

Área de Ciências Exatas e Engenharias Lógica Computacional e Programação

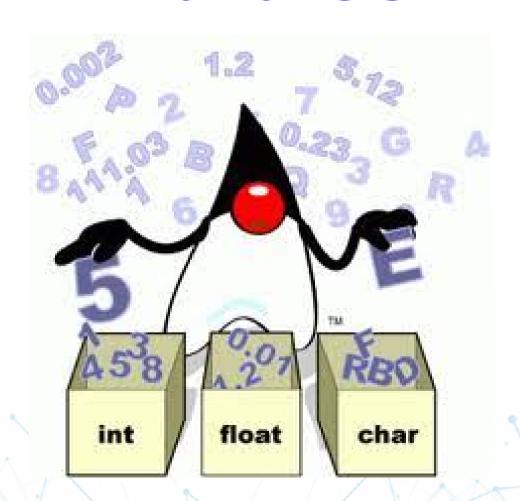
Comandos Sequenciais

Professoras:
Carine Webber
Maria de Fátima Webber do Prado Lima





Tipos de Dados e Variáveis







Tipos de Dados

- Os computadores podem trabalhar com diversos tipos de dados.
- Os tipos de dados mais comuns são:
 - Dados numéricos inteiros (int)
 - Dados numéricos reais (float)
 - Dados literais (strings)
 - Dados lógicos





Tipos de Dados

- Dados numéricos inteiros (int):
 - Números que pertencem aos conjuntos N (naturais) e Z (inteiros).
 - Exemplos: 24, 0, -12.
- Dados numéricos reais (float):
 - Números que pertencem aos conjuntos Q (fracionários)
 e R (reais).
 - Utiliza-se "." para separar a parte inteira da parte decimal.
 - Exemplos: 12.01, 14.543, -13.3





Tipos de Dados

- Dados literais ou cadeia de caracteres ou strings:
 - Sequência de caracteres contendo qualquer letra, dígito ou símbolo representado pelo computador.
 - Delimitado por aspas duplas ("...") ou aspas simples ('....').
 - Exemplos: "Asdfg", 'XYZ', "\$ 200", '123', 'R\$ 233,00'
- Dados lógicos ou booleanos:
 - Representa os dois únicos valores lógicos possíveis:
 VERDADEIRO e FALSO.





Constantes

- São valores fixos.
- Não são alterados na execução do programa.
- Exemplo: quando calculamos a média aritmética de 3 números (n1, n2 e n3) através da fórmula (n1+n2+n3)/3:
 - o divisor 3 nunca é alterado, pois ele é uma constante nessa fórmula.
- O Python possui algumas constantes definidas:
 - Exemplo: PI = 3.14159265





Variáveis

- Os dados manipulados pelos computadores são armazenados na memória RAM (Random Access Memory) do computador:
 - Portanto, os dados que nossos programas trabalham são armazenados na memória RAM
- É necessário:
 - Alocar (reservar) uma posição para armazenar os dados.
 - Indicar a posição de memória reservada para guardar o dado.
 - Armazenar o dado.
 - Indicar a posição de memória reservada para acessar o dado armazenado na memória.
 - Acessar o dado.





Variáveis

 O computador automaticamente aloca a posição de memória, realiza o acesso a memória, armazena o dado e acessa o dado através de uma abstração da memória RAM:

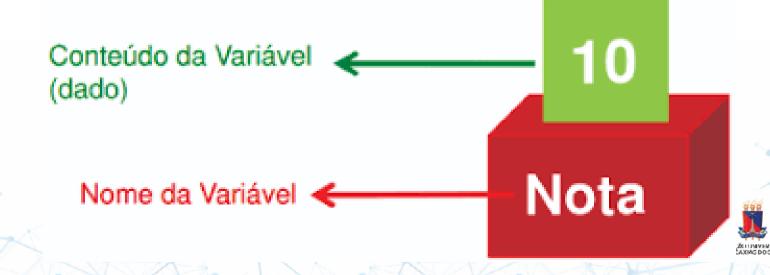
Variável





Variáveis

- A variável pode ser vista como uma entidade destinada a guardar uma informação.
- Toda variável possui:
 - Um tipo
 - Um nome
 - Um valor armazenado





Tipo de uma Variável

GUARANÁ







Garrafeira para transporte de 24 garrafas de 340 ml





Tipo de uma Variável





Garrafeira de 12 garrafas PET (retornável) de 2000 ml.







Tipo de uma Variável

- Indica o tamanho da posição de memória que deve ser reservado para armazenar um dado.
 - Podem ser:
 - Int (números inteiros)
 - Float (número reais)
 - String (cadeia de caracteres)
 - Bool (booleanos)





Nome de uma Variável

- É uma abstração do endereço de memória onde a informação está armazenada:
 - Nos programas criamos variáveis para acessar os valores que estão armazenados na memória.
 - O computador automaticamente associa esse nome de variável ao endereço de memória onde o valor está armazenado.





Nome de uma Variável

- É aconselhável definirmos nomes de variáveis que estejam associados com o conteúdo que será armazenado. Exemplos:
 - media: armazenar o valor do cálculo de uma média.
 - desc: armazenar o valor do desconto.
 - sal_liq: armazenar o valor do salário líquido.





Nome de uma Variável

- · Regras para definição dos nomes das variáveis:
 - O primeiro caractere do nome deverá ser sempre alfabético.
 - Não podem ser utilizados caracteres especiais (& %#@:+-):
 - Exceção: Underscore "_".





As variáveis possuem nomes válidos?

A

X

y1

123

2a

valor

val medio

val-medio

val@medio

val_medio







Como atribuímos valores as variáveis?

- Através de comandos de entrada de dados.
- Através de comandos de atribuição. Exemplos:

$$x=5$$

$$y=7$$







Python







Python

- Linguagem de programação de alto nível
- Criada pelo holandês Guido von Rossum
- Características:
 - Orientada a objetos
 - Modular
 - Disponível em vários sistemas operacionais
 - Interpretada
 - Tipagem dinâmica





Python

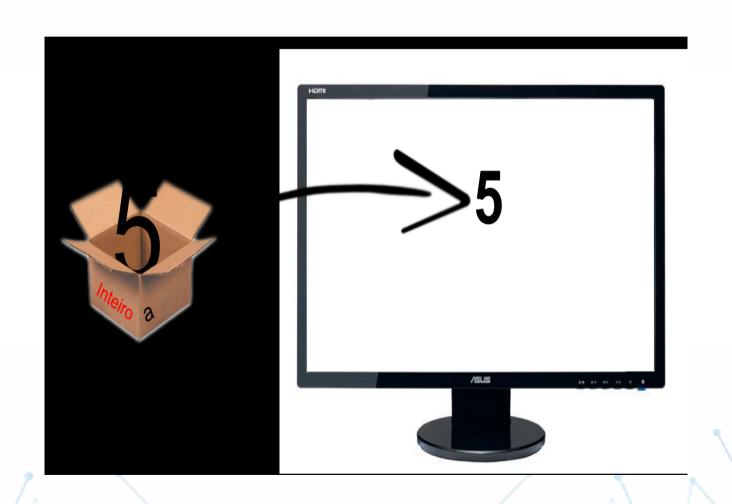
- Características:
 - Iterativa: o interpretador disponibiliza uma interface onde é possível inserir os comandos um por um e visualizar o efeito de cada um deles.

```
Type "copyright", "credits" or "license()" for more information.
>>> print ('Pensamento Computacional')
Pensamento Computacional
>>> x=10
>>> print (x)
10
>>>
```





Comandos de Saída de Dados







Qual a finalidade dos comandos de saída de dados?







Qual a finalidade dos comandos de saída de dados?

Mostrar / armazenar o resultado do processamento efetuado pelo computador.













Qual o comando em python é utilizado para mostrar os dados processados pelo computador na tela?





Qual o comando em python é utilizado para mostrar os dados processados pelo computador na tela?

Print





Comando print

- Utilizado para mostrar mensagens na tela:
- As mensagens podem ser:
 - string :cadeia de caracteres alfanuméricos entre aspas
 - conteúdo de variáveis





```
>>> print('Lógica Computacional e Programação')
Lógica Computacional e Programação
>>>
```





$$>>> x=78$$

78





```
>>> z=100
>>> print('O valor de z é ', z , '.')
O valor de z é 100 .
```

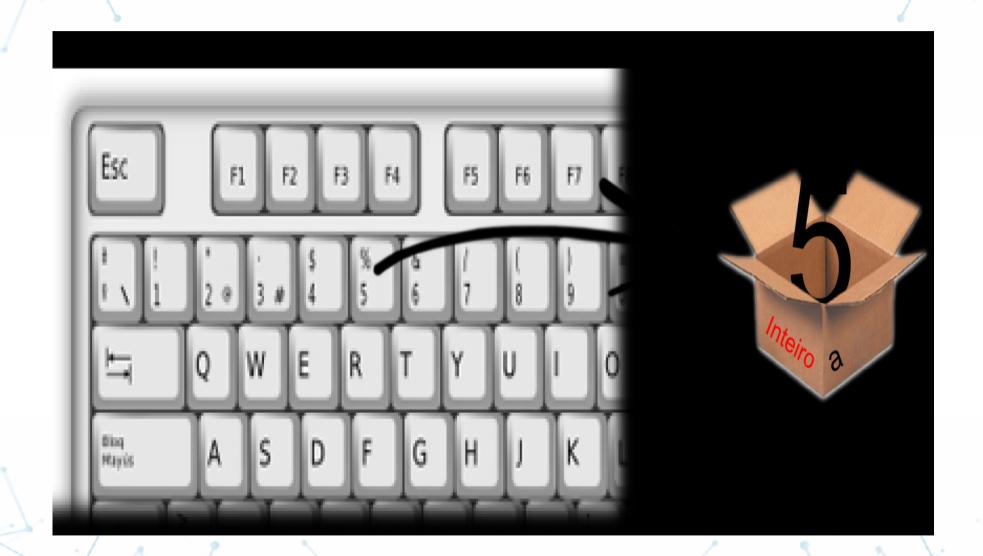








Comandos de Entrada de Dados









Qual a finalidade dos comandos de entrada de dados?





Qual a finalidade dos comandos de entrada de dados?

Atribuir conteúdo as variáveis através de informações oriundas do mundo externo.



















Qual o comando em python é utilizado para permitir que o usuário atribua conteúdo a uma variável?





Qual o comando em python é utilizado para permitir que o usuário atribua conteúdo a uma variável?

Input





Comando input

- Recebe dados ou valores que um usuário fornece através do teclado.
- Formato:

variável = input(pergunta)

- Na execução do comando input:
 - O programa pára e espera pela digitação de algum texto seguido do ENTER.
 - Quando o usuário digitar ENTER, o valor digitado é armazenado na variável.
 - pergunta é uma string opcional que será exibida para indicar o valor que usuário espera receber.





Comando input

Exemplo:

```
nome = input('Qual é o seu nome? ')
```

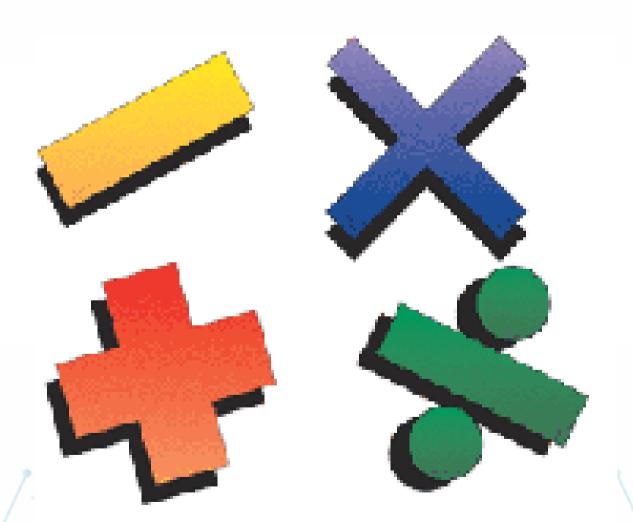
- O comando input sempre retorna um string e não um número.
- Para que o comando input retorne valores numéricos é necessário indicar o tipo de valor na leitura dos dados, antes do comando input:

```
cod=<u>int(</u>input('Digite o código do produto'))
vlr=<u>float(</u>input('Digite o valor do produto'))
```





Operadores Aritméticos







Operadores Aritméticos

Operação	Operador
Adição	+
Subtração	_
Multiplicação	*
Divisão	/
Potenciação	**
Parte inteira	//
Resto da divisão	%





Precedência Geral de Operadores Aritméticos

Ordem	Operação	Operador
1a	Parênteses	()
2a	Potenciação	**
3a	Multiplicação, Divisão, Resto e Divisão Inteira	*,/, % e //
4 a	Soma e Subtração	+ e -





Exemplos





Exemplos

```
>>> z=x*y
```

21

$$>>> z=x/y$$

2.33333333333333

343





Exemplos





Iniciando o Desenvolvimento de Programas.....







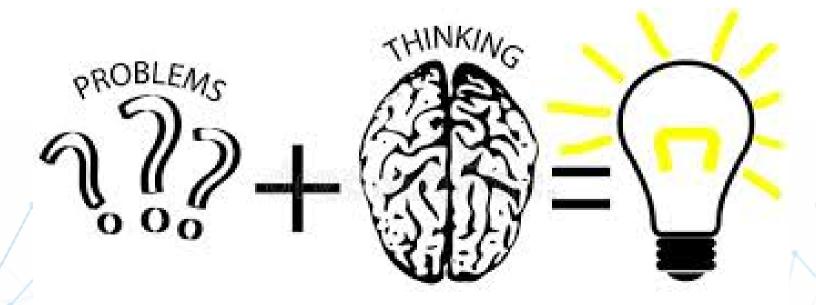
Como construir um programa?

O mais importante!

Pense sobre o problema a ser resolvido!

Não inicie construindo diretamente o programa!

Primeiro entenda o problema!







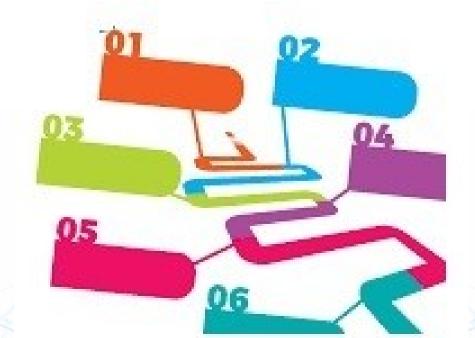
Como iniciar?

- 1.Leia o problema mais de uma vez.
- 2. Escreva qual o objetivo do problema, ou seja, o que o problema solicita que você faça.
- 3. Escreva como você resolveria o problema, sem pensar no programa, mas escrevendo um exemplo prático do que precisa ser feito no problema.
- 4. Escreva as fórmulas que devem ser utilizadas na resolução do problema.
- 5.Observando o exemplo prático, escreva quais são as informações necessárias para resolver este problema?
- 6.Destas informações:
 - a)quais são as informações que o problema não fornece, portanto o usuário deve informar?
 - b)quais são as informações que o problema já fornece (que são valores fixos)?



Durante a resolução do problema

- 1.Pare e revise cada passo para ver se você cometeu algum erro.
- 2.Releia o problema e verifique se você esqueceu alguma parte importante.
- 3. Mude de estratégia se você achar que se perdeu e não consegue chegar a lugar algum.







Após a resolução do problema

- 1.Pense em uma forma de verificar se a solução que você construiu está correta.
- 2.Revise tudo o que você fez para ter certeza que você não esqueceu nada.

3. Releia a descrição da tarefa e pergunte-se se a sua solução realmente alcança o objetivo proposto pela tarefa.





Exemplo

Faça um algoritmo que leia três notas de um aluno, calcule e escreva a média final deste aluno. Considerar que a média é ponderada e que o pesos das notas são: 2, 3 e 5, respectivamente.





O que o problema solicita?

Que seja calculada a média ponderada entre três notas com pesos 2, 3 e 5.







Como o problema é resolvido?

Multiplicando as notas pelos respectivos pesos e dividindo pela soma dos pesos.

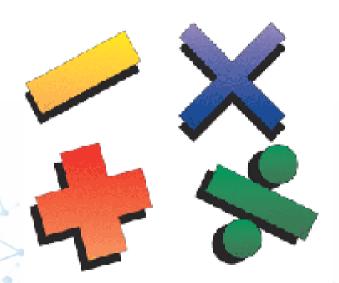
Supondo que a primeira nota seja 7, a segunda 8 e a terceira 8,5. O cálculo é feito da seguinte forma:

$$(7 * 2) + (8 * 3) + (8,5 * 5) = 8,05$$





Quais fórmulas devem ser utilizadas?







Quais são as informações necessárias para a resolução do problema?

- As três notas
- O peso de cada nota







Quais são as informações que o problema já fornece?

- O peso de cada nota (2, 3, 5):
 - Estes valores devem ser colocados diretamente no programa.
 - São constantes.





Quais são as informações que o problema não fornece, que o usuário deve informar?

- As três notas.
- Estes valores devem ser digitados pelo usuário, portanto devem ser solicitados através do comando de entrada de dados (input).







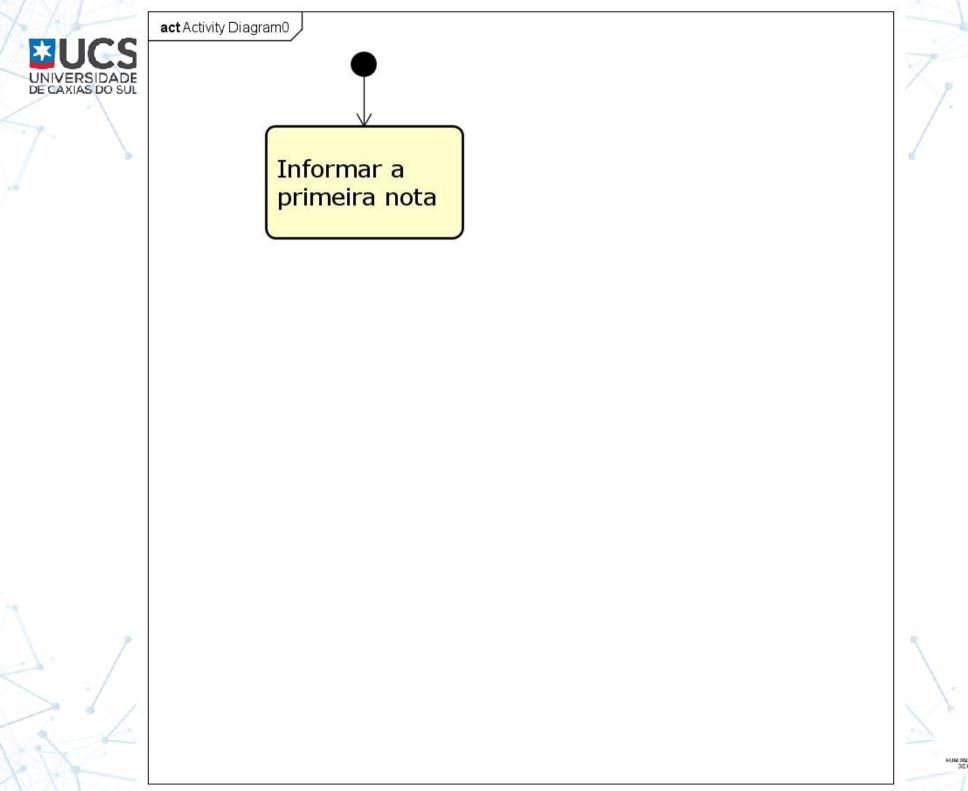




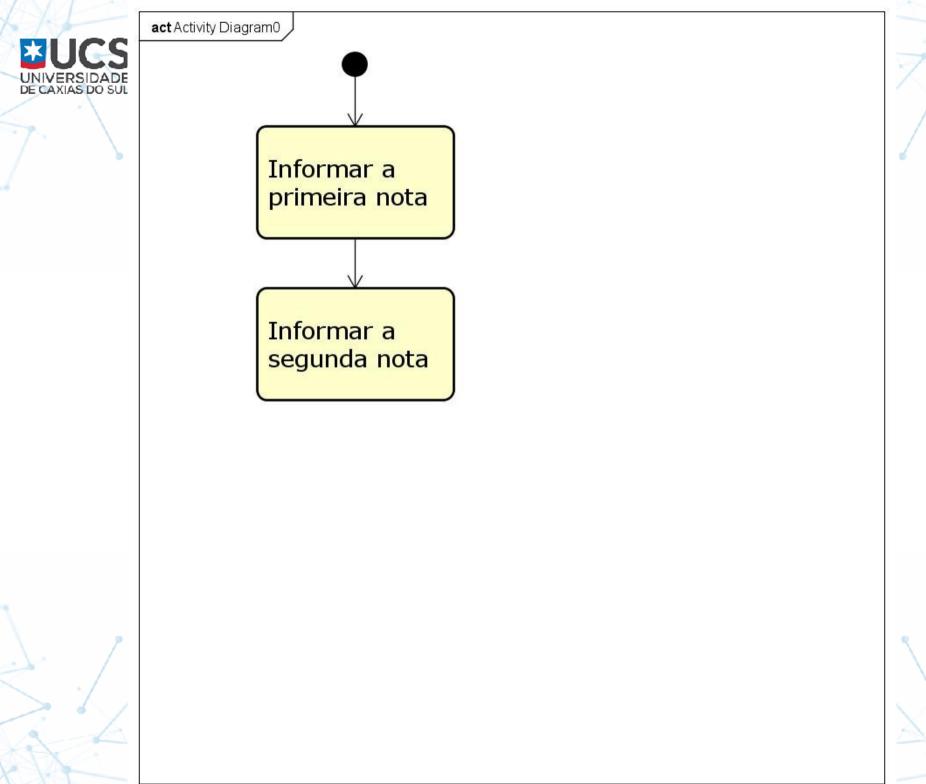
Programa

- Todo o programa tem três blocos essenciais:
 - Bloco de entrada: onde são informados os dados necessários para a resolução do problema, que o programa não tem, portanto o usuário deve digitar.
 - Bloco de processamento: onde devem ser executados os cálculos.
 - Bloco de saída: onde o resultados dos dados calculados devem ser mostrados para o usuário.



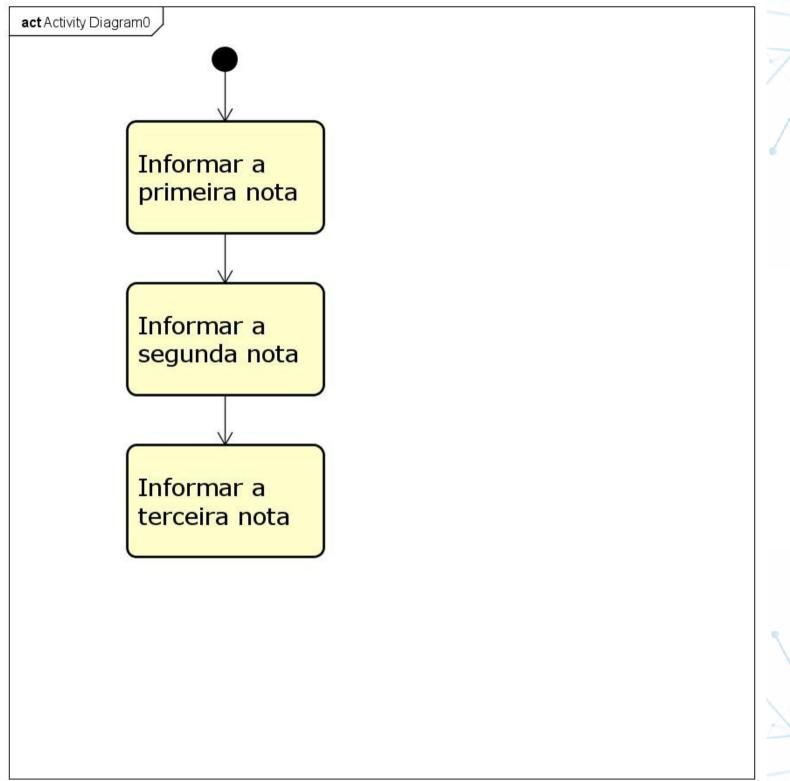






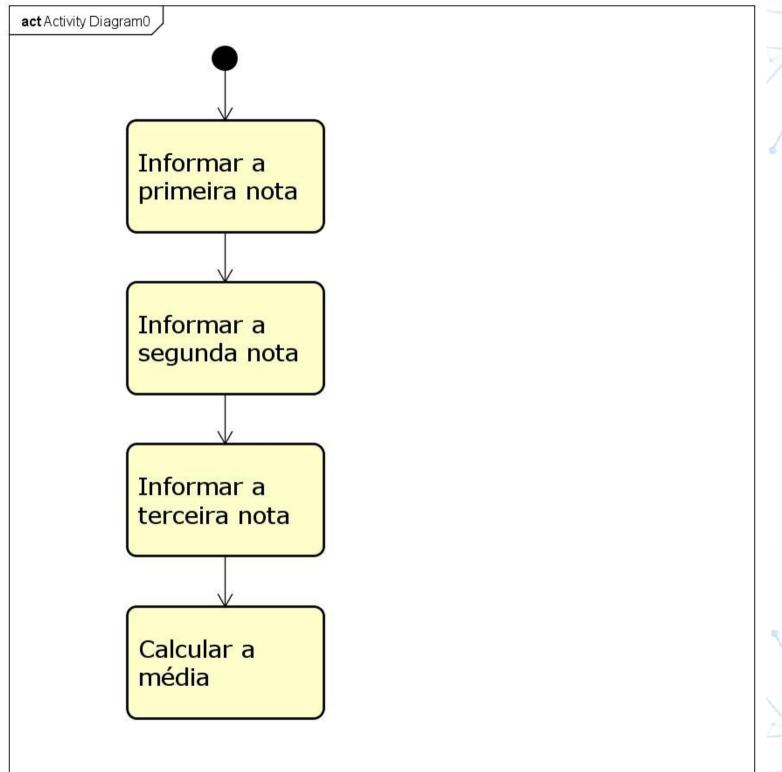






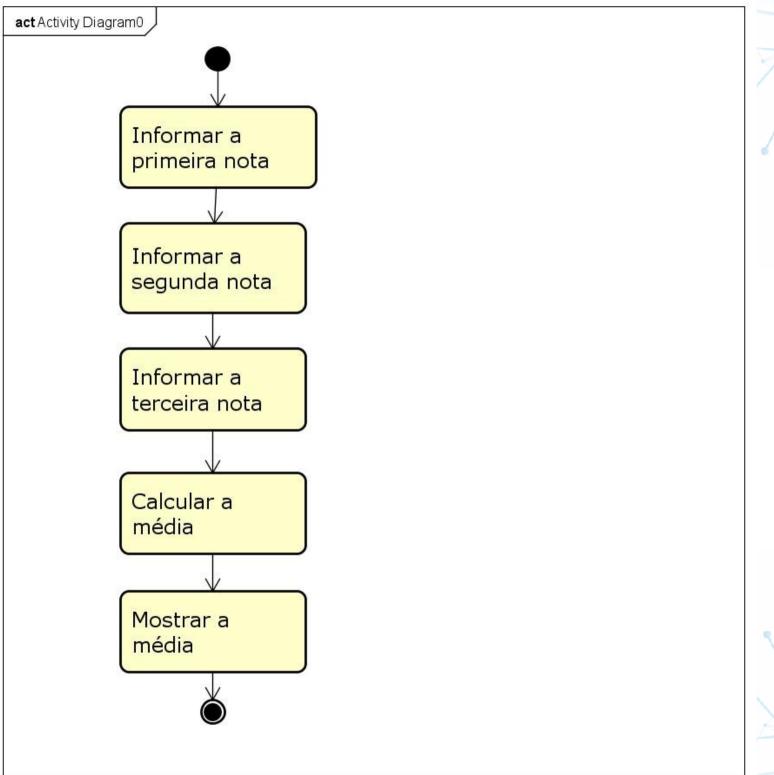
















Depois de sabermos como resolver o problema, utilizamos os comandos para desenvolver o programa!







