Algoritmos e Lógica de Programação

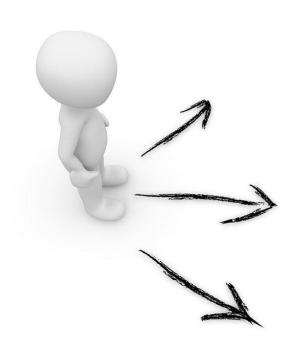
Prof. Flavius Gorgônio

Semana 4

Estruturas de controle de decisão: A estrutura IF

Agenda

- Motivação
- Estruturas de controle de fluxo
 - Estruturas de controle de decisão
 - Estruturas de controle de repetição (próximo assunto)
- > Estruturas de controle de decisão
 - Decisão simples
 - Decisão composta
 - Decisões compostas encadeadas
 - Decisão por seleção múltipla
- Aplicações e exemplos

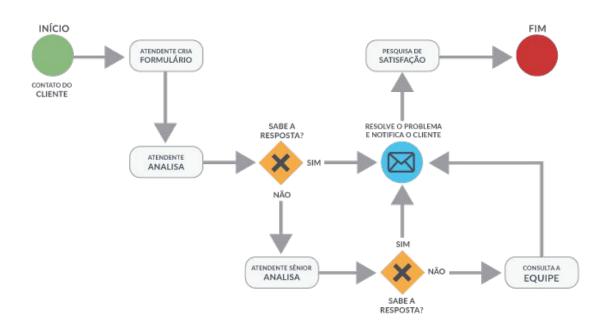


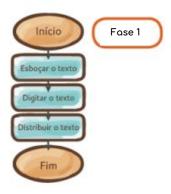
Motivação

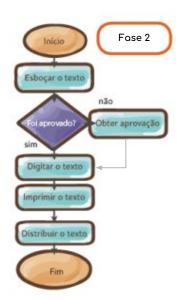
Suponha que você precisa implementar o fluxograma de atendimento da imagem ao lado através de um programa de computador

Perceba que o **programa** precisará TOMAR DECISÕES em determinados momentos do fluxo

Nos losangos amarelos, o passo seguinte depende necessariamente







Computado niciar o Word Revisar o texto Fazer correções correções? o texto Distribuir o texto

Esboçar o texto

Fase 3

Algumas vezes, é necessário fazer refinamentos sucessivos

Exemplos de decisões em sistemas eletrônicos

Sistemas de hardware

- O termostato do ar condicionado percebe que a sala está quente e aciona o condensador para refrigerar o ambiente
- O sensor de presença percebe uma movimentação e dispara um alarme
- O sistema de acionamento de um portão eletrônico recebe um sinal de radiofrequência, reconhece o sinal e aciona o motor do portão para abri-lo

Sistemas de software

- O sistema verifica login e senha informados pelo usuário e decide se ele deve ter seu acesso permitido ou negado
- O SIGAA verifica quais usuários da biblioteca estão com livros em atraso e envia um email de notificação
- O sistema comercial de uma loja consulta o CPF do cliente e verifica sua margem de crédito

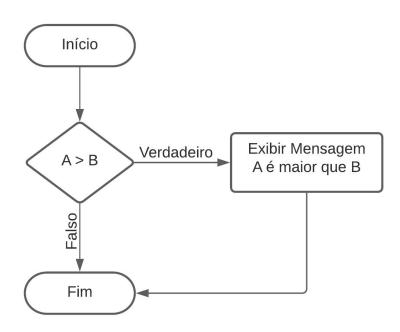
Decisões simples

Uma decisão simples permite executar ou não um ou mais comandos (bloco de comandos)

Se a expressão resultar em VERDADEIRO, o comando (ou bloco de comandos) subordinado à estrutura é executado

Se resultar em FALSO, o comando (ou bloco de comandos) subordinado à estrutura não é executado

Em ambos os casos, a execução continua com os comandos subsequentes à estrutura de decisão



Estrutura if em Python

Python avalia uma expressão lógica (boolean) para decidir se executa ou não um comando (ou bloco de comandos) subordinado ao if

Expressões lógicas são construídas a partir de operadores aritméticos, relacionais e lógicos, além de expressões

O trecho afastado à direita (indentado) só será realizado se a condição for verdadeira

```
# trecho antes do if
if <expressão_lógica>:
    instrução_1
    instrução_2
    ...
    instrução_n
# trecho após o if
# continuação do programa
```

Implementando uma decisão simples em Python

Escreva um programa em Python que realize os seguintes passos:

- Cumprimente o usuário
- Solicite que ele informe o seu nome
- Solicite seu ano de nascimento
- Obtenha o ano corrente a partir do sistema operacional
- Calcule a idade do usuário a partir do seu ano de nascimento e do ano atual
- Verifique se o usuário está na lista de prioridades para receber a 4ª dose da vacina contra a COVID-19 (idade ≥ 60)

```
import datetime
idade limite = 60
print("Olá, prezado usuário!")
nome = input("Qual o seu nome? ")
ano nasc = int(input("Em que ano você nasceu? "))
ano atual = datetime.date.today().year
idade = ano atual - ano nasc
print("%s, esse ano você completa %d anos"%(nome, idade))
if idade >= idade limite:
   print("%s, você já se vacinou contra a COVID?"%nome)
   print("Estou vendo aqui que já chegou na sua faixa etária")
   print("Se não tomou ainda, procure um posto de saúde")
print("Tenha um bom dia!")
```

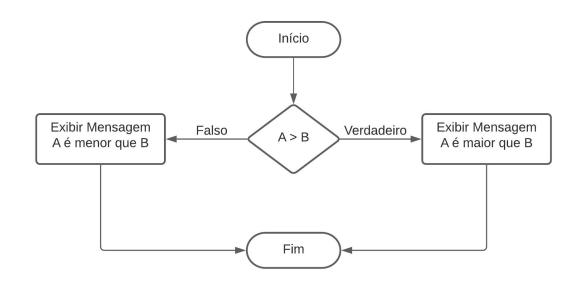
Decisões compostas

Uma decisão composta permite executar um ou outro comando (ou bloco de comandos)

Se a expressão resultar em VERDADEIRO, o comando (ou bloco de comandos) subordinado à estrutura if é executado

Se resultar em FALSO, o comando (ou bloco de comandos) subordinado à cláusula <u>else</u> é executado

Em ambos os casos, a execução continua com os comandos subsequentes à estrutura de decisão



Estrutura if...else em Python

Python avalia uma expressão lógica para decidir se executa <u>um</u> ou <u>outro</u> bloco de comandos subordinado ao if...else

A cláusula <u>else</u> não possui teste e só é executada se o resultado do teste <u>if</u> for falso. Em ambos os casos um, e apenas um, dos trechos será executado

O trecho indentado em azul só será executado se a expressão for <u>True</u> e o trecho indentado em verde só será executado se a expressão for <u>False</u>

```
# trecho antes do if
if <expressão lógica>:
    instrução 1 do if
    instrução n do if
else:
    instrução 1 do else
    instrução_p_do_else
# trecho após o if...else
# continuação do programa
```

Implementando uma decisão composta em Python

Escreva um programa em Python que realize os seguintes passos:

- Cumprimente o usuário
- Solicite que ele informe o seu nome
- Solicite seu ano de nascimento
- Obtenha o ano corrente a partir do sistema operacional
- Calcule a idade do usuário a partir do seu ano de nascimento e do ano atual
- Verifique se o usuário está na lista de prioridades para receber a última dose da vacina contra a COVID-19 (idade ≥ 60)

```
import datetime
idade limite = 60
print("Olá, prezado usuário!")
nome = input("Qual o seu nome? ")
ano nasc = int(input("Em que ano você nasceu? "))
ano atual = datetime.date.today().year
idade = ano atual - ano nasc
print("%s, esse ano você completa %d anos"%(nome, idade))
if idade >= idade limite:
   print("%s, você já se vacinou contra a COVID?"%nome)
   print("Estou vendo aqui que já chegou na sua faixa etária")
   print("Se não tomou ainda, procure um posto de saúde")
else:
   print("%s, ainda não chegou sua vez de se vacinar"%nome)
   print("Mas fique atento(a) ao calendário de vacinação")
print("Tenha um bom dia!")
```

Vamos exercitar um pouco?

- 1. Escreva um programa em Python que leia os nomes e as idades de dois usuários. Em seguida, o programa deve verificar e exibir na tela quem é o mais velho e quem é o mais novo (exibir os nomes dos usuários)
- 2. Escreva um programa em Python que leia os nomes e as idades de dois usuários. O programa deve calcular e exibir na tela a diferença de idade entre eles. Independente da ordem em que os dados forem digitados, a diferença entre as idades deve ser exibida sempre como um valor positivo

- 3. Escreva um programa em Python que determine se um determinado ano é ou não bissexto. Um ano é bissexto quando é divisível por 4 e não é divisível por 100, ou ainda, quando é divisível por 400.
- 4. Escreva um programa em Python que leia a data de nascimento de um usuário, obtenha a data atual do sistema e calcule, exatamente, quantos anos ele possui, verificando se o usuário já completou ou não aniversário no ano atual.

Solução Exercício 1

Escreva um programa em Python que leia os nomes e as idades de dois usuários. Em seguida, o programa deve verificar e exibir na tela quem é o mais velho e quem é o mais novo (exibir os nomes dos usuários)



funciona?

```
print("Programa Compara Idades")
print("Informe os nomes e idades de duas pessoas")
nome1 = input("Nome da pessoa 1: ")
idade1 = int(input("Idade da pessoa 1: "))
nome2 = input("Nome da pessoa 2: ")
idade2 = int(input("Idade da pessoa 2: "))
if idade1 > idade2:
 print(nome1, "é mais velho que", nome2)
else:
  print(nome2, "é mais velho que", nome1)
```

Decisões compostas encadeadas

A maioria das LPs permite o encadeamento de estruturas de decisão compostas, onde vários comandos (ou blocos de comandos) são sequenciados para a construção de uma estrutura única de decisão exclusiva dentre múltiplas condições

Nesse caso, após a cláusula SENÃO segue outro teste SE, e assim por diante

Se a expressão resultar em VERDADEIRO, o comando (ou bloco de comandos) diretamente subordinado ao teste é executado

Se resultar em FALSO, o próximo teste é verificado e, como no caso anterior, se a expressão resultar em VERDADEIRO, o comando (ou bloco de comandos) diretamente subordinado ao teste é executado

Se resultar em FALSO, passa-se ao teste seguinte e assim por diante

Como na estrutura anterior, se o último bloco for um SENÃO, então apenas UM dentre os comandos (ou blocos de comandos) é executado

Se não houver uma cláusula SENÃO, pode ser que nenhum dos blocos seja executado

Em ambos os casos, a execução continua com os comandos subsequentes à estrutura de decisão

Estrutura if...elif...else em Python

Python avalia a primeira expressão lógica para decidir se executa ou não o bloco de comandos subordinado ao teste if

Se o primeiro teste falhar, o Python avalia a segunda expressão, que está subordinada ao elif, aplicando as mesmas regras do teste anterior

Se qualquer dos testes for verdadeiro, o bloco subordinado é executado e a estrutura é encerrada. Se todos os testes falharem, executa-se o bloco do <u>else</u>

```
# trecho antes do if
if <expressão lógica>:
     instrução 1 do if
     instrução n do if
elif <expressão lógica>:
     instrução 1 do elif
     instrução n do elif
else:
     instrução 1 do else
     instrução p do else
# trecho após o if...elif...else
# continuação do programa
```

Outra solução para o Exercício 1

Escreva um programa em Python que leia os nomes e as idades de dois usuários. Em seguida, o programa deve verificar e exibir na tela quem é o mais velho e quem é o mais novo (exibir os nomes dos usuários)



```
print("Programa Compara Idades")
print("Informe os nomes e idades de duas pessoas")
nome1 = input("Nome da pessoa 1: ")
idade1 = int(input("Idade da pessoa 1: "))
nome2 = input("Nome da pessoa 2: ")
idade2 = int(input("Idade da pessoa 2: "))
if idade1 > idade2:
  print(nome1, "é mais velho que", nome2)
elif idade2 > idade1:
  print(nome2, "é mais velho que", nome1)
else:
  print(nome2, "e", nome1, "possuem a mesma idade")
```

Solução Exercício 2

Escreva um programa em Python que leia os nomes e as idades de dois usuários

O programa deve calcular e exibir na tela a diferença de idade entre eles. Independente da ordem em que os dados forem digitados, a diferença entre as idades deve ser exibida sempre como um valor positivo

```
print("Programa Diferença de Idades")
print("Informe os nomes e idades de duas pessoas")
nome1 = input("Nome da pessoa 1: ")
idade1 = int(input("Idade da pessoa 1: "))
nome2 = input("Nome da pessoa 2: ")
idade2 = int(input("Idade da pessoa 2: "))
if idade1 > idade2:
 dif = idade1 - idade2
 print("%s é %d ano(s) mais velho(a) do que %s"%(nome1, dif, nome2))
elif idade2 > idade1:
  diferenca = idade2 - idade1
 print("%s é %d ano(s) mais velho(a) do que %s"%(nome2, dif, nome1))
else:
 print(nome1, "e", nome2, "possuem a mesma idade")
```

Solução Exercício 3

Escreva um programa em Python que determine se um determinado ano é ou não bissexto

Um ano é bissexto quando é divisível por 4 e não é divisível por 100, ou ainda, quando é divisível por 400

```
(ano % 4) == 0 and (ano % 100) != 0
ou (ano % 400) == 0
```

```
if ((ano % 4 == 0) and (ano % 100 != 0)) or (ano % 400 == 0):
    print(ano, "é bissexto")
else:
    print(ano, "não é bissexto")
```

```
print("Programa Testa Bissexto")
ano = int(input("Informe o ano: "))
if (ano % 4 == 0) and (ano % 100 != 0):
  print(ano, "é bissexto")
elif (ano % 400 == 0):
  print(ano, "é bissexto")
else:
  print(ano, "não é bissexto")
```

Desafio Semana 4

Escreva um programa em Python que leia a data de nascimento de um usuário, obtenha a data atual do sistema e calcule, exatamente, quantos anos ele possui, verificando se o usuário já completou ou não aniversário no ano atual



Sugestões de bibliografia

Básica

- MENEZES, Nilo Ney Coutinho. Introdução à programação com Python: algoritmos e lógica de programação para iniciantes. 2a ed. São Paulo: Novatec, 2010. 328p. ISBN: 9788575224083
- FARRELL, Joyce. Lógica e design de programação: introdução. São Paulo: Cengage Learning, 2010. xiv, 416p. ISBN: 9788522107575.
- SOUZA, Marco; GOMES, Marcelo; SOARES, Márcio; CONCILIO, Ricardo. Algoritmos e Lógica de Programação. 2a ed. São Paulo: Cengage Learning, 2013. 240 p. ISBN: 9788522111299

Complementar

- ASCENCIO, Ana Fernanda; CAMPOS, Edilene Aparecida. Fundamentos da programação de computadores: Algoritmos,
 Pascal, C/C++(Padrão Ansi) e Java. 3a ed. São Paulo: Pearson Prentice Hall. 569 p. ISBN: 9788564574168
- > ZELLE, John M. Python programming: an introduction to computer science. 2nd ed. Sherwood, Or.: Franklin, Beedle & Associates, c2010. xiv, 514 p. ISBN: 9781590282410.
- LUTZ, Mark; ASCHER, David. Learning Python. 4nd ed. Sebastopol, CA: O Reilly, c2009. xliv, 1162 p. ISBN: 978059615064.



Copyright

Este material é para uso exclusivo durante as aulas da disciplina DCT1101 - ALGORITMOS E LÓGICA DE PROGRAMAÇÃO, do curso de Bacharelado em Sistemas de Informação da Universidade Federal do Rio Grande do Norte, não estando autorizada a sua publicação, compartilhamento, divulgação ou utilização em outros contextos diferentes dos aqui apresentados.

Todos os textos, imagens, exemplo de código e demais materiais utilizados nesses slides são apenas para fins didáticos, o autor e a instituição não permitem a sua utilização sem autorização expressa, assim como não se responsabilizam pelo seu uso indevido ou por danos causados pelos mesmos.