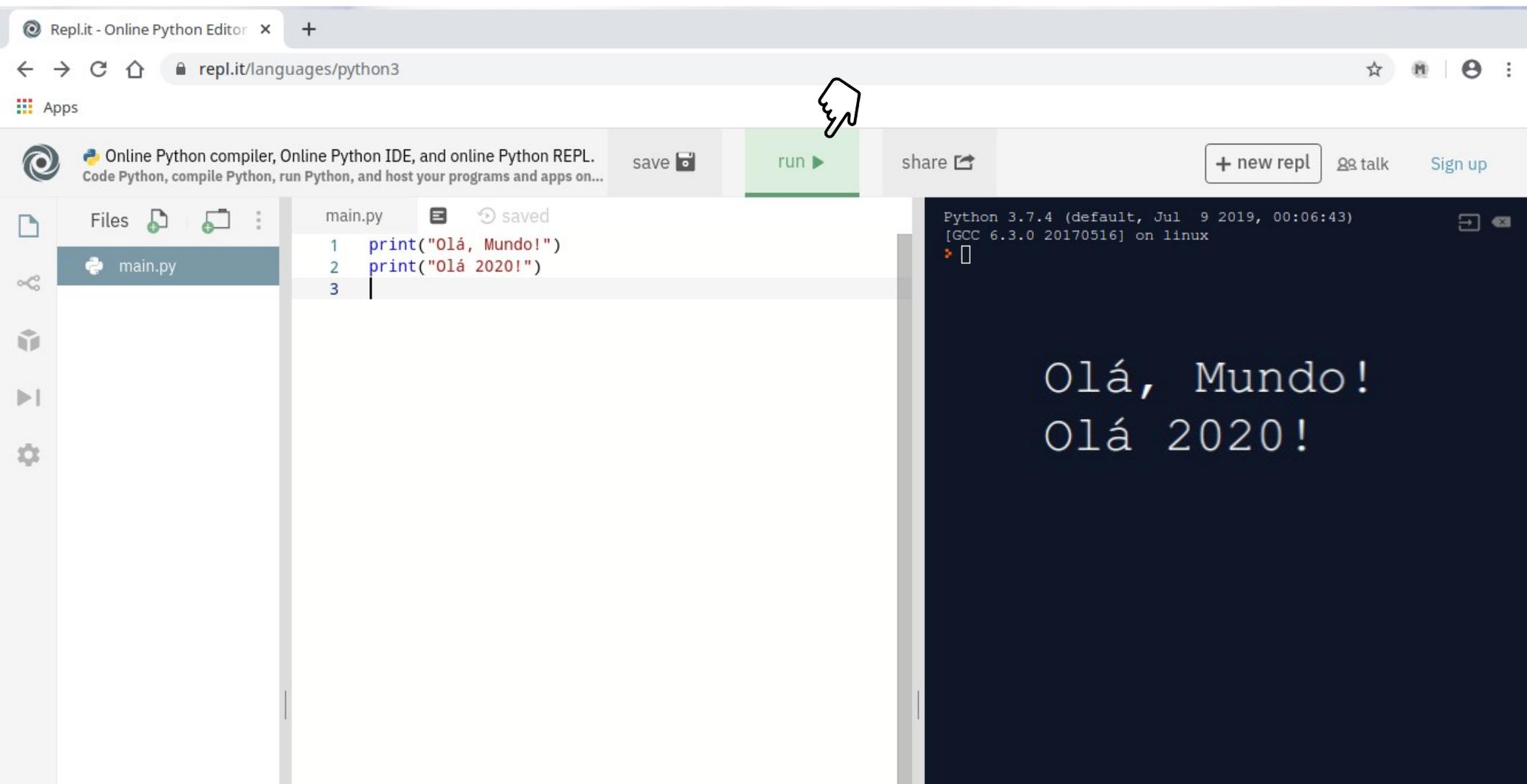


IDE 2



Em laboratório usaremos inicialmente IDEs 'simples'

(1) Repl.it (use o navegar web)



The screenshot shows the Repl.it online Python editor interface. The browser address bar displays `repl.it/languages/python3`. The editor contains a file named `main.py` with the following Python code:

```
1 print("Olá, Mundo!")
2 print("Olá 2020!")
3
```

A hand cursor is pointing to the `run` button. The terminal output shows the execution results:

```
Python 3.7.4 (default, Jul 9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux
>
Olá, Mundo!
Olá 2020!
```

<https://repl.it/languages/python3>

Teste 2

main.py



saved

```
1 nome = input("Qual o seu nome? ")
2 print("Olá, ", nome)
3
```

```
Qual o seu nome? João
Olá, João
>
```

Teste 2

main.py



saved

```
1 nome = input("Qual o seu nome? ")
2 print("Olá, ", nome)
3
```

```
Qual o seu nome? João
Olá, João
>
```

Essa janela é um **interpretador** da linguagem

Teste 3: Use a janela do interpretador

Operações básicas

1 + 1

8 - 1

10 * 2

35 / 5

Divisão inteira (arredonda para baixo)

5 // 3

-5 // 3

5.0 // 3.0



Resto da divisão

7 % 3

Exponenciação

2**3

Forma de leitura 'padrão'?



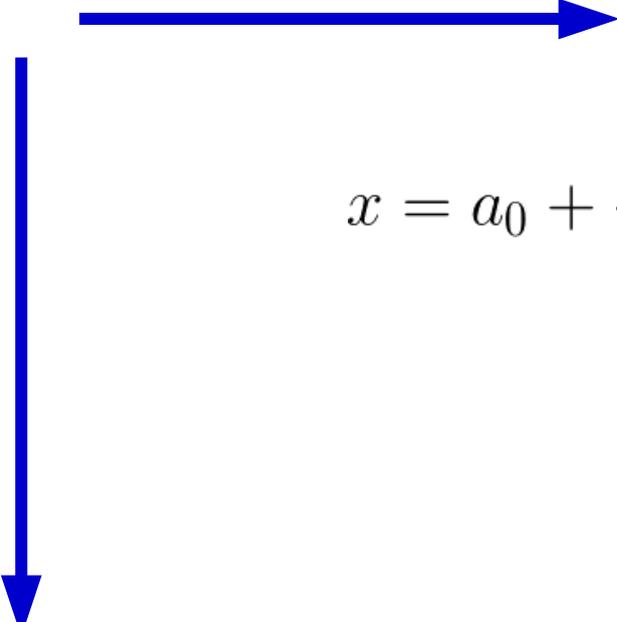
Hebraico

الجامعة الاتحادية ABC
إنها جامعة ممتازة.
البرمجة رائعة.

Árabe

Forma de leitura

- Instruções em sequência: de cima para baixo
- Leitura de esquerda para direita
- Estruturas aninhadas


$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$$

$$x = \frac{(1 * 3)}{(1 - (45 + \sqrt{34}))}$$

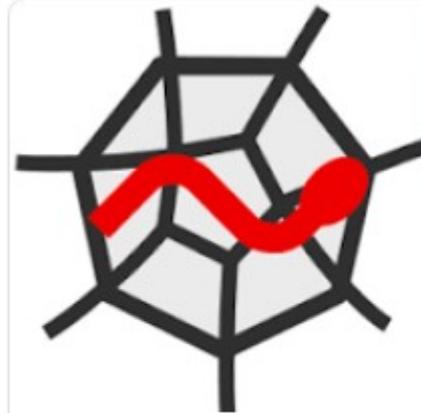
(2) Spyder

Procure o programa Spyder no seu computador.

Os computadores dos laboratórios tem esse programa instalado.

Teste os exemplos anteriores com o Spyder.

Salve o programa.



Spyder

Software

Spyder is an open source cross-platform integrated development environment for scientific programming in the Python language.

[Wikipedia](#)

License: [MIT](#)

Written in: [Python](#)

Stable release: 4.0.1 / 2 January 2020; 31 days ago

Initial release: 18 October 2009; 10 years ago

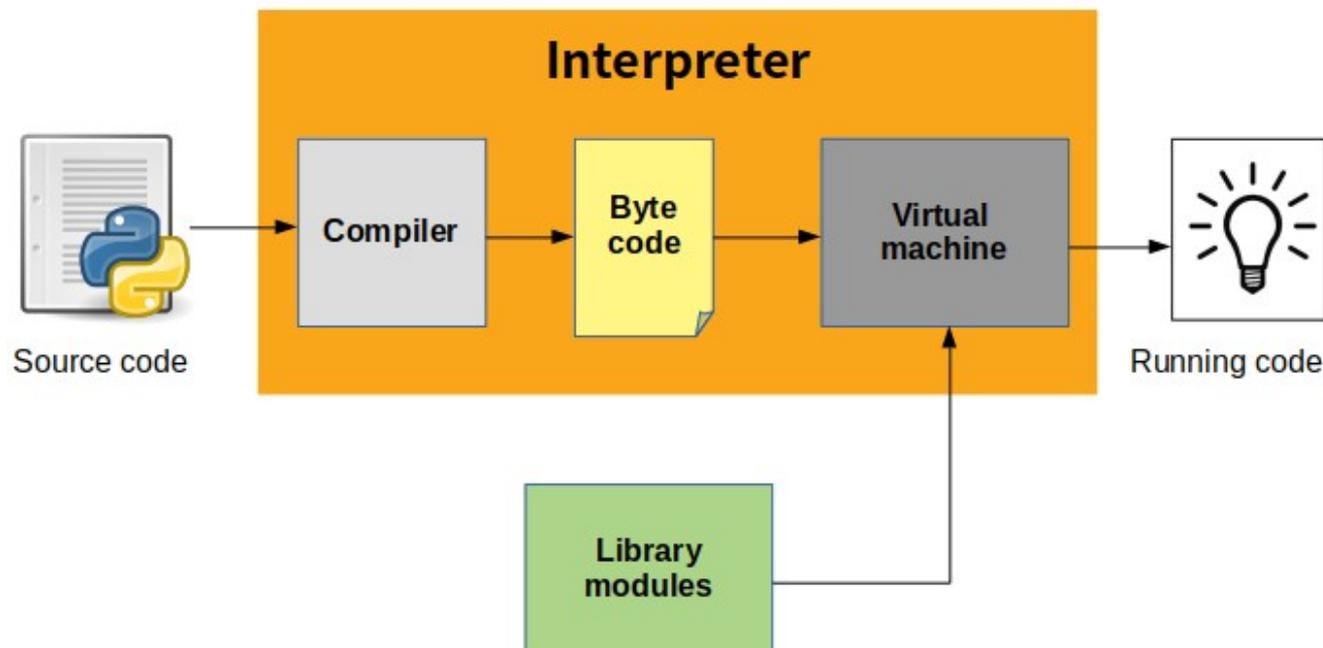
Original author(s): Pierre Raybaut

Platforms: [Qt](#), [Microsoft Windows](#), [macOS](#), [Linux](#)

Formato do arquivo: .py

O programa é, na verdade, um arquivo de **texto**, mas é necessário ter o **sufixo .py**

Quando o arquivo de texto vira, de fato, em programa?





File Edit Search Source Run Debug Consoles Projects Tools View Help

ome/jmenac/.config/spyder-py3

Editor - /home/jmenac/.config/spyder-py3/temp.py

temp.py*

```
1
2 import requests
3
4 arquivo = requests.get("http://www.ufabc.edu.br")
5
6 print(arquivo.text)
7
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our [tutorial](#)

Variable explorer File explorer Help

IPython console

Console 1/A

```
<div class="custom" >
  <ul class="listagem">
<li><strong><a href="/concursos/docentes" title="Docentes">Docentes</a></strong><br />(inscrições
abertas)</li>
<li><strong><a href="/concursos/tecnicos-administrativos" title="Técnicos Administrativos">Técnicos
Administrativos</a></strong><br />(em andamento)</li>
</ul></div>

</div>
<div class="module">
header">
<h2 class="outstanding-title">Pós-Graduação</h2>
</div>

<div class="custom" >
  <ul class="listagem">
<li><a href="http://progp.ufabc.edu.br/matriculas/" target="_blank" rel="noopener noreferrer"
title="Matriculas 1º quadrimestre de 2020">Matriculas 1º quadrimestre de 2020</a>&nbsp;</li>
<li><a href="http://progp.ufabc.edu.br/bancas-de-defesa" title="Agenda de Bancas de Defesa"
target="_blank" rel="noopener noreferrer">Agenda de Bancas de Defesa</a></li>
<li><a href="http://progp.ufabc.edu.br/processos-seletivos/" title="Processos seletivos" target="_blank"
rel="noopener noreferrer">Processos seletivos</a></li>
</ul></div>
```

IPython console History log

(3) <https://ava.ufabc.edu.br>

Sequencial (EP1)



-  EP1_1 - Alo, mundo!
-  EP1_1 [Vídeo] Submissão pela aba Editar (Python)
-  EP1_1 [Vídeo] Submissão pela aba Enviar (Python)
-  EP1_1 [Vídeo] Submissão pela aba Editar (Java)
-  EP1_1 [Vídeo] Submissão pela aba Enviar (Java)
-  EP1_2 - Média de 3 notas
-  EP1_3 - Cálculo do Delta
-  EP1_4 - Distância entre pontos
-  EP1_5 - Celsius para Fahrenheit
-  EP1_6 - Desconto 10% + 10%
-  EP1_7 - Encriptação
-  EP1_8 - Caminhão
-  Discussões sobre EP1

(3) <https://ava.ufabc.edu.br>

Descrição

Enviar

Editar

Visualizar envios

EP1_1 - Alo, mundo!

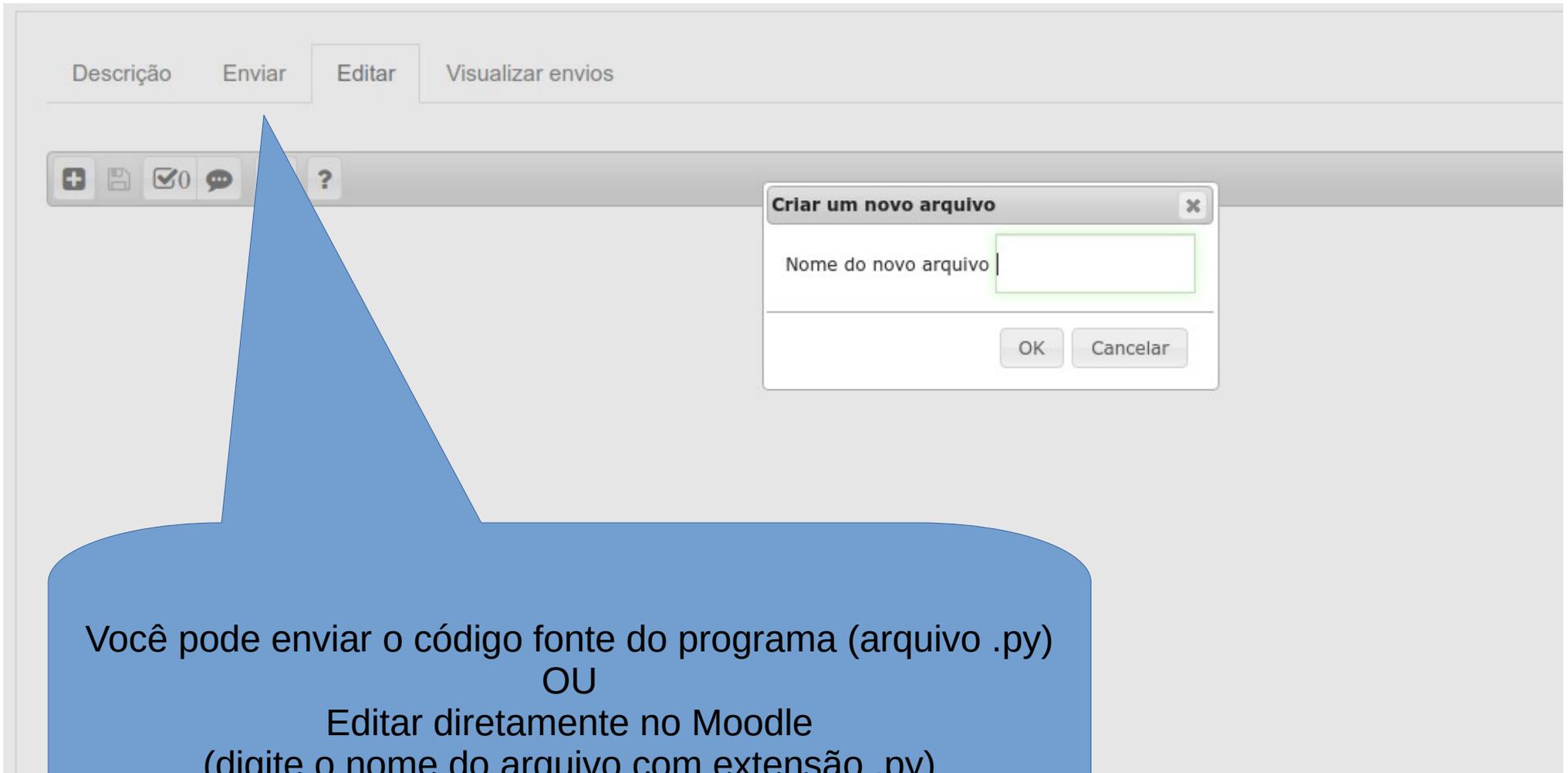
Disponível até: segunda, 2 Mar 2020, 06:00

Número máximo de arquivos: 1

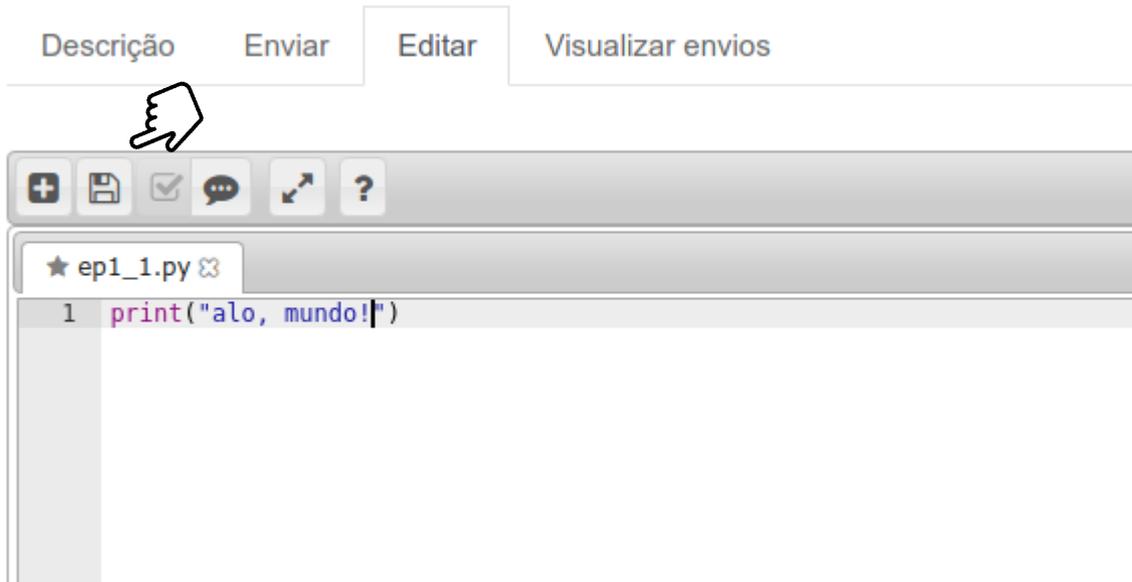
Tipo de trabalho: Trabalho individual

Escreva um programa que mostre a mensagem "Alo, mundo!" (**observe que não há acentuação na mensagem**).

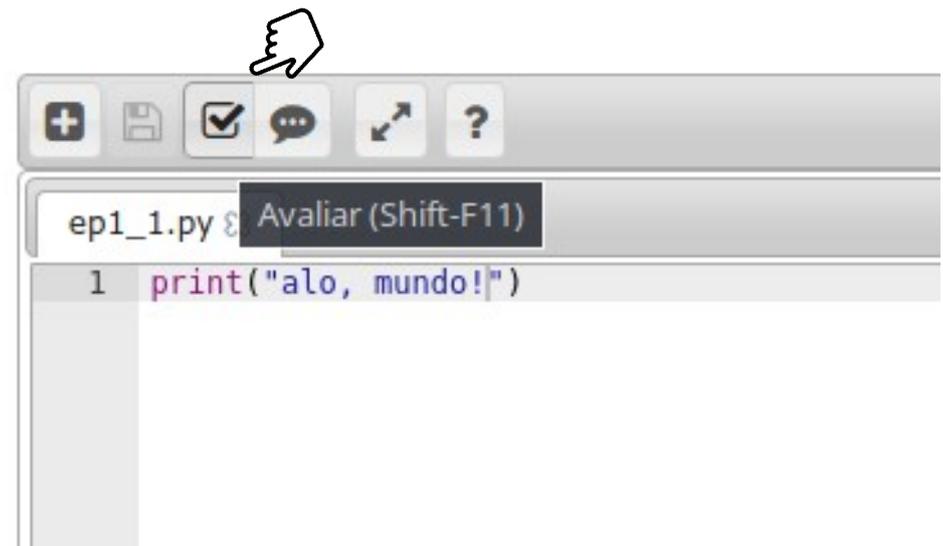
(3) <https://ava.ufabc.edu.br>



(3) <https://ava.ufabc.edu.br>



Salve o arquivo e envie para avaliação
Do sistema



(3) <https://ava.ufabc.edu.br>

Descrição

Enviar

Editar

Visualizar envios

EP1_2 - Média de 3 notas

Disponível até: segunda, 2 Mar 2020, 06:00

Número máximo de arquivos: 1

Tipo de trabalho: Trabalho individual

Escreva um programa que leia 3 notas e calcule a média (aritmética).

Entrada:

- 3 valores (notas)

Saída:

- Média obtida (**com duas casas decimais**)

Fonte: Código 1.14 (cap 1) - livro texto (Neves, R.; Zampirolli, F. Processando a Informação. Editora UFABC, 2017.

(3) <https://ava.ufabc.edu.br>

Veja se o seguinte programa funciona ... funciona?

```
a = float(input())
b = float(input())
c = float(input())

resposta = (a+b+c)/2

print ( round(resposta,2) )

# print (f'{resposta:.2f}')
```