

PPGTER/DES.39.2023.SEQ

Sequência Didática para o Ensino de Lógica de Programação utilizando o Jupyter Notebook

Autores

Mayara Leal Fernandes
mayaralleal@gmail.com

Andre Zanki Cordenonsi
andre.cordenonsi@ufsm.br



Versão 1.0
Status: Final
Distribuição: Externa
Dezembro 2023



2023 PPGTER – Programa de Pós-Graduação em Tecnologias Educacionais em Rede

Atribuição-Não Comercial 4.0 Internacional (CC BY-NC 4.0)

Você tem o direito de compartilhar, copiar e redistribuir o material em qualquer suporte ou formato; adaptar, remixar, transformar, e criar a partir do material, de acordo com o seguinte: você deve dar o crédito apropriado, prover um link para a licença e indicar se mudanças forem feitas. Você deve fazê-lo em qualquer circunstância razoável, mas de nenhuma maneira que sugira que o licenciante apoia você ou seu uso. Você não pode usar o material para fins comerciais.

PPGTER

PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS EDUCACIONAIS EM REDE
UNIVERSIDADE FEDERAL DE SANTA MARIA - CENTRO DE EDUCAÇÃO

Editoria Técnica do PPGTER

Universidade Federal de Santa Maria

Av. Roraima n. 1000

Centro de Educação, Prédio 16, sala 3146

Santa Maria – RS – CEP 97105-900

Fone / FAX: 55 3220 9414

ppgter@ufsm.br

edtec.ppgter@gmail.com

ISSN: 2675-0309

Relatórios Técnicos do Programa de Pós-Graduação em Tecnologias Educacionais em Rede / Programa de Pós-Graduação em Tecnologias Educacionais em Rede, Universidade Federal de Santa Maria. – Vol. 5. n. 1 (2023) – Santa Maria: PPGTER/UFSM, 2021.

Periodicidade anual.

1. Tecnologia Educacional. 2. Desenvolvimento de Tecnologias Educacionais. 3. Gestão de Tecnologias Educacionais. I. Universidade Federal de Santa Maria. Programa de Pós-Graduação em Tecnologias Educacionais em Rede.

Como citar este relatório:

FERNANDES, M.L., CORDENONSI, A. Z. **Sequência Didática para o Ensino de Lógica de Programação utilizando o Jupyter Notebook**. Santa Maria: 2023. Relatórios Técnicos do Programa de Pós-Graduação em Tecnologias Educacionais em Rede, v. 5., n.1. Disponível em: <https://www.ufsm.br/cursos/pos-graduacao/santa-maria/ppgter/ppgter-des-39-2023>

Resumo

Este relatório técnico tem como objetivo apresentar o produto final do trabalho denominado “Utilização dos cadernos digitais para o ensino de lógica de programação”, desenvolvido como parte da dissertação de mestrado que apresentada ao Programa de Pós-graduação em Tecnologias Educacionais em Rede, na linha de pesquisa Desenvolvimento de Tecnologias Educacionais em Rede. O objetivo proposto da dissertação era investigar os desafios e as potencialidades da utilização dos cadernos digitais no ensino e aprendizagem de Lógica de Programação para alunos do curso técnico em Redes de Computadores integrado ao médio do Instituto Federal de Educação, Ciência e Tecnologia do Maranhão - Campus São João dos Patos.

APÊNDICE A

Sequência Didática



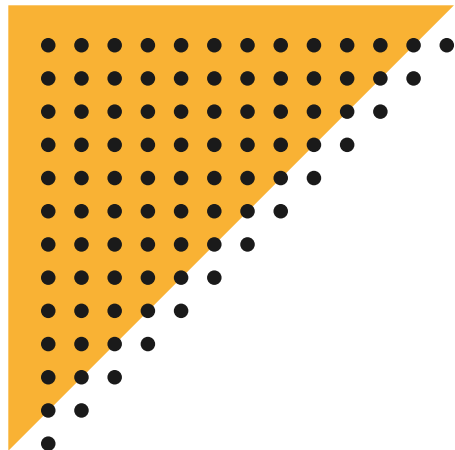
SEQUÊNCIA DIDÁTICA PARA O ENSINO DE LÓGICA DE PROGRAMAÇÃO UTILIZANDO O JUPYTER NOTEBOOK

AUTORA:

Mayara Leal Reis Fernandes

ORIENTADOR:

André Zanki Cordenonsi



SEQUÊNCIA DIDÁTICA PARA O ENSINO DE LÓGICA DE PROGRAMAÇÃO UTILIZANDO O JUPYTER NOTEBOOK

1ª EDIÇÃO

Projeto gráfico e diagramação: Mayara Leal Reis Fernandes

Plataforma de edição: Canva

Material desenvolvido sob a licença
Atribuição-NãoComercial-Compartilhalgual 3.0 Brasil (CC BY-NC-SA 3.0 BR)



<https://creativecommons.org/licenses/by-nc-sa/3.0/br/deed.pt>

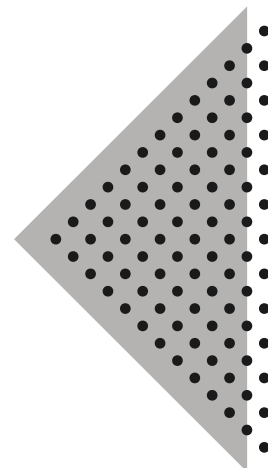
AUTORA:

Mayara Leal Reis Fernandes

ORIENTADOR:

André Zanki Cordenonsi

**SANTA MARIA-RS
2023**





SUMÁRIO

Orientações preliminares	5
Anaconda	5
Produto Educacional	6
Síntese da aplicação da sequência didática.....	6
D0 - Primeiro encontro	7
D0.1 - Primeiro momento da oficina	7
D0.2 - Segundo momento da oficina	8
D0.3 - Terceiro momento da oficina	9
D1 - Segundo encontro	11
D1.1 - Primeiro momento da oficina	11
D1.2 - Segundo momento da oficina	12
D0 - Terceiro encontro	14
D3.1 - Primeiro momento da oficina	14
D3.2 - Segundo momento da oficina	15
D3.3 - Terceiro momento da oficina	16
Referências	17
Apêndice A - Tutorial prático de instalação do Anaconda no Windows	18
Apêndice B - Questionário inicial de identificação e caracterização dos sujeitos da pesquisa	25
Apêndice C - Caderno digital: Variáveis, constantes e operadores aritmético....	33
Apêndice D - Caderno digital: Estrutura sequencial	37
Apêndice E - Caderno digital: Estrutura sequencial (exercícios)	40
Apêndice F - Caderno digital: Estruturas de decisão	41
Apêndice G - Caderno digital: Estruturas de decisão (exercícios)	44
Apêndice H - Questionário avaliação do produto educacional.....	45



APRESENTAÇÃO

O acesso universal a informação proporcionado pelo advento da Internet e das mídias digitais, transformou a sociedade e, com ela, a forma de viver, consumir, trabalhar e, sobretudo, aprender. As novas Tecnologias da Informação e da Comunicação (TIC) têm sido investigadas como formas inovadoras de apoiar o processo de ensino e aprendizagem em escolas, universidades, empresas de programas de formação, entre tantos outros campos.

Inserido nesse contexto, este produto educacional que está vinculado à linha de pesquisa Gestão de Tecnologias Educacionais em Rede, do Programa de Pós-Graduação em Tecnologias Educacionais em Rede (PPGTER), da Universidade Federal de Santa Maria (UFSM), foi construído a partir da dissertação de mestrado intitulada Utilização dos Cadernos Digitais para o Ensino de Lógica de Programação que conta com a concepção, fundamentação, estruturação e elaboração de uma Sequência Didática com conteúdos teóricos e práticos sobre Lógica de Programação.

Esse produto visa contribuir com o campo da educação, no intuito de explorar os potenciais de uma ferramenta tecnológica, intitulada Jupyter Notebook, através dos cadernos digitais, para desenvolver e compartilhar materiais educacionais, sobre o currículo Lógica de Programação, especificamente os conceitos introdutórios, conteúdos e exercícios de estrutura sequencial e estruturas de decisão, combinando diferentes tipos de recursos como texto, imagens e código em linguagem de programação em um único documento, acessível até mesmo por meio de um navegador web, buscando potencializar o processo de aprendizagem e gerar conhecimento, como uma alternativa interativa, dinâmica e motivacional.



Orientações preliminares

Os cadernos digitais podem ser desenvolvidos e acessados localmente por meio da instalação de uma IDE (*Integrated Development Environment*), Ambiente de Desenvolvimento Integrado, um programa de computador utilizado por desenvolvedores de aplicações cujo o objetivo é facilitar o processo através interface gráfica do usuário (*Graphical User Interface* (GUI), em inglês), com suporte para linguagem Python. Nesta sequência didática, é utilizada a interface gráfica **Anaconda**. Trata-se de uma distribuição de software gratuito e disponível para vários sistemas operacionais, dentre eles o Windows, Linux e Mac OS X.



Além dessa, existe outra alternativa, onde qualquer navegador pode ser empregado para implementá-los e apreciá-los, sem necessidade de instalação, facilmente realizado acessando o link <http://try.jupyter.org>. Assim sendo, o usuário pode abrir, escrever e executar as células dos cadernos digitais sem instalar o Python ou qualquer outra biblioteca em um computador pessoal.

Vale ressaltar, ainda, que os usuários podem utilizar outras ferramentas em nuvem que permitam criar e executar códigos na linguagem Python, inclusive a própria versão do Anaconda.

Anaconda

O Anaconda é uma plataforma de ciência de dados poderosa e fácil de usar. Ele permite que os usuários adicionem e gerenciem facilmente mais de 1.500 pacotes pré-instalados, sem precisar lidar com problemas de compatibilidade ou conflitos entre pacotes. Além disso, inclui o Jupyter Notebook, uma ferramenta popular para a criação e compartilhamento de documentos interativos que combinam código, texto, equações e visualizações. Para mais informações de instalação e utilização da ferramenta, acesse o Apêndice A.



Produto Educacional

Tipo do produto: Proposta de ensino (Sequência didática).

Tempo estimado para aplicação da sequência didática: 3 aulas de 50 minutos cada, divididas em 3 períodos/dias diferentes.

Público alvo: Estudantes da Educação Básica, Técnica e Tecnológica.

Nível de escolaridade: Estudantes do 1º ano do ensino médio integrado ao curso técnico.

Requisito mínimo: Conhecimento de Informática Básica e acesso ao computador.

Quem pode aplicar: Preferencialmente professor(a) com habilitação na área de computação.

Objetivo da sequência didática: Desenvolver o ensino de Lógica de Programação por meio da execução das atividades organizadas em uma Sequência Didática mediante o uso de uma plataforma computacional intitulada Jupyter Notebook.

Síntese da aplicação da sequência didática

Esta Sequência Didática (SD) é apresentada para aplicação em três encontros com duração de 3 horas/aulas, totalizando 9h/a. Ao término de cada sessão é desenvolvida uma atividade prática no computador, com o uso do Jupyter Notebook, que pode ser de forma individual ou em grupo, dependendo da disponibilidade de recursos do professor, quantidade de alunos, etc. As demais atividades propostas no decorrer dos cadernos digitais são de caráter factual para auxiliar na construção do conhecimento sobre lógica de programação e podem ser desenvolvidas de forma individual ou coletiva. Para iniciar, leia com atenção as orientações que constam neste material.



D0 - Primeiro encontro

Duração: 3 aulas.

Objetivos: Apresentar e discutir a realização do projeto. Aplicar questionário de identificação e caracterização dos estudantes. Explorar os principais comandos do Jupyter Notebook com a utilização e manipulação dos cadernos digitais por meio dos elementos de desenvolvimento de algoritmos, como tipos de dados, variáveis, constantes e operadores.

Infraestrutura: Computador.

Softwares: Anaconda/Jupyter Notebook.

D0.1 - Primeiro momento da oficina



Orientações ao professor

A oficina é iniciada com a apresentação e discursão sobre realização do projeto, suas etapas, bem como os objetivos, as atividades e os dias e os horários da oficina que integram a proposta do estudo.



Desenvolvimento

O professor pode compartilhar o quadro a seguir através do projetor de mídia ou somente dialogar com os estudantes sobre a execução da sequência didática. A coluna **data/horário** deve ser preenchida de acordo com o planejamento do professor(a) contemplando 3 horas/aulas para cada encontro.



Encontro	Data/Horário	Objetivos
0	____/____/____ Das ____:____ às ____:____	Apresentar e discutir a realização do projeto. Aplicar questionário de identificação e caracterização dos estudantes. Explorar os principais comandos do Jupyter Notebook com a utilização e manipulação dos cadernos digitais por meio dos elementos de desenvolvimento de algoritmos, como tipos de dados, variáveis, constantes e operadores.
1	____/____/____ Das ____:____ às ____:____	Estudar sobre estrutura de programação sequencial utilizando os cadernos digitais através da plataforma computacional Jupyter Notebook.
2	____/____/____ Das ____:____ às ____:____	Conhecer e desenvolver algoritmos para controle de fluxo de dados, com o uso de estruturas de decisão, por meio dos cadernos digitais, através da plataforma computacional Jupyter Notebook.

D0.2 - Segundo momento da oficina



Orientações ao professor

Neste momento da aula, o professor socializa o formulário impresso ou online para os estudantes responderem questionário inicial (Apêndice B).



O questionário inicial apresentado está disponível na íntegra virtualmente no link:

<https://forms.gle/myN587bvtn19YmD27>



Desenvolvimento

O professor pode realizar a leitura do questionário de identificação e caracterização que possui 26 questões divididas em 4 blocos: perfil dos participantes, conhecimento quanto às ferramentas tecnológicas educacionais, lógica e linguagem de programação. Recomenda-se que, no decorrer da leitura do professor, sejam registradas as respostas dos alunos.

D0.3 - Terceiro momento da oficina



Orientações ao professor

Neste momento a aula é expositiva. O professor explica sobre a introdução da ferramenta Jupyter Notebook, como realizar a utilização e manipulação dos cadernos digitais e os estudantes irão interagir, conhecer a interface e acompanhar o passo-a-passo, juntamente com a explanação. Recomenda-se realizar esta etapa em um laboratório de informática, podendo ser executada de forma individual, em dupla ou grupo, dependendo dos recursos do professor.



Desenvolvimento

Inicialmente, o professor orienta os alunos para a instalação do software (Anexo A). Em seguida, o professor compartilha o arquivo contendo o caderno digital referente a Introdução à Lógica de Programação.



O professor poderá exibir o vídeo de curta duração explicando o funcionamento do Jupyter Notebook.
Sugestão: <https://www.youtube.com/watch?v=dPb4acFiaYs>



Se o professor preferir, pode solicitar ao responsável pelo laboratório da escola que faça a instalação do Jupyter Notebook em todos os computadores.



Os arquivos dos cadernos digitais podem ser compartilhados em uma plataforma de hospedagem de código-fonte ou em um sistema de gerenciamento de conteúdo da escola.



Aula expositiva

Neste momento da aula, os estudantes acompanham o conteúdo através do caderno digital referente ao **Módulo 0 - Variáveis, constantes e operadores aritméticos** (Apêndice D), executado no Jupyter Notebook. O professor apresenta os principais comandos do Jupyter Notebook, com a utilização e manipulação dos cadernos digitais, por meio dos elementos de desenvolvimento de algoritmos, como tipos de dados, variáveis, constantes e operadores.



O caderno digital apresentado está disponível na íntegra virtualmente no link:

<https://github.com/mayaralleal/notebooks/blob/main/Intro.ipynb>



No decorrer da apresentação dos conceitos de introdutórios de algoritmos, o professor propicia o diálogo e a interação com os estudantes e realiza a resolução dos problemas propostos.

Atividade proposta: Criar algoritmos para cálculos matemáticos.

Objetivos da atividade prática: Desenvolver programas com processamento de dados utilizando os operadores aritméticos, relacionais e lógicos, além de funções matemáticas pré-definidas.



D1 - Segundo encontro

Duração: 3 aulas.

Objetivo: Estudar a estrutura sequencial de programação, utilizando os cadernos digitais, através da plataforma computacional Jupyter Notebook.

Infraestrutura: Computador.

Softwares: Anaconda/Jupyter Notebook.

D1.1 - Primeiro momento da oficina



Orientações ao professor

Neste momento, a aula é expositiva. O professor apresenta ao aluno a estrutura sequencial e ele será capaz de entender e elaborar algoritmos utilizando essa estrutura. Recomenda-se realizar essa etapa em um laboratório de informática, podendo ser executada de forma individual, em dupla ou grupo, dependendo dos recursos do professor.



Desenvolvimento

O professor orienta os estudantes à acessarem o repositório onde estão hospedados os cadernos digitais. Posteriormente, realizam o download do material **Módulo 1 - Estrutura sequencial** (Apêndice E) e, em seguida, abrem o arquivo utilizando a ferramenta Jupyter Notebook.



Aula expositiva

Neste momento da aula os estudantes acompanham o conteúdo através do caderno digital referente ao módulo da aula executado no Jupyter Notebook. São abordados os programas em Python com entrada, processamento e saída de resultados com uma estrutura sequencial. No decorrer da explicação dos conteúdos, além de exibir um vídeo de curta duração, proposto no caderno digital, o professor propicia o diálogo e a interação com os estudantes e realiza a resolução dos problemas propostos.



O caderno digital **Módulo 1 - Estrutura Sequencial** está disponível na íntegra virtualmente no link:

<https://github.com/mayaralleal/notebooks/blob/main/EstrutSeq.ipynb>



D1.2 - Segundo momento da aula



Orientações ao professor

Os alunos desenvolvem a atividade proposta com a orientação do professor. A atividade possui exercícios cujo grau de dificuldade aumenta a cada questão. Quanto mais próximo do final da atividade o aluno chegar, maior é a sua apropriação do conteúdo proposto.



Desenvolvimento

O professor orienta os estudantes a acessar o repositório onde estão hospedados os cadernos digitais. Posteriormente, realizaram o download do material **Exercícios - Estrutura Sequencial** (Apêndice F) e, em seguida, abrem o arquivo utilizando a ferramenta Jupyter Notebook e os instrui a resolver as questões.



O caderno digital **Exercícios - Estrutura Sequencial** está disponível na íntegra virtualmente no link:

<https://github.com/mayaralleal/notebooks/blob/main/ExerEstrutSeq.ipynb>



Atividade proposta: Trabalhar estrutura sequencial.

Objetivos da atividade prática: Desenvolver programas em Linguagem Python com atividades do cotidiano e treinar o conhecimento em estrutura de sequencial.



D3 - Terceiro encontro

Duração: 3 aulas.

Objetivos: Conhecer e desenvolver algoritmos para controle de fluxo de dados, com o uso de estrutura de decisão, por meio dos cadernos digitais, através da plataforma computacional Jupyter Notebook.

Infraestrutura: Computador.

Softwares: Anaconda/Jupyter Notebook.

D3.1 - Primeiro momento da oficina



Orientações ao professor

Neste momento, a aula é expositiva. O professor aborda conceitos pertinentes ao conteúdo de estruturas de decisão simples, composta e aninhada. Recomenda-se realizar esta etapa em um laboratório de informática, podendo ser executada de forma individual, em dupla ou grupo, dependendo dos recursos do professor.



Desenvolvimento

O professor orienta os estudantes à acessarem o repositório onde estão hospedados os cadernos digitais. Posteriormente, realizam o download do material **Módulo 2 - Estrutura de decisão** (Apêndice G) e, em seguida, abrem o arquivo utilizando a ferramenta Jupyter Notebook.



Aula expositiva

Neste momento da aula, os estudantes acompanham o conteúdo através do caderno digital referente ao módulo da aula executado no Jupyter Notebook. São abordados definições para construir algoritmos com estruturas de decisão. Os estudantes vão aprender como os algoritmos tomam decisões e selecionam os blocos de comandos para serem executados de acordo com as condições estabelecidas.



O caderno digital **Módulo 2 - Estrutura de decisão** está disponível na íntegra virtualmente no link:

<https://github.com/mayaralleal/notebooks/blob/main/EstrutDec.ipynb>



D3.2 - Segundo momento da oficina



Orientações ao professor

Os alunos desenvolvem a atividade proposta com a orientação do professor. A atividade possui exercícios cujo grau de dificuldade aumenta a cada questão. Quanto mais próximo do final da atividade o aluno chegar, maior é a sua apropriação do conteúdo proposto.



Desenvolvimento

O professor orienta os estudantes à acessarem o repositório onde estão hospedados os cadernos digitais, posteriormente, realizar o download do material **Exercícios - Estrutura de decisão If/Else** (Apêndice H) e, em seguida, abrem o arquivo utilizando a ferramenta Jupyter Notebook e os instrui a resolver as questões.



O caderno digital **Exercícios - Estrutura de decisão If/Else** está disponível na íntegra virtualmente no link:

<https://github.com/mayaralleal/notebooks/blob/main/ExerEstrutDec.ipynb>



Atividade proposta: Trabalhar estruturas de decisão.

Objetivos da atividade prática: Treinar o conhecimento em estrutura de decisão decisão - If/Else, onde o fluxo de execução dos comandos do algoritmo é determinado pelos dados conforme o andamento de determinadas condições.

D3.3 - Terceiro momento da oficina



Orientações ao professor

Neste momento da aula, o professor comenta detalhadamente sobre disponibilização do questionário de avaliação do produto (Apêndice C) que será encaminhado através de e-mail para os participantes da pesquisa.



O questionário final apresentado está disponível na íntegra virtualmente no link:

<https://forms.gle/LGz9iUhJX6x2HJX69>



Desenvolvimento

O professor pode realizar a leitura do formulário com 14 questões, cuja intenção é verificar se utilização dos cadernos digitais contribuem, motivaram e prepararam os alunos para a aprendizagem dos currículos estrutura sequencial e estrutura de decisão da disciplina de lógica de programação. Recomenda-se que sejam registradas as respostas dos alunos em até 7 dias após a finalização da oficina.



Referências

BENEDUZZI, Humberto Martins; METZ, João Ariberto. Lógica e Linguagem de Programação: introdução ao desenvolvimento de software. Curitiba: Editora do Livro Técnico, 2010.

FERRI, Juliana. Ensino de Linguagem de Programação na Educação Básica: Uma Proposta de Sequência Didática para Desenvolver o Pensamento Computacional. Produção Técnica Educacional apresentada ao Programa de Pós-Graduação em Ensino da Universidade Estadual do Norte do Paraná – Campus Cornélio Procópio (2017). Disponível em: <https://produtoeducacionalsequenciadidaticapensamentocomputacional.files.wordpress.com/2017/07/pensando-igual-ao-computador-sd.pdf>. Acesso em: 22 jul. 2023.

INTROCOMP. Um projeto de extensão sem fins lucrativos da Universidade Federal do Espírito Santo. Disponível em: <https://introcomp.pet.inf.ufes.br>. Acesso em: 22 jul. 2023.

Jupyter. 2022. Disponível em: <https://jupyter.org/>. Acesso em: 22 jul. 2022.

LACERDA, Liluyoud Cury de; RAMOS, José Marcio Benite; DUARTE, Sara Luize Oliveira. Lógica de programação – Cuiabá: Ed.UFMT, 2014.

MALBOUISSON, Helena; DAMIÃO, Dilson de Jesus; MORA, Clemencia. Lishep Sessão A. Cadernos Eletrônicos Digitais. Universidade do Estado do Rio de Janeiro. Disponível em: https://colab.research.google.com/drive/1kZDxACinkG9WI_ctVytZFCHgmz5y2rYs?usp=sharing. Acesso em: 22 jul. 2023.

MANZANO, José Augusto Navarro Garcia. Algoritmos: lógica para desenvolvimento de programação de computadores. 27ª Edição, Revisada. São Paulo: Érica, 2014.

SOUZA, Marco A. Furlan D.; GOMES, Marcelo M.; SOARES, Marcio V.; CONCILIO, Ricardo. Algoritmos e lógica de programação: um texto introdutório para a engenharia. Cengage Learning Brasil, 2019.

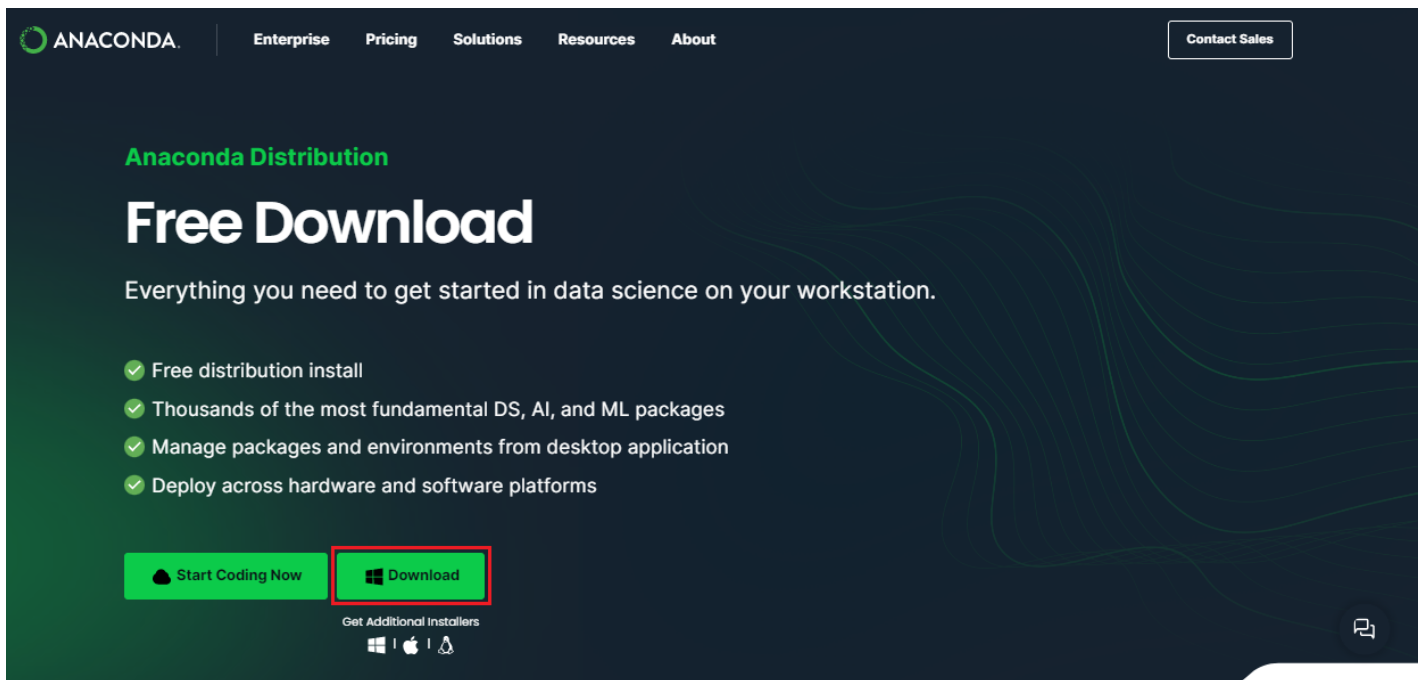
VASCONCELLOS, Isadora Lopes Barbosa. Apostila de Lógica de Programação. Produto do Mestrado Profissional em Ensino e suas Tecnologias. Instituto Federal de Educação, Ciência e Tecnologia Fluminense (IFFluminense) – Campus Campos Centro (2019). Disponível em: <http://www.uel.br/laboratorios/labsan/arquivosEliandro/Apostila-CursodeLogicadeProgramacao.pdf>. Acesso em 19 de jul. de 2023.

WILLE, Patrícia Bonow Fassbender. Sequência Didática: Encontros Formativos/Investigativos – As Expectativas Formativas dos Estudantes do Curso Normal. Produto do Mestrado em Ciências e Tecnologias na Educação. Instituto Federal Sul-Rio-Grandense – Campus Pelotas-Visconde da Graça (2020). Disponível em: https://proedu.rnp.br/bitstream/handle/123456789/1697/Produto_Educacional_Patricia_Bonow_Fassbender_Wille.pdf?sequence=1&isAllowed=y. Acesso em 19 de jul. de 2023.

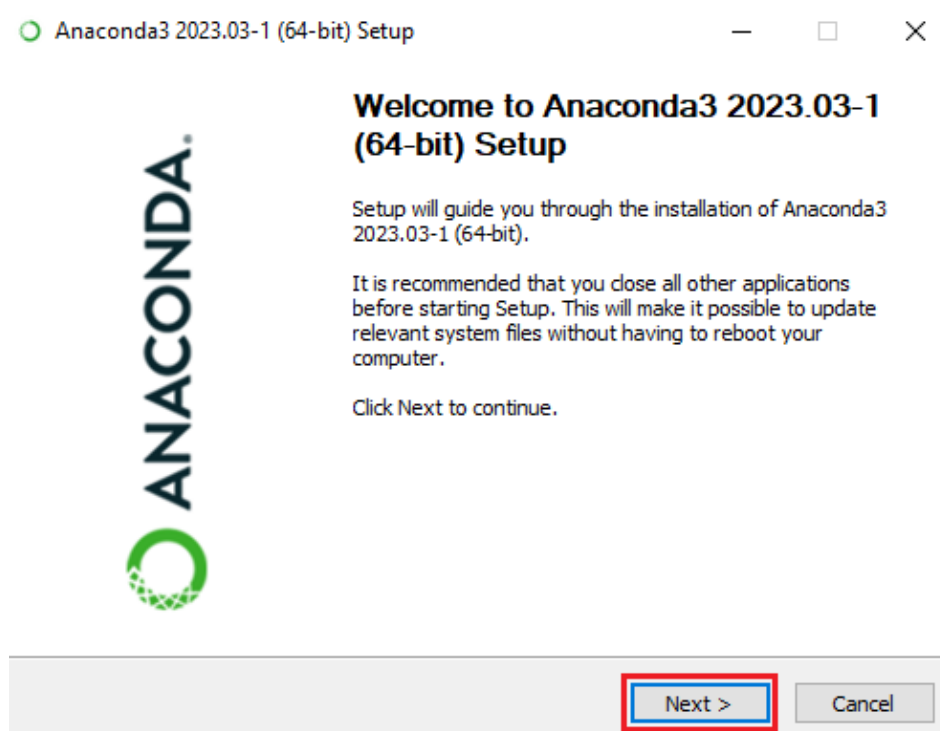


Apêndice A - Tutorial prático de instalação do Anaconda no Windows

1. Baixe o instalador do Anaconda no próprio site da distribuidora.
2. <https://www.anaconda.com/download>
3. Clique em **Download** para baixar e duas vezes no instalador para iniciar.

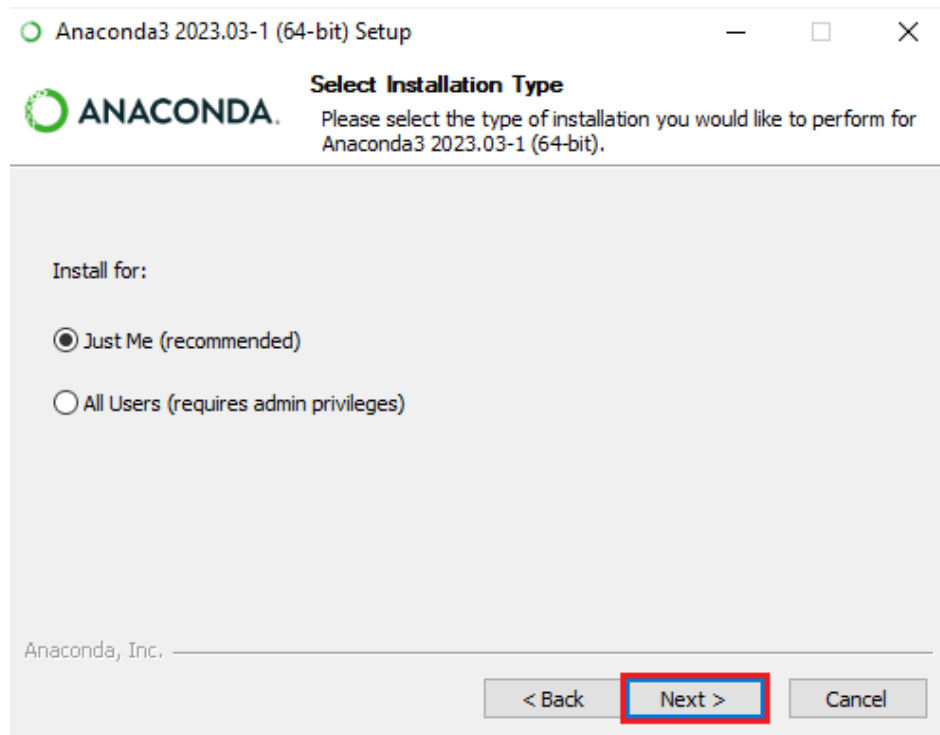


4. Clique em **Next** para iniciar a instalação.

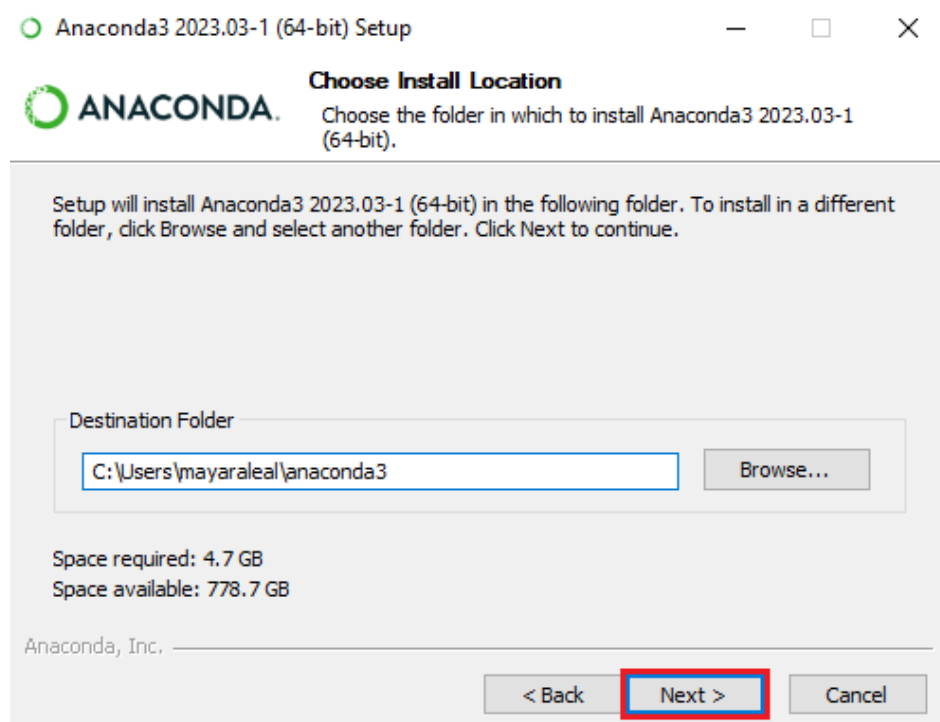




5. Leia os termos de licenciamento e clique em **I Agree** (Concordo).
6. Selecione uma instalação para o **Just Me** (Apenas eu), a menos que você esteja instalando para todos os usuários (o que requer privilégios de administrador do Windows) e clique em **Next**.

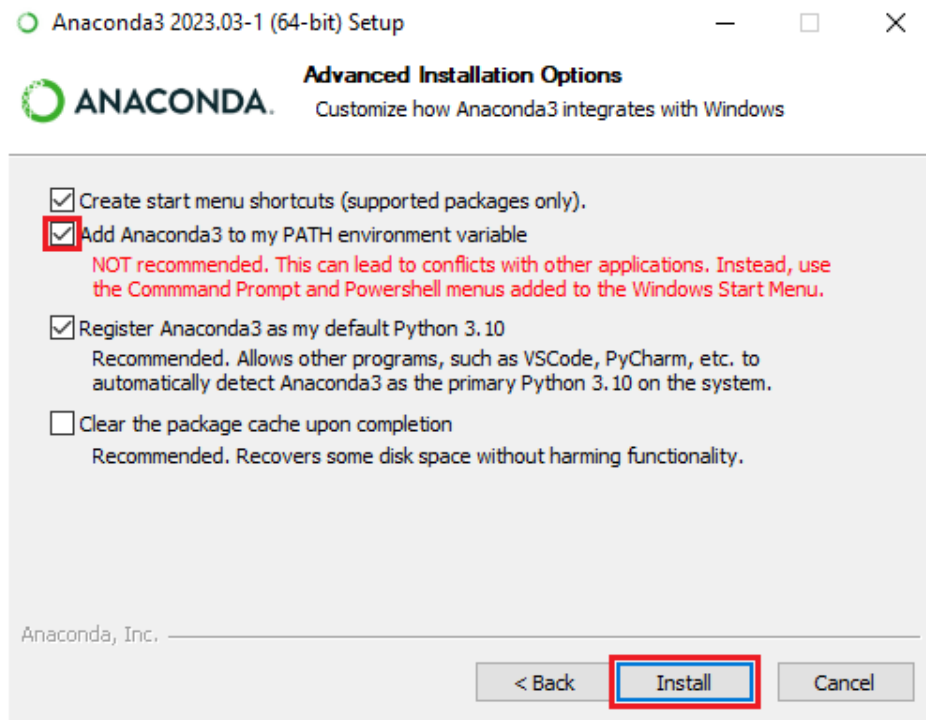


7. Selecione uma pasta de destino para instalar o Anaconda e clique no botão **Next**.

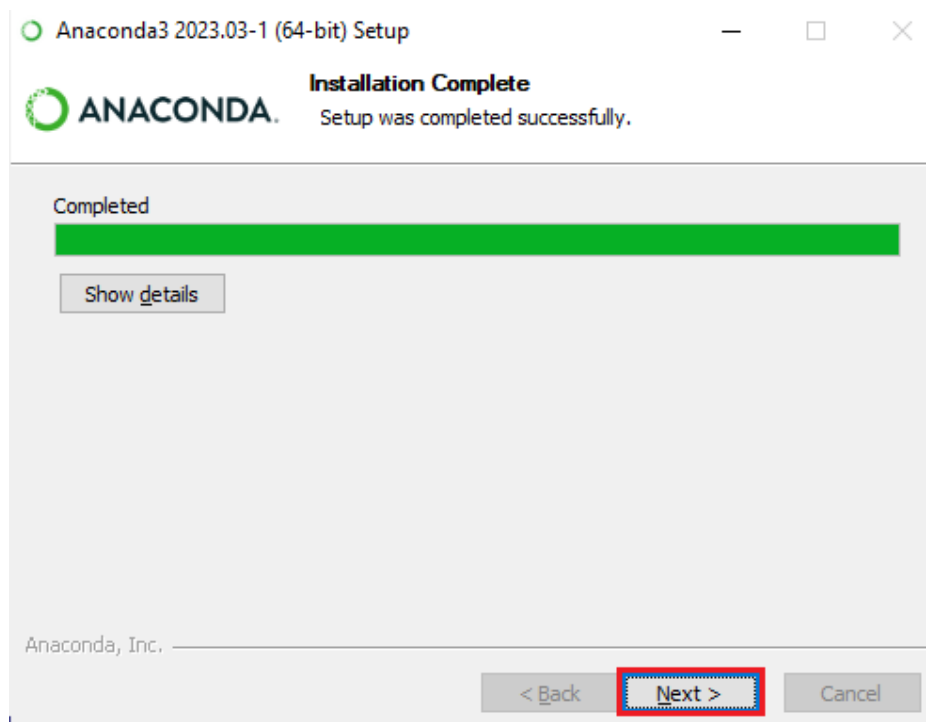




8. Marque a opção **Add Anaconda3 to my PATH environment variable** (Adicionar Anaconda à minha variável de ambiente PATH) durante a instalação e, em seguida, clique em **Install**.

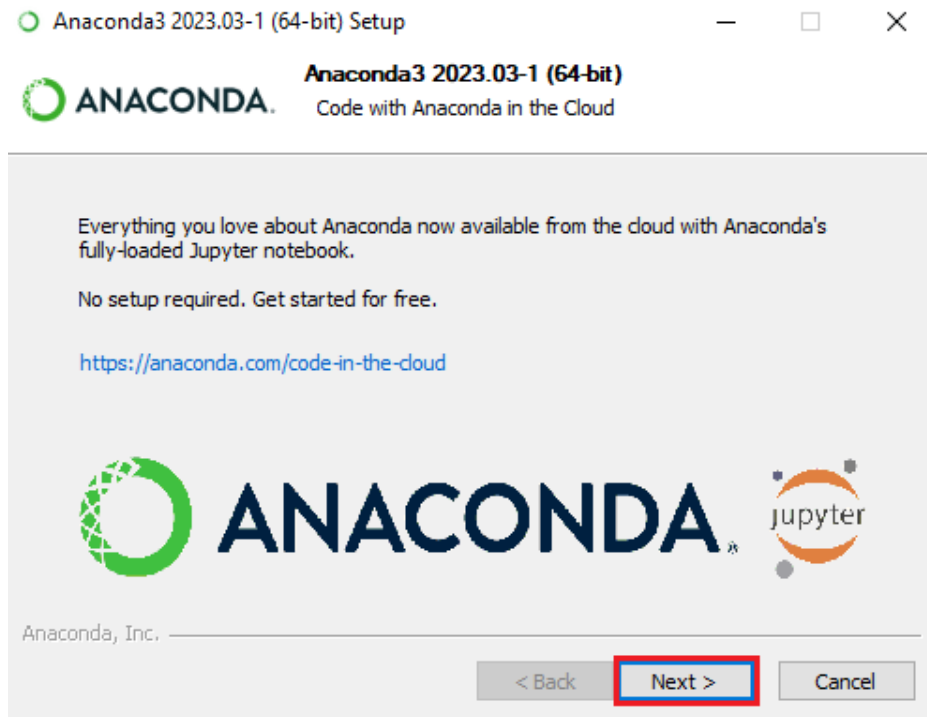


9. Após a configuração ser concluída com sucesso, clique em **Next**.

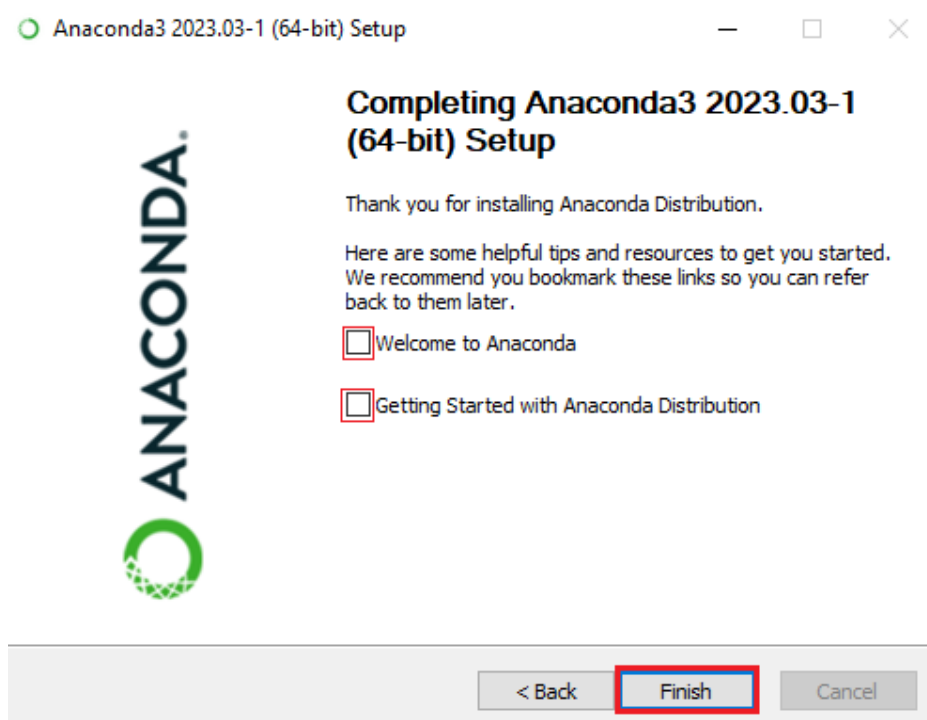




10. O Anaconda está disponível na nuvem e pode ser acessado mediante cadastro através de e-mail e senha pelo site: <https://www.anaconda.com/code-in-the-cloud>). Clique em **Next**.

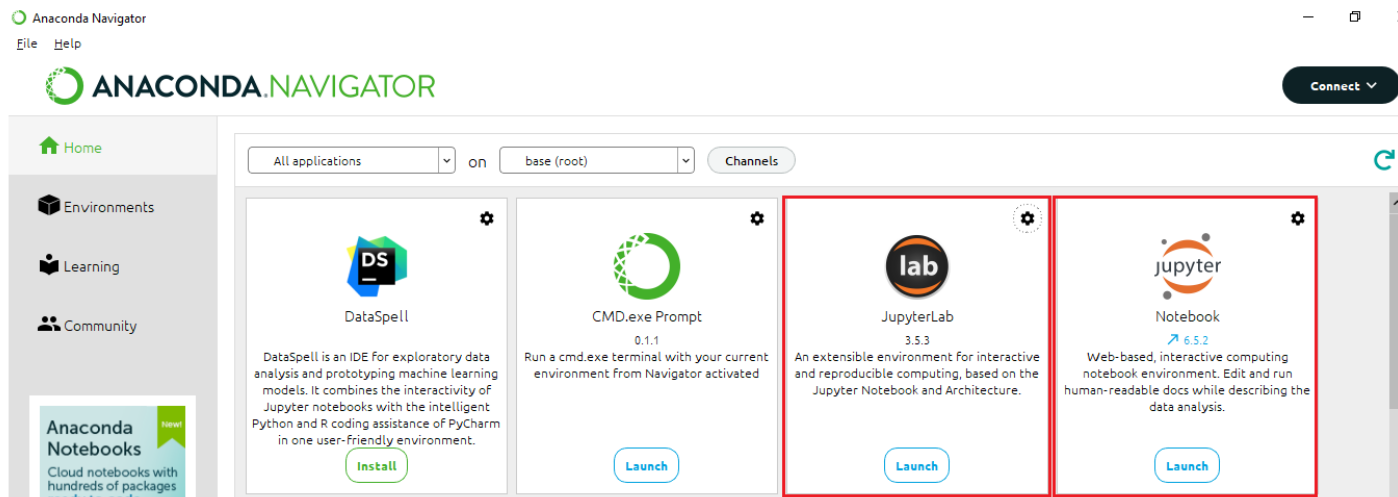


11. Na parte final após a instalação desmarque as duas caixas para que não abra o tutorial e o arquivo de ajuda.



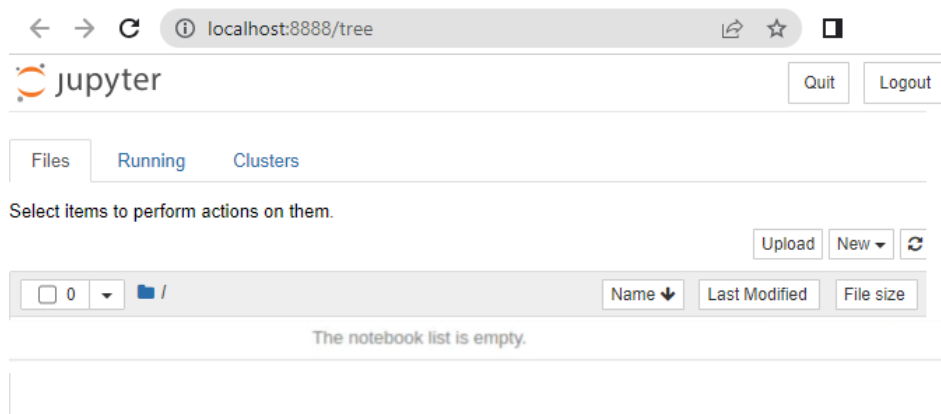


12. Ao executar o Anaconda abrirá o painel de navegação para iniciar o Jupyter Notebook ou Jupyter Lab, uma nova geração da interface web para o projeto Jupyter.



Como funciona o Jupyter Notebook?

Ao executar o Jupyter Notebook aparecerá a página inicial do seu caderno digital, conforme imagem abaixo. Ela é separada em 3 abas: Files, Running e Clusters. A aba default é a Files, é nela que visualiza, abre e cria cadernos digitais.

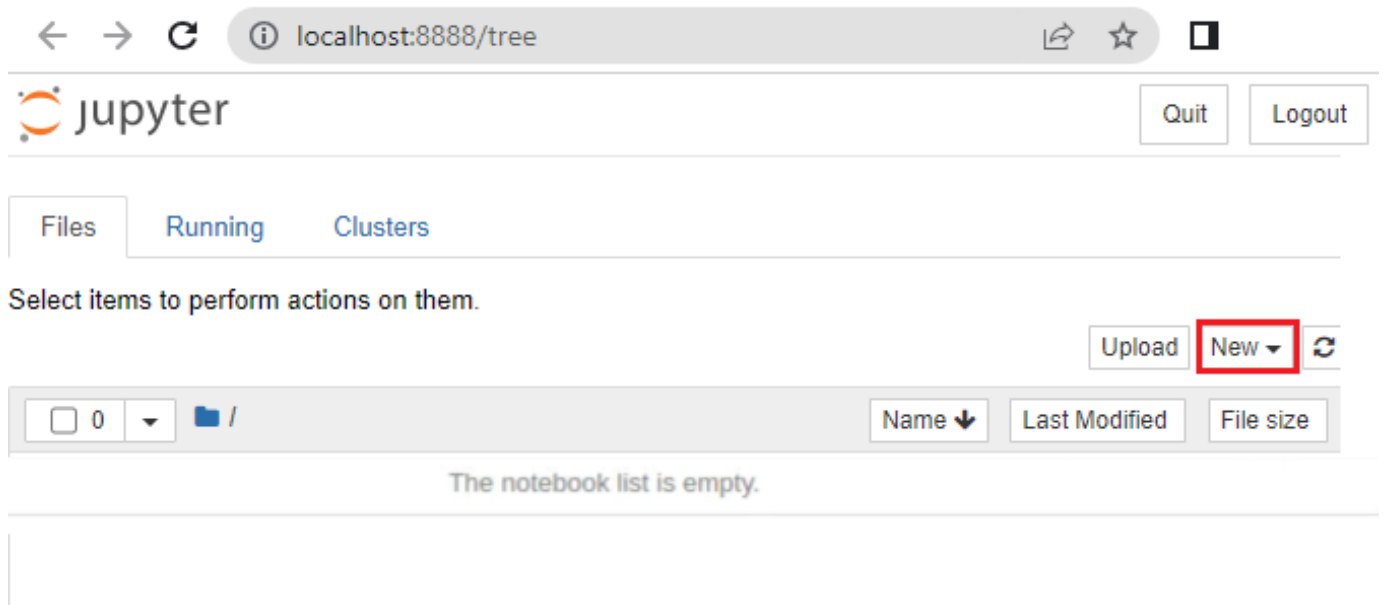


Com essa página, além de servir como um navegador simples de diretórios, poderá abrir cadernos digitais já existentes, criar novos cadernos digitais, abrir terminais do sistema, criar e deletar diretórios, e criar e editar arquivos de texto.

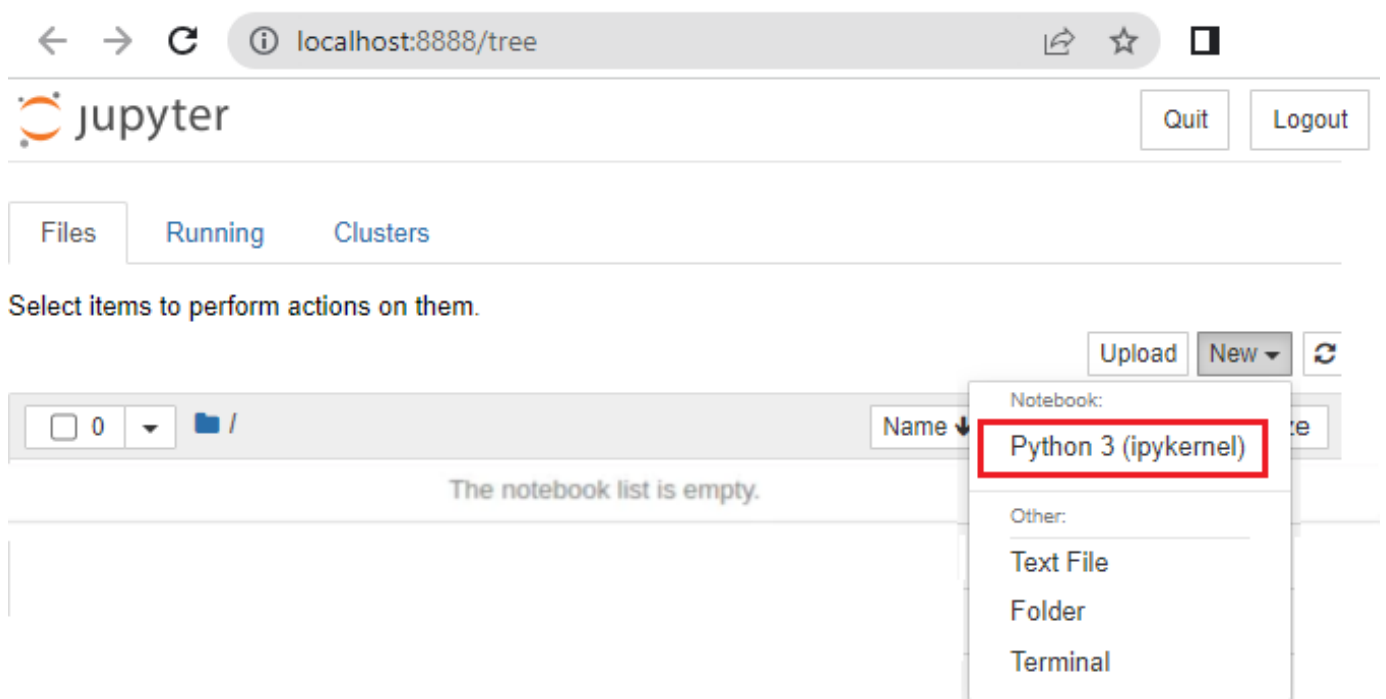


Abrindo ou editando um notebook

Para criar um caderno digital selecione o botão **new** no canto superior esquerdo da página:

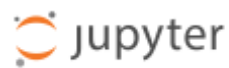


O Jupyter dará a opção de criar um caderno digital cuja execução será suportada pelo kernel de uma linguagem, no caso da figura abaixo o Python 3:





Para abrir um caderno digital já existente, basta procurar por um arquivo do tipo .ipynb (de Ipython Notebook) com o auxílio da pasta inicial do Jupyter:

[Quit](#)[Logout](#)[Files](#)[Running](#)[Clusters](#)

Select items to perform actions on them.

[Upload](#)[New ▾](#)☐ 0 ▾[/ Documents / PPGTER / Dissertação / Notebooks jupyter](#)[Name ▾](#)[Last Modified](#)[File size](#)

..

há poucos segundos

 [Estrutura Condicional.ipynb](#)

Running há 9 meses

609 B

 [Estrutura Sequencial .ipynb](#)

Running há 9 meses

1.66 kB

 [Exercícios Estrutura Condicional.ipynb](#)

Running há 9 meses

589 B

 [Exercícios Estrutura Sequencial.ipynb](#)

Running há 9 meses

589 B



Apêndice B - Questionário inicial de identificação e caracterização dos sujeitos da pesquisa

Questionário Inicial

Este questionário é parte integrante do Produto Educacional SEQUÊNCIA DIDÁTICA PARA O ENSINO DE LÓGICA DE PROGRAMAÇÃO UTILIZANDO O JUPYTER NOTEBOOK, apresentado ao Programa de Pós-Graduação em Tecnologias Educacionais em Rede (PPGTER), da Universidade Federal de Santa Maria (UFSM, RS). Desenvolvido por Mayara Leal Reis Fernandes sob a orientação do Prof. Dr. André Zanki Cordenonsi.

** Indica uma pergunta obrigatória*

1. Você aceita participar da pesquisa? *

Marcar apenas uma oval.

☐ Sim

☐ Não

BLOCO 1 - DADOS GERAIS

2. Nome completo *

3. E-mail *

4. Idade *

Apenas números.

5. Telefone celular *

Preferencialmente, Whatsapp.



6. Em que ano você começou a estudar no IFMA? *

Marcar apenas uma oval.

- ☐ 2019 ou antes
☐ 2020
☐ 2021
☐ 2022

7. Qual a sua forma de ingresso no IFMA? *

Marcar apenas uma oval.

- ☐ Ampla concorrência
☐ Cotas
☐ Outro: _____

8. Qual a sua turma atualmente? *

Marcar apenas uma oval.

- ☐ Técnico em Redes de Computadores Integrado ao Médio 1º ano (tarde)
☐ Técnico em Redes de Computadores Integrado ao Médio 2º ano (manhã)
☐ Técnico em Redes de Computadores Integrado ao Médio 3º ano (manhã)
☐ Outro: _____

9. Possui algum tipo de deficiência? *

Marcar apenas uma oval.

- ☐ Sim
☐ Não *Pular para a pergunta 11*

Portador(a) de deficiência



10. Qual a sua deficiência? *

BLOCO 2 - CARACTERIZAÇÃO SOCIOECONÔMICA

11. Contando com você, quantas pessoas moram no seu domicílio? *

12. Qual a renda mensal de toda a sua família? *

(Some a renda de todas as pessoas do domicílio onde você mora, incluindo a sua, se houver).

Marcar apenas uma oval.

- ☐ Nenhuma renda
- ☐ Até meio salário mínimo (até R\$ 606,00)
- ☐ De meio a 1 salário mínimo (de R\$ 606,01 a R\$ 1.212,00)
- ☐ De 1 a 2 salários mínimos (de R\$ 1.212,01 a R\$ 2.424,00)
- ☐ De 2 a 3 salários mínimos (de R\$ 2.424,01 a R\$ 3.636,00)
- ☐ De 3 a 5 salários mínimos (de R\$ 3.636,01 a R\$ 6.060,00)
- ☐ De 5 a 10 salários mínimos (de R\$ 6.060,01 a R\$ 12.120,00)
- ☐ De 10 a 20 salários mínimos (de R\$ 12.120,01 a R\$ 24.240,00)
- ☐ Mais de 20 salários mínimos (mais de R\$ 24.240,00)

13. Considerando apenas o tempo antes de ingressar no IFMA, em que tipo de escola você * estudou na maior parte da sua vida?

Marcar apenas uma oval.

- ☐ Escola pública
- ☐ Escola privada (particular)

BLOCO 3 (Parte 1) - CONDIÇÕES TECNOLÓGICAS E PEDAGÓGICAS DE ACESSO A CONTEÚDO DIGITAL



14. Dos itens listados abaixo, quais você possui? *

Considere apenas os aparelhos que pertencem a você, excluindo aqueles que pertencem a outras pessoas do seu domicílio. Marque apenas uma resposta para cada item.

Marcar apenas uma oval por linha.

	SIM	NÃO
Computador de mesa (Desktop)	<input type="radio"/>	<input type="radio"/>
Tablet	<input type="radio"/>	<input type="radio"/>
Notebook	<input type="radio"/>	<input type="radio"/>
Smartphone	<input type="radio"/>	<input type="radio"/>

15. Com qual frequência você utiliza os itens listados abaixo? *

Marcar apenas uma oval por linha.

	Não uso	Uso até 2h/dia	Uso entre 2h/dia e 4h/dia	Uso 5h ou mais por dia
Computador de mesa (Desktop)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tablet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Notebook	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Smartphone	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

16. Seu domicílio tem acesso à Internet? *

Marcar apenas uma oval.

☐ Sim

☐ Não



17. De modo geral, como você classifica a qualidade da Internet do seu domicílio? *

Marcar apenas uma oval.

- ☐ Ótima
- ☐ Boa
- ☐ Regular
- ☐ Ruim
- ☐ Péssima

18. O que você mais faz com seu dispositivo eletrônico (computador, *tablet*, *notebook*, *smartphone*)? *

Marque todas que se aplicam.

- ☐ Busca materiais para estudos.
- ☐ Mantém comunicação (incluindo as redes sociais).
- ☐ Lê materiais informativos.
- ☐ Faz compras.
- ☐ Utiliza para trabalho.
- ☐ Utiliza para lazer (jogos).

19. Você tem interesse em utilizar ferramentas digitais no seu processo de aprendizagem? *

Marcar apenas uma oval.

- ☐ Sim, totalmente.
- ☐ Sim, parcialmente.
- ☐ Tenho interesse.
- ☐ Não consigo opinar.
- ☐ Não tenho interesse.



20. De modo geral, você considera positiva a utilização de ferramentas digitais no processo de ensino e aprendizagem? *

Marcar apenas uma oval.

- ☐ Muito positiva.
- ☐ Razoavelmente positiva.
- ☐ Pouco positiva.
- ☐ Inadequada.

21. Você já utilizou alguma ferramenta para auxiliar o processo de ensino e aprendizagem? *
Considere o período da pandemia da COVID-19.

Marcar apenas uma oval.

- ☐ Sim *Pular para a pergunta 22*
- ☐ Não *Pular para a pergunta 24*

BLOCO 3 (Parte 2) - CONDIÇÕES TECNOLÓGICAS E PEDAGÓGICAS DE ACESSO A CONTEÚDO DIGITAL

22. Qual(is) a(s) principal(is) ferramenta(s) digitais utilizada(s) por você para apoiar o processo de aprendizagem? *

23. Você já interagiu com a ferramenta Notebooks? *

Marcar apenas uma oval.

- ☐ Sim, tenho muito conhecimento.
- ☐ Sim, conheço superficialmente.
- ☐ Já ouvi falar mas nunca interagi.
- ☐ Nunca ouvi falar nessa ferramenta.

BLOCO 4 - DIAGNÓSTICO DOS CONHECIMENTOS INICIAIS SOBRE LÓGICA DE PROGRAMAÇÃO



24. Antes de cursar a disciplina você já tinha estudado Lógica de Programação? *

Marcar apenas uma oval.

☐ Sim

☐ Não

25. Considerando os conteúdos estudados até o momento, como você classifica o nível de dificuldade da disciplina de Lógica de Programação? *

Marcar apenas uma oval.

☐ Muito difícil.

☐ Difícil.

☐ Nem fácil e nem difícil.

☐ Fácil.

☐ Muito fácil.

26. Você já utilizou alguma linguagem de programação? *

Por exemplo: VBA, C, C#, Python, Php, Java, etc.

Marcar apenas uma oval.

☐ Sim *Pular para a pergunta 27*

☐ Não

**BLOCO 5 (Parte 1) - DIAGNÓSTICO DOS CONHECIMENTOS INICIAIS
SOBRE PYTHON**



27. Você já interagiu com a linguagem de programação Python? *

Marcar apenas uma oval.

- ☐ Sim, tenho muito conhecimento. *Pular para a pergunta 28*
- ☐ Sim, conheço superficialmente. *Pular para a pergunta 28*
- ☐ Já ouvi falar mas nunca interagi.
- ☐ Nunca ouvi falar nessa linguagem de programação.

BLOCO 5 (Parte 2) - DIAGNÓSTICO DOS CONHECIMENTOS INICIAIS SOBRE PYTHON

28. Considerando seu conhecimento sobre Python, como você classifica o nível de dificuldade dessa linguagem de programação? *

Marcar apenas uma oval.

- ☐ Muito difícil.
- ☐ Difícil.
- ☐ Nem fácil e nem difícil.
- ☐ Fácil.
- ☐ Muito fácil.

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários



Apêndice C - Caderno digital: Variáveis, constantes e operadores aritméticos

Introdução à Lógica de Programação

Módulo 0 - Variáveis, constantes e operadores aritméticos

Conteúdos:

- Conceitos de constantes e variáveis
- Tipos de operadores de atribuição e aritméticos
- Comandos de entrada e saída de dados

Ao final deste módulo, você será capaz de:

- Inicializar constantes e variáveis
- Reconhecer o uso dos operadores de atribuição e aritméticos
- Identificar e operar de forma correta os tipos de dados e suas relações
- Construir expressões com operadores e precedência
- Criar, reconhecer, interpretar e aplicar os algoritmos através dos comandos de entrada e saída (`input` e `print`)

CONSTANTES E VARIÁVEIS

O que é uma constante?

Você sabe o que é uma constante?

- Em programação, uma constante armazena um valor fixo, que **NÃO** mudará com o tempo de execução do programa. Ou seja, o valor será definido uma única vez e jamais será alterado durante a execução da aplicação;
- Uma constante deve ser utilizada quando uma informação **NÃO** tem qualquer possibilidade de alteração, ou variação, no decorrer da execução do algoritmo (programa).

Exemplos: π : 3.1415926 - Velocidade da luz no vácuo: 299 792 458 m/s

O que é uma variável?

- É uma entidade destinada a guardar uma informação;
- Chama-se **variável**, pois o valor contido nesta varia com o tempo, ou seja, não é um valor fixo;
- O conteúdo de uma variável pode ser alterado, consultado ou apagado quantas vezes forem necessárias no algoritmo;
- Ao alterar o conteúdo de uma variável, a informação anterior é perdida. Ou seja, a variável armazena sempre a última informação recebida;
- Em geral, uma variável possui três atributos: **nome**, **tipo de dado** e **a informação** por ela guardada.

Python possui os seguintes tipos básicos que veremos nesta aula: * **int**: corresponde aos números inteiros. Ex: 10, -24. **float**: corresponde aos números reais. Ex: 2.4142, 3.141592. **str** ou **string**: corresponde a textos. Ex: "Olá, turma".

⚠️ Regras para nomeação de constantes e variáveis ⚠️ Devem ser iniciadas sempre por uma letra; não devem conter caracteres especiais; não devem conter espaços em branco; não



devem conter hífen entre os nomes (utilize underline).

ATRIBUIÇÃO

Agora iremos falar sobre a atribuição de tais expressões às variáveis:

- Quando um computador executa um algoritmo, o mesmo precisa armazenar e manipular múltiplos valores;
- Esses valores são armazenados e acessados na memória do computador;
- As variáveis podem ser entendidas como "caixas" que podem guardar **um** valor;
- Cada vez que guardamos um valor novo na variável, apagamos o valor antigo;
- Para guardarmos um valor em uma variável usamos algum **comando de atribuição**;
- Como uma demonstração, iremos utilizar o sinal de igual **=** para definir atribuição de valores;
- Será estabelecido que a variável a ser atribuída irá se encontrar à esquerda do símbolo;
- A parte à direita será uma expressão;
- O resultado da expressão será calculado e posteriormente armazenado na variável.

⚠ O comando `=` do Python é o comando de atribuição. Ele associa a variável do lado esquerdo do comando com o objeto do lado direito do comando. Um objeto pode ter um nome associado com ele, mais de um nome ou nenhum nome.

```
In [ ]: ##### Atribuindo o valor 1 à variável teste
teste = 999;
```

OPERADORES

Comparando com a Matemática, as linguagens de programação oferecem um conjunto completo de operadores. Eles consistem em estruturas que permitem a execução de cálculos e comparações.

- Operadores são símbolos que representam atribuições, cálculos e ordem dos dados;
- As operações possuem uma ordem de prioridades (alguns cálculos são processados antes de outros);
- Os operadores são utilizados nas expressões matemáticas, lógicas, relacionais e de atribuição.

A linguagem Python dispõe dos seguintes operadores aritméticos básicos: `+`: adição; `-`: subtração; `*`: multiplicação; `/`: divisão; `%`: resto de uma divisão inteira; `//`: quociente de uma divisão; `**`: exponenciação.

```
In [ ]: ##### Operações matemáticas realizadas com os operações adição, subtração, multiplicação e
adicao = 3+5;
subtracao = 10-7;
multi = 4*6;
div = 25/5;
```

A ordem de prioridade das operações seguem como o usual:

- Primeiro as radiciações e potenciações (`**`);
- Depois, produtos e divisões (`*` e `/`);
- E, por último, somas e subtrações (`+` e `-`).

A ordem de prioridade obedecem às regras matemáticas comuns:

- As expressões dentro de parênteses são sempre resolvidas antes das expressões fora dos parênteses;



- Quando existe um parêntese dentro de outro, a solução sempre inicia do parêntese mais interno até o mais externo (de dentro para fora);
- Quando duas ou mais expressões tiverem a mesma prioridade, a solução é sempre iniciada da expressão mais à esquerda até a mais à direita.

Então, quando a expressão contiver parênteses, deve ser resolvida do mais interno ao mais externo. Por exemplo:

A expressão $(2 + 2) * 4$ é equivalente à 16

A expressão $2 + 2 * 4$ é equivalente à 10

⚠ A ordem de precedência desses operadores segue a mesma regra aplicada na Matemática, ou seja, a multiplicação e divisão têm precedência maior que a adição e a subtração. Isso significa que em uma expressão aritmética, as duas primeiras serão executadas antes das duas últimas. Para definir uma ordem de execução diferente, envolva as expressões desejadas com parênteses. Veja os exemplos a seguir:

```
In [ ]: 5 + 4 * 3 - 8 / 2; #Precedência normal
        (5 + 4) * (3 - 8) / 2; #Precedência alterada
```

No exemplo anterior, as operações de multiplicação $4 * 3$ e de divisão $8 / 2$ são executadas em primeiro lugar, para só então serem executadas a adição e a subtração. Já no segundo caso, como a adição $(5 + 4)$ e a subtração $(3 - 8)$ foram envoltas em parênteses, elas serão executadas primeiro e depois virão a multiplicação e a divisão.

Exemplos:

Aqui estão representados 4 algoritmos distintos e independentes. Determine o valor da variável `x` no final de cada um deles. Escreva o valor que você acredita ser o correto na variável `sua_resposta` e aperte Ctrl + Enter; caso o resultado seja True, sua resposta está correta. Caso contrário, está errada. Ignore as linhas em que o `==` está presente. Ele não afetará no resultado e será explicado nas próximas aulas!

```
In [ ]: #1
        #Tente descobrir o valor de x por linha, mas coloque apenas o valor final na variavel sua_resposta

        x = 5
        x = 2 + 3 * 5
        x = ( 2 + 3 ) * 5

        sua_resposta = 25 #coloque sua resposta aqui!

        x==sua_resposta
```

```
In [ ]: #2

        x = 5
        x = 2 * x

        sua_resposta = 0

        x==sua_resposta
```

```
In [ ]: #3

        y = 6
        z = 8
```



```
x = 2 * z + y  
  
sua_resposta = 0  
  
x==sua_resposta
```

```
In [ ]: #4  
  
x = 6  
x = 1  
x = 3 * x  
  
sua_resposta = 0  
  
x==sua_resposta
```

Saída de dados: *print()*

Para imprimir um texto na tela, utilizamos o comando **print()**.

```
In [ ]: # Imprimindo um texto na tela  
print ("Olá, estamos usando cadernos digitais interativos");  
# Imprimindo o valor da variável  
teste = 3;  
print (teste);
```

Entrada de dados: *input()*

- Realiza a leitura de dados a partir do teclado.
- Aguarda que o usuário digite um valor e atribui o valor digitado a um objeto na memória.
- Todos os dados lidos são do tipo **string*.

```
In [ ]: nome = (input("Digite seu nome: "));  
print (nome);
```

Podemos converter uma *string* lida do teclado em um número inteiro usando a função *int()*.

```
In [ ]: #Exemplo da conversão de uma string para inteiro  
n = int(input("Digite um valor: "));  
print (n);
```

Prática

```
In [ ]: #Atribua o valor 10 para constante x e 11 para constante y,  
#realize a operação matemática soma e imprima o resultado na tela.
```

```
In [ ]: #Faça um programa que leia dois números inteiros e  
#calcule a diferença do primeiro pelo segundo.
```

```
In [ ]: #Faça um programa que leia dois números inteiros e calcule a soma,  
#subtração, multiplicação e divisão do primeiro pelo segundo.
```



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)



Apêndice D - Caderno digital: Estrutura sequencial

Introdução à Lógica de Programação

Módulo 1 – Estrutura sequencial

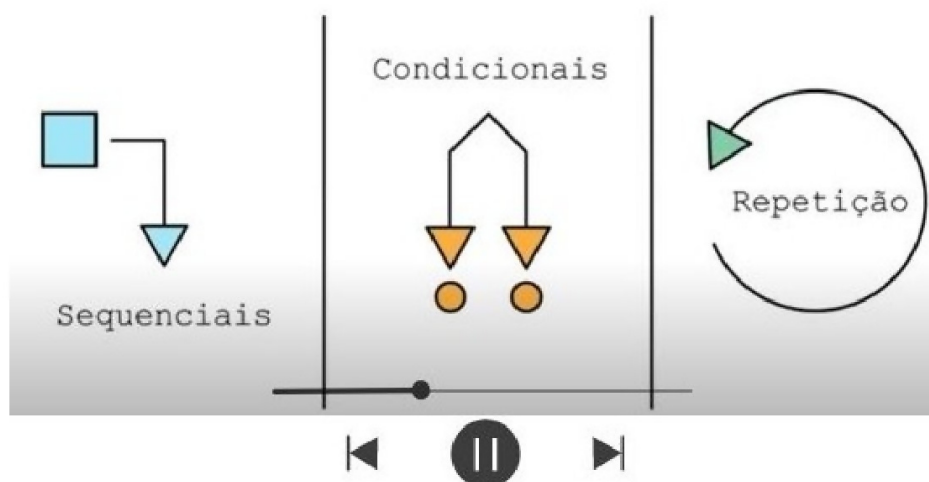
Conteúdos:

- Conceitos de estrutura sequencial
- Funcionamento da estrutura sequencial

Ao final deste módulo, você será capaz de:

- Identificar a necessidade de utilizar estrutura sequencial
- Utilizar a estrutura sequencial

ESTRUTURA DE FLUXO



Fonte: <https://www.youtube.com/watch?v=MNPZMBRdsAU>

Um algoritmo é constituído por estruturas de fluxo que apresentam um conjunto de ações, informações ou instruções e se comportam de diferentes maneiras, sendo elas:

- Estrutura sequencial
- Estruturas de decisão
- Estruturas de repetição

ESTRUTURA SEQUENCIAL

Na estrutura sequencial, os comandos de um algoritmo são executados numa sequência pré-estabelecida. Cada comando é executado somente após o término do comando anterior. Assim sendo, a estrutura sequencial caracteriza-se por um conjunto de comandos dispostos ordenadamente.

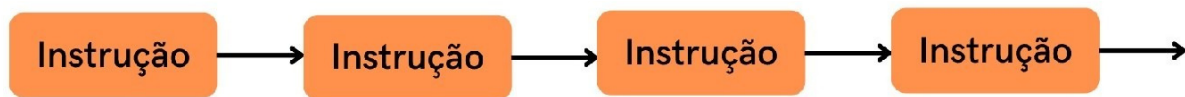


Figura 1: Representação de estrutura sequencial.

Nessa estrutura, os comandos de um algoritmo são executados na ordem em que aparecem. Para que esse conjunto de ações (comandos) se torne viável, deve existir uma perfeita relação lógica intrínseca ao modo pelo qual são executadas, o fluxo de execução.

Imagine a seguinte situação: Você está em casa e gostaria de fazer um delicioso bolo. Como você faria isso?



Instruções:

1. Bata as claras em neve e reserve.
2. Misture as gemas, a margarina e o açúcar até obter uma massa homogênea.
3. Acrescente o leite e a farinha de trigo aos poucos, sem parar de bater.
4. Por último, adicione as claras em neve e o fermento.
5. Despeje a massa em uma forma grande de furo central untada e enfarinhada.
6. Asse em forno médio 180 °C, preaquecido, por aproximadamente 40 minutos ou ao furar o bolo com um garfo, este saia limpo.

Tudo o que os computadores fazem é executar diversas operações baseadas em certas informações fornecidas a ele. Essas informações são as **entradas**. Elas permitem ao usuário a comunicação entre o mundo exterior e a máquina, executando o programa. Informações fornecidas pelo usuário são, na maioria das vezes, armazenadas no que chamamos de **variáveis**. Por outro lado, as saídas são os resultados esperados do programa, podendo ser mensagens, expressões e valores armazenados em variáveis. Por exemplo, em uma função $y = x + 3$, x é o valor de entrada, enquanto y , valor que desejo descobrir, é a saída.

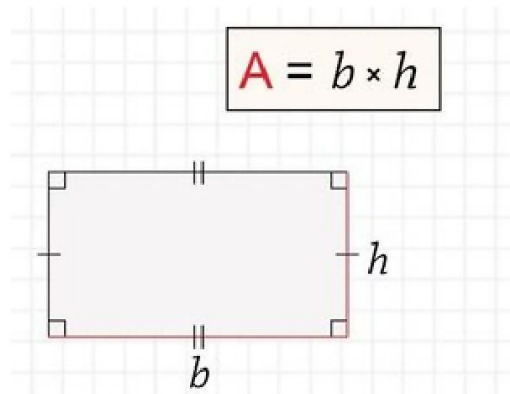
Portanto, em algoritmo, primeiramente, é feita a declaração das variáveis. Posteriormente, são executados os comandos de entrada e/ou atribuição. Na sequência, é realizado o processamento dos dados e, no final, realiza-se a saída de dados.



ESTRUTURA SEQUENCIAL EM PYTHON

A estrutura sequencial é um programa Python que possui entrada de dados, processamento desses dados em resultados e a saída desses resultados.

Exemplo 1: Descobrir a área de um retângulo dado o valor da base e da altura. É fundamental que saibamos as entradas e as saídas dos problemas que queremos resolver. Neste algoritmo, nosso objetivo é encontrar a área (que configura como nossa saída), utilizando obrigatoriamente de duas entradas: base e altura.



Instruções:

1. Informar o valor da base.
2. Informar o valor da altura.
3. Multiplicar o valor da base pelo valor da altura.
4. O resultado do passo 3 é a área do retângulo.

```
In [ ]: #Exemplo 1
#Desenvolva o exemplo do algoritmo do exemplo 1 em Python.
b = int(input("Informe o valor da base do retângulo: "));
h = int(input("Informe o valor da altura do retângulo: "));
a = b*h;
print ("Área (b*h) =",a);
```



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)



Apêndice E - Caderno digital: Estrutura sequencial (exercícios)

Exercícios - Estrutura sequencial

Faça um algoritmo para calcular a área de uma circunferência, considerando a fórmula $\text{ÁREA} = \pi * \text{RAIO}^2$. Utilize as variáveis AREA e RAIO, a constante π ($\pi = 3,14159$) e os operadores aritméticos de multiplicação.

In []:

Faça um algoritmo que calcule a área de um triângulo, considerando a fórmula $\text{AREA} = (\text{BASE} * \text{ALTURA}) / 2$. Utilize as variáveis AREA, BASE e ALTURA e os operadores aritméticos de multiplicação e divisão.

In []:

Calcule a área de um retângulo. A área de uma figura geométrica retangular é calculada pela fórmula $\text{área} = \text{largura} * \text{altura}$.

In []:

Faça um algoritmo que leia uma temperatura em graus Celsius e apresente-a convertida em graus Fahrenheit. A fórmula de conversão é: $F = (9 * C + 160) / 5$, na qual F é a temperatura em Fahrenheit e C é a temperatura em Celsius.

In []:

Faça um algoritmo que calcule e apresente o valor do volume de uma lata de óleo, utilizando a fórmula $\text{VOLUME} = 3,14159 * \text{RAIO}^2 * \text{ALTURA}$.

In []:

Faça um programa que leia em horas e defina a quantidade de minutos.

In []:



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)



Apêndice F - Caderno digital: Estruturas de decisão

Introdução à Lógica de Programação

Módulo 2 – Estruturas de decisão

Conteúdos:

- Conceitos de estruturas de decisão
- Funcionamento das estruturas de decisão If/Else

Ao final deste módulo, você será capaz de:

- Identificar a necessidade de utilizar estruturas de decisão
- Utilizar as estruturas de decisão If/Else

ESTRUTURAS DE DECISÃO

Você sabe o que são estruturas de decisão, também conhecidas como estruturas condicionais?

- São comandos que auxiliam no direcionamento da sequência de execução de um programa por meio da avaliação de **condições lógicas**;
- Tem como função validar condições e comparar o resultado destas.

Algoritmo condicional: Permite a **escolha** de um grupo de **ações** a ser executado quando determinadas **condições**, representadas por expressões lógicas, são ou não satisfeitas.

- Um comando condicional é aquele que permite decidir se um determinado bloco de comandos deve ou não ser executado, a partir do resultado de uma expressão relacional ou lógica.

Para que servem?

- Permitem alterar o **fluxo de execução** do algoritmo, de forma a selecionar qual parte deve ser executada;
- Essa “decisão” de execução é tomada a partir de uma condição, que pode resultar apenas dois valores: **verdadeiro** ou **falso**;
- Uma condição é representada por **expressões relacionais** ou **lógicas**.

Expressões relacionais

Expressões relacionais são aquelas que realizam uma comparação entre duas expressões e retornam:

- False, se o resultado é falso.
- True, se o resultado é verdadeiro.

Operadores relacionais

Os operadores relacionais da linguagem Python são:

- == : igualdade
- != : diferente
- > : maior que



- < : menor que
- >= : maior ou igual que
- <= : menor ou igual que

⚠ Uma expressão booleana é uma expressão que produz como resultado um valor booleano. Expressões relacionais produzem um resultado booleano, logo, são **expressões booleanas**. ⚠

Expressões lógicas

- Expressões lógicas são aquelas que realizam uma operação lógica (**ou, e, não**, etc...) e retornam **True** ou **False** (como as expressões relacionais).
- Na linguagem Python temos os seguintes operadores lógicos:
 - **and**: operador E
 - **or**: operador OU
 - **not**: operador NÃO
- expressão **and** expressão : Retorna verdadeiro quando ambas as expressões são verdadeiras.
- expressão **or** expressão : Retorna verdadeiro quando pelo menos uma das expressões é verdadeira.
- **not** expressão : Retorna verdadeiro quando a expressão é falsa e vice-versa.

Funcionamento das estruturas de decisão

- Após executar as funções de validação e comparação, as estruturas de seleção irão executar os blocos de comando, definidos de acordo com o resultado da comparação (**verdadeiro** ou **falso**).

Tipos de estruturas de decisão

- If/Else (Se/Então);
- Switch/Case (Escolha/Caso)

ESTRUTURAS DE DECISÃO IF/ELSE

Tipos de estruturas IF/ELSE:

- Simples;
- Compostas;
- Aninhadas.

Estrutura de decisão simples

Como funciona?

- É um conjunto de instruções agrupadas.
- Os comandos agrupados do bloco devem estar indentados dentro de um comando anterior seguido de dois pontos.
- A indentação é feita em geral com 2 espaços em branco (ou quantos você quiser) antes de cada comando que deve estar dentro do bloco.
- O principal comando condicional é o if, cuja sintaxe é:

if expressão relacional ou lógica:



comandos executados se a expressão for verdadeira

- Os comandos são executados somente se a expressão relacional/lógica for verdadeira.

O programa determina se um valor é par.

```
In [ ]: # Informa se o número é par.
```

Estrutura de decisão composta

- Uma variação do comando if é o if/else, cuja sintaxe é:

if expressão relacional ou lógica:

comandos executados se a expressão for verdadeira

else:

comandos executados se a expressão for falsa

```
In [ ]: #Faça um algoritmo que lê três notas, calcula a média e informa se o aluno está
#aprovado ou não. O aluno estará aprovado se sua média for maior ou igual a 7.
```

```
In [ ]: # Um programa que o usuário deve digitar um número entre 5 e 15. Se o
# número não estiver nessa faixa, deve ser exibida a mensagem "FORA DO CONJUNTO",
#se o número atende a regra, mostre a mensagem "PERTENCE AO CONJUNTO".
```

```
In [ ]: # Informa se o número é par ou ímpar.
```

```
In [ ]: # Faça um programa que leia dos números e determine o menor deles.
```

Estrutura de decisão aninhada

- É utilizada, em geral, quando é necessário realizar várias comparações com a mesma variável;
- É chamada de aninhada porque na sua representação fica uma seleção dentro de outra seleção;
- Também é conhecida como seleção "encadeada";
- Permite fazer a escolha de apenas um entre vários comandos possíveis.

```
In [ ]: #Usando apenas operadores relacionais e aritméticos, vamos escrever um programa que lê u
#e verifica em qual dos seguintes casos o número se enquadra:
#Par e menor que 100
#Par e maior ou igual a 100
#Ímpar e menor que 100
#Ímpar e maior ou igual a 100
```

```
In [ ]: #Escreva um programa que lê três números e imprime o maior deles.
```

```
In [ ]: #Escreva um programa que lê três números distintos e os imprime em ordem decrescente.
```

```
In [ ]: #Um o programa pede 3 notas do aluno. Se a média das notas for maior ou igual a 7,0 exib
#a mensagem "APROVADO", caso contrário o programa solicita a nota da recuperação. Se a m
#após a recuperação for maior ou igual a 5, exibe a mensagem "APROVADO APÓS A RECUPERAÇÃ
#caso contrário o programa exibe a mensagem "REPROVADO".
```





Apêndice G - Caderno digital: Estruturas de decisão (exercícios)

Exercícios - Estruturas de decisão If/Else

Vamos supor a seguinte situação: escreva um algoritmo que realiza a soma de dois números e confira se o resultado dessa soma será maior ou igual a 50. Se for maior, deverá retornar o valor da soma e, caso a condição não seja atendida, o algoritmo deve encerrar sua execução.

In []:

Escreva um algoritmo que realiza a soma de dois números e confira se o valor obtido como resultado dessa soma seja maior ou igual a 50. Vamos definir que caso essa condição seja aceita, o valor da soma deverá ser somado com 2; caso o valor somado não seja maior ou igual a 50, o valor da soma deve ser subtraído por 10.

In []:

Uma produtora de eventos vai oferecer descontos no ingresso de um evento baseado na idade do usuário, segundo os seguintes critérios: quem tem entre 0 e 11 anos (inclusive), desconto de 100%; de 12 anos e menor de 26 anos (inclusive), 50% de desconto; maior de 26 anos, não terá desconto. Escreva um algoritmo que, dada a idade do usuário, informe o percentual de desconto.

In []:

Uma empresa vai conceder um aumento diferenciado a seus funcionários, segundo os seguintes critérios: quem ganha entre 1000 e 2000 (inclusive), aumento de 18 %; entre 2000 e 4000 (inclusive) aumento de 15 % e acima de 4000 aumento de 10 %. Escreva um algoritmo que, dado um valor de salário, calcule o novo valor após o aumento.

In []:

Sabendo que um triângulo é dito equilátero quando tem 3 lados iguais, isósceles quando tem 2 lados iguais e escaleno quando todos os lados têm tamanhos diferentes, escreva um algoritmo que receba os valores dos três lados de um triângulo e imprima se ele é equilátero, isósceles ou escaleno.

In []:

Uma fruteira está vendendo frutas com a seguinte tabela de preços:

	Até 5 Kg	Acima de 5 Kg
Morango	R\$ 2,50 por Kg	R\$ 2,20 por Kg
Maçã	R\$ 1,80 por Kg	R\$ 1,50 por Kg

Se o cliente comprar mais de 8 Kg em frutas ou o valor total da compra ultrapassar R\$ 25,00, receberá ainda um desconto de 10% sobre este total. Escreva um algoritmo para ler a quantidade (em Kg) de morangos e a quantidade (em Kg) de maçãs adquiridas e escreva o valor a ser pago pelo cliente.

In []:





Apêndice H - Questionário de avaliação do produto educacional



Avaliação do Produto Educacional

Este questionário faz parte do Produto Educacional SEQUÊNCIA DIDÁTICA PARA O ENSINO DE LÓGICA DE PROGRAMAÇÃO UTILIZANDO O JUPYTER NOTEBOOK, apresentado ao Programa de Pós-Graduação em Tecnologias Educacionais em Rede (PPGTER), da Universidade Federal de Santa Maria (UFSM, RS). Desenvolvido por Mayara Leal Reis Fernandes sob a orientação do Prof. Dr. André Zanki Cordenonsi.

mayara.fernandes@ifma.edu.br [Alternar conta](#)

 Não compartilhado



* Indica uma pergunta obrigatória

Nome *

Sua resposta



Por favor, **marque uma opção** de acordo com o quanto você concorda ou discorda de cada afirmação abaixo em relação aos cadernos digitais utilizados na oficina. *

Marque uma opção conforme sua avaliação.

	Discordo totalmente	Discordo	Nem discordo, nem concordo	Concordo	Concordo totalmente
Conseguir compreender facilmente o funcionamento dos cadernos digitais.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eu acho que a maioria das pessoas aprenderiam utilizar os cadernos digitais rapidamente.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estou satisfeito(a) com a organização dos conteúdos na sequência didática do capítulo Estrutura Sequencial.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estou satisfeito(a) com a organização dos exercícios da sequência didática do capítulo Estruturas de Decisão.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>





A utilização dos cadernos digitais contribuiu para o aprendizado de Lógica de Programação.

☐☐☐☐☐

A utilização dos cadernos digitais me motivou a aprender Lógica de Programação.

☐☐☐☐☐

Me sinto satisfeito(a) com em aprender lógica de programação com Python utilizando os cadernos digitais.

☐☐☐☐☐

Completar as tarefas práticas utilizando os cadernos digitais me deu um sentimento de realização.

☐☐☐☐☐

Utilizar os cadernos digitais proporcionou uma aula dinâmica.

☐☐☐☐☐

Eu prefiro aprender lógica de programação utilizando os cadernos digitais do que

☐☐☐☐☐



de outra forma
(outro método
de ensino).

Indicaria os
cadernos
digitais à outros
colegas



O que você mais gostou ao utilizar os cadernos digitais para aprender lógica de programação? *

Sua resposta

O que você acha que poderia ser melhorado? *

Sua resposta

Gostaria de fazer algum comentário sobre a oficina?

Sua resposta

Página 1 de 1

Enviar

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este formulário foi criado em Webmail da Universidade Federal de Santa Maria. [Denunciar abuso](#)

Google Formulários

