

1502

Competência (teoria) é você entender como foi feito e habilidade (prática) é saber reproduzir.

Lógica é interpretar (implementar) não deduzir.

Se algo ainda não funcionou não é porque você não tem competência, você ainda não sabe a habilidade certa.

Raciocínio é a estruturação de pensamentos de acordo com a lógica.

Onde há organização e ação, tem um sistema.

O método é a forma de fazer algo.

COMPETÊNCIAS

Conceitos de Lógica de Programação e construção de algoritmos; Princípios de programação; Paradigmas de programação; Conceitos de usabilidade de sistemas; Linguagens de programação e códigos fonte, objeto e arquivo executável; Comandos da linguagem de programação: Funções pré-definidas; Expressões e tabela da verdade; Tratamento de erros e exceções; Memória, tipos de dados e variáveis; Entrada, saída e conversão de tipos; Operadores aritméticos, relacionais e lógicos.

HABILIDADES

Implementar algoritmos de programação; Desenvolver sistemas aplicando princípios e paradigmas de programação; Elaborar algoritmos; Codificar programas, utilizando técnica de programação estruturada; Depurar (trocar o código que não funciona) e visionar (criar novas versões para documentos em gerar ou bater o olho e ver o erro e concertá-lo) programas, utilizando ambiente de desenvolvimento integrado.

2202

RACIOCINANDO A LÓGICA

O principal objetivo da lógica é raciocinar logicamente. É entender a resposta de resoluções de problemas e permitir que outras pessoas entendam suas propostas de resolução de problemas.

Ao tomar uma decisão, sempre use a forma mais lógica de tomá-la, coloque o pensamento em prática e faça ele se transformar em raciocínio. Tenha uma saída que beneficie você (e os outros).

Tudo que a gente faz no dia a dia é resultado de uma sequência ordenada, a lógica. Por menor que a tarefa seja, nós ordenamos esses passos para torná-los mais rápidos de se fazer e mais eficientes. Ao se preparar antes para algo, aquilo se torna mais fácil de fazer.

“A lógica sempre nos acompanha! Quando falamos, escrevemos ou fazemos alguma ação estamos pensando de forma ordenada e sequenciada para que as coisas aconteçam de forma correta. Logo a lógica consiste em colocar “ordem no nosso pensamento”. Quanto mais próximo da necessidade mais próximo dessa forma correta. A lógica são as ações mais coerentes para chegar num objetivo, defende o uso da razão. Não existe uma única forma de executar algo.”

Você tem que pensar em como fazer algo antes de realizá-lo. E toda a ação que acontece é uma variação de uma ação que aconteceu antes.

A máquina precisa ser programada para entender o que a gente está querendo dizer, e para isso nós vamos “conversar” com ela através de um codificador. Aí entram as linguagens de codificação como Java, Phyton etc.

Em Informática, instrução é uma ordem ou um comando que indica ao computador uma ação elementar a ser executada.

Algoritmo: Conjunto de regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema por meio de um número finito de etapas. Ou seja, o algoritmo pode ser considerado como um rascunho. O que está escrito nele é um código (algoritmo se escreve em PT e codifica em ING).

Código: É tudo aquilo que você faz, é uma representação.

O código está dentro do algoritmo e o algoritmo é formado por códigos.

0103

Algoritmo: É um conjunto de regras, um passo a passo de uma sequência lógica para fazer uma tarefa/resolver um problema (você coloca como você entendeu aquele problema e passa para a máquina seguir).

Aqueles 10 passos são algoritmos e o caminho que a máquina vai seguir para resolver é o que você vai colocar para ela (o que você acha que é mais lógico e rápido). A máquina só vai fazer se você pensar por ela, já que ela só executa o que você pensou, logo ela tem que entender o que você escreveu.

Sequência de Fibonacci: sequencia matemática (século XII). Representa uma sequência e ela é encontrada na natureza, em copas das arvores, ramos de uma planta e nas pétalas das flores. É uma sequência de números, mas ela não precisa estar na ordem que a gente conhece, mas tem uma sequência correta a ser seguida (sucessão de números/escritas).

Cadastro cliente

Cpf

Endereço

Cadastro de compras

Descrição

Valor

Cadastro produto

Descrição

quantidade|

 **POR QUE APRENDER?**

A lógica para programação é o processo de saber pensar na mesma sequência em que o computador executa as tarefas.

Aprende-se a imaginar como as ações serão executadas partindo-se do estudo de um problema até chegar à sua solução por meio da construção de um algoritmo. Por isso, a lógica de programação é tão importante! Sendo assim, esta agenda traz conceitos para você compreender a lógica de programação, conhecer e desenvolver algoritmos e fluxogramas; interpretar algoritmos, pseudocódigo e outras especificações para codificar programas.

Observe a figura acima e pense em trocar a ordem de algum dos passos ilustrados. Por exemplo:

1 – Tomar conteúdo

2 – Abrir a lata

3 – Pegar o recipiente

Será que é possível beber o refrigerante da lata?

Claro que não! É preciso seguir a ordem dos passos ilustrados acima para que consigamos atingir o objetivo final, que neste caso, é tomar o conteúdo.

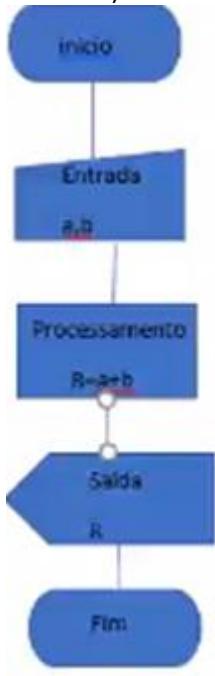
A mesma coisa acontece com o computador. Quando pensamos em fazer um programa no computador, temos que ter em mente que a máquina desconhece totalmente alguns conceitos que para nós são muito óbvios. Por isto devemos descrever cada passo, por mais simples que seja, para que haja uma sequência lógica em nossa programação e assim, o computador executar todas as instruções necessárias para uma determinada tarefa. Este é o objetivo desta agenda!

Entender a necessidade de descrever com exatidão cada uma das tarefas a serem executadas pela máquina.

Então, vamos começar assistindo ao vídeo – Conceitos Básicos da Lógica de Programação, disponível em:

<https://www.youtube.com/watch?v=vAlBmbCcmD4> - 8:15

É assim que representa o algoritmo e coloca a ação dentro do processamento (as letras são as variáveis):



SÍMBOLO		NOME E FUNÇÃO
[rectangle]		NOME = TERMINAL FUNÇÃO = indica INÍCIO ou FIM de um processamento
[rectangle]		NOME = PROCESSAMENTO FUNÇÃO = definição de variáveis ou processamentos em geral (cálculos)
[trapezoid]		NOME = ENTRADA MANUAL FUNÇÃO = entrada de dados via teclado, idêntico ao comando LEIA.
[oval]		NOME = DISPLAY FUNÇÃO = saída de dados, mostra um texto e/ou variável na tela, idêntico ao comando ESCREVA
[parallelogram]		NOME = DOCUMENTO FUNÇÃO = saída de dados, envia um texto e/ou variável para a impressora, usado em relatórios, idêntico ao comando IMPRIMA
[diamond]		NOME = DECISÃO FUNÇÃO = decisão a ser tomada, retornando verdadeiro ou falso, idêntico ao comando SE



Figura 05 - Algoritmo com os passos da latinha

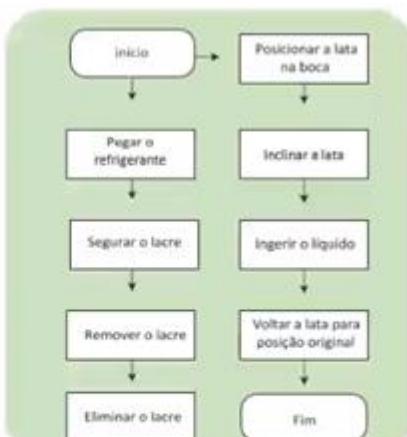


Figura 05.1 - Algoritmo detalhado da latinha

Observe que, à esquerda, temos um fluxograma resumido que representa os passos necessários para que alguém tome um refrigerante em uma lata: 1. pegar a latinha; 2. abrir a tampa; 3. tomar o refrigerante.

A partir dessa mesma situação, é possível inserir mais operações em nosso algoritmo, deixando-o mais detalhado. Veja a imagem à direita - Algoritmo detalhado da latinha.

Pseudocódigo: É uma escrita mais próxima da Linguagem de Programação, ou seja, não usaremos nenhuma informação técnica da Linguagem, apenas utilizaremos o nosso idioma (português) escrevendo mais próximo das instruções computacionais. Muitos autores chamam o Pseudocódigo de “Portugol” ou “Português Estruturado” devido a estas características.

A seguir um pequeno modelo, como orientação, para escrever um Pseudocódigo:

Programa nome_do_programa Declare <i>(declaração de variáveis)</i> Início <i>(instruções e comandos)</i> Fim	Título do Algoritmo Declarções de variáveis Comandos e instruções ordenados de forma lógica
---	--

Vídeos auxiliares:

[Lógica de programação - Aula 01 - Introdução](#)

[Informática - M1 - Agenda 10 | C005.1.03A10AR](#)

[Lógica de programação - Aula 02 - Tipos de algoritmo](#)

[Lógica de Programação - Conceitos Iniciais](#)

1503

Algoritmo: algo que é abstrato e vai se tornar real. Abstração é tirar do pensamento (pensar em algo) e colocar no papel (tornar concreto). O papel da planta de uma casa é concreto, mas a planta em si é abstrata.

Variável: número 30 é o Renato. Toda vez que aparecer o 30, vai se pensar no Renato, mas se aparecer 29, não vai ter nome. O que muda é o número, e não o nome. Aquilo vai variar de acordo como você nomeou aquilo.

Dentro de um intervalo de 1 a 10, tem a variante das 10 posições ($1+9 = 10$) e depende de qual ela ocupa. Pode ser a 1,2,3,4...

2203

Descrição Narrativa
<p>Algoritmo para converter real em dólar:</p> <ol style="list-style-type: none">1. Informar o valor disponível em reais para comprar dólares.2. Informar a cotação do dólar do dia.3. Dividir o valor disponível pela cotação do dia.4. Exibir o resultado do cálculo para o usuário.  An illustration at the bottom shows a circular coin on the left with a dollar sign (\$) and a portrait of a person. An arrow points from the coin to a US dollar bill on the right, which also features a portrait of a person. This visual metaphor represents the process of converting one currency into another.

Pseudocódigo

```
Algoritmo Conversao_real_dolar
// rotina que converte um determinado valor em reais em
dólares
Var
    VALOR_REAL, COTACAO_DOLAR, CONVERSAO;
real
    VALOR_REAL = 0
    COTACAO_DOLAR = 0
    CONVERSAO = 0
Início
    Escreva "Programa para converter reais em
    dólares"
    Escreva "Informe o valor disponível em reais (para
    comprar dólares): R$"
    Leia VALOR_REAL
    Escreva "Informe o valor do dólar em reais (cotação
    do dia): R$"
    Leia COTACAO_DOLAR
    CONVERSAO = VALOR_REAL / COTACAO_DOLAR
    Escreva "Com essa quantia será possível comprar:
    US$"
    Escreva CONVERSAO
    Escreva "Boa viagem!"
Fim
```

// É comentário

```
CONVERSAO = 0
Início
    Escreva "Programa para converter reais em
    dólares"
    Escreva "Informe o valor disponível em reais (para
    comprar dólares): R$"
    Leia VALOR_REAL
    Escreva "Informe o valor do dólar em reais (cotação
    do dia): R$"
    Leia COTACAO_DOLAR
```

```
CONVERSAO      =      VALOR_REAL      /
COTACAO_DOLAR
    Escreva "Com essa quantia será possível comprar:
    US$"
    Escreva CONVERSAO
    Escreva "Boa viagem!"
Fim
```

Descrição Narrativa versus Pseudocódigo

Algoritmo para converter real em dólar:

1. **Informar** o valor disponível em reais para comprar dólares.
2. **Informar** a cotação do dólar do dia.
3. **Dividir** o valor disponível pela cotação do dia.
4. **Exibir** o resultado do cálculo para o usuário.

```

Algoritmo Conversao_real_dolar
// rotina que converte um determinado valor em reais e
m dólares
Var
    VALOR_REAL, COTACAO_DOLAR, CONVE
RSAO: real
    VALOR_REAL = 0
    COTACAO_DOLAR = 0
    CONVERSAO = 0
Início
    Escreva "Programa para converter reais em
dólares"
    Escreva "Informe o valor disponível em reais
(para comprar dólares): R$"
    Lêa VALOR_REAL

```

Apresentaremos, a seguir, a forma básica de um algoritmo em português estruturado:

```

Algoritmo <nome_do_algoritmo>
// <explicação resumida do que a rotina irá fazer>
Var
    <declaração das variáveis e do tipo de informaç
ão que elas receberão>
Início
    <instruções e processamentos encadeados de f
orma lógica>
Fim

```

Todo algoritmo começa com a definição de um nome de identificação para ele, na primeira linha:
<nome_do_algoritmo>

Na linha de baixo, deve ser feita a declaração de variáveis, cuja palavra reservada é **Var**. O termo **Var** é uma redução da palavra "variável". Nesse sentido, o termo **Var** é um marcador, assim como as palavras INÍCIO, FIM, SE, SENÃO, ENTÃO, FAÇA CASO etc. Um marcador ou uma palavra reservada são termos interpretados de forma diferenciada pelos programas. Por isso, não devem ter outros usos, como nomear variáveis.

ATIVIDADE

1. Escreva aqui um programa em Portugol e utilizando o Word faça um Fluxograma, que leia dois valores A e B, calcule a soma entre eles e exiba para o usuário.
2. Desenvolva aqui um programa em Portugol e utilizando o Word faça um fluxograma que leia o valor do lado de um quadrado, calcule o perímetro e exiba para o usuário.
3. Faça aqui um programa em Portugol e utilizando o Word faça um fluxograma, que leia um número inteiro, calcule o seu quadrado e exiba o resultado para o usuário.

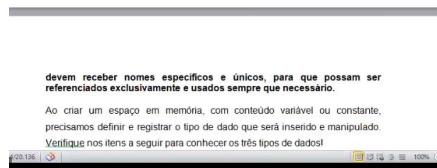
4.Sabendo que a formula para calcular a área de uma circunferência é $A = 3,14 * (R * R)$, elabore aqui um programa em Portugol e utilizando o Word faça um fluxograma que leia o valor do raio da circunferência, calcule a área e exiba o resultado para o usuário.

0504

2.5 - Tipos de Dados

Programar computadores requer a definição e o uso de espaços em memória para receber informações inseridas pelos usuários. Os espaços em memória recebem informações que podem variar no decorrer de um processamento. Por isso, é muito importante aprender a criar e definir o tipo, bem como manipular variáveis e constantes.

Em linhas gerais, variáveis são espaços em memória que servem para guardar um tipo de dado ou informação. Esses espaços em memória



Quais as características que a variável terá. Palavra = caractere

Tipo de dado = aquilo q atribui alguma coisa = alfa numérico (inteiro, fracionário, real...)

VARIÁVEL = número: 1 2 3 4 5 6 7 8 9 ... → tem q declarar para a máquina entender:

Var

num1, num2, num3: inteiro; num1=32 num2=45 num3=70

Leia num1;

Leia num2;

Leia num3;

Esse num1 não é necessariamente o número 1, é qualquer número mas tem q ser inteiro e se colocar um número com vírgula o computador não vai ler, vai ser incompatível

Var

Num11, num12, num13: real num11=1,2 num12=32,5 num3=70,45

Leia num11;

Leia num12;

Leia num13;

Ao criar um espaço em memória, com conteúdo variável ou constante, precisamos definir e registrar o tipo de dado que será inserido e manipulado.

Verifique nos itens a seguir para conhecer os três tipos de dados!

2.6 - Literais

I

Dados literais – ou caracteres – são sequências contendo letras, números e outros símbolos especiais. Uma sequência de caracteres deve ser indicada entre aspas (""). Esse tipo de dado é conhecido também como Alfanumérico, String, Literal ou Cadeia. Como exemplos, temos: "Fundação Bradesco", "Técnico em Desenvolvimento de Sistemas", "84", "843.48", entre outros numéricos.

2.7 - Numéricos Reais

Caractere: a,b,c,d,e,f,g,h.... 1.2.3.4.5.6.7.8... → I[a em cima é número porque ele foi declarado como inteiro

Se tivesse declarado inteiro e fizesse $32,55 * 5$ ia ter o resultado de 162,55, mas ele ia arredondar para 163 porque passou do 5. Se der 162,55555555555555 infinito ele vai continuar 162 e se tiver 162,5555555556 vai ser 163

Não existe acento na codificação, Técnico = Tecnico.

2.6 - Literais

I

Dados literais – ou caracteres – são sequências contendo letras, números e outros símbolos especiais. Uma sequência de caracteres deve ser indicada entre aspas (""). Esse tipo de dado é conhecido também como Alfanumérico, String, Literal ou Cadeia. Como exemplos, temos: "Fundação Bradesco", "Técnico em Desenvolvimento de Sistemas", "84", "843.48", entre outros numéricos.

2.7 - Numéricos Reais

Caractere: a,b,c,d,e,f,g,h.... 1.2.3.4.5.6.7.8... → I[a em cima é número porque ele foi declarado como inteiro

Se tivesse declarado inteiro e fizesse $32,55 * 5$ ia ter o resultado de 162,55, mas ele ia arredondar para 163 porque passou do 5. Se der 162,55555555555555 infinito ele vai continuar 162 e se tiver 162,5555555556 vai ser 163

Não existe acento na codificação, Técnico = Tecnico.

2.7 - Números Reais

Caractere: a,b,c,d,e,f,g,h... 1.2.3.4.5.6.7.8... → [a em cima é número porque ele foi declarado como inteiro

Se tivesse declarado inteiro e fizesse $32.55 * 5$ ia ter o resultado de 162.55, mas ele ia arredondar para 163 porque passou do 5. Se der 162,55555555555555 infinito ele vai continuar 162 e se tiver 162,5555555556 vai ser 163

Não existe acento na codificação. Técnico = Técnico.

2.9 - Uso e Funções de Constantes e Variáveis

Já mencionamos algumas vezes as palavras "constante" e "variável". Você deve estar curioso para saber melhor o que elas significam, não é mesmo? Agora, vamos definir cada uma delas e dar exemplos de suas funções

Chamamos de constante todo valor fixo que não será alterado durante a execução de um programa.

Conforme o contexto de utilização, as constantes podem ser classificadas como:

Numéricas: 1, 0, 3, 9, 5, 4, 6, 8, 2
Literais: “”, “ ”
Lógicas: ✓, ✗

2.7 - Números Reais

Caractere: a,b,c,d,e,f,g,h... 1.2.3.4.5.6.7.8... → [a em cima é número porque ele foi declarado como inteiro

Se tivesse declarado inteiro e fizesse $32.55 * 5$ ia ter o resultado de 162.55, mas ele ia arredondar para 163 porque passou do 5. Se der 162,55555555555555 infinito ele vai continuar 162 e se tiver 162,5555555556 vai ser 163

Não existe acento na codificação. Técnico = Técnico.

Ferramentas de Imagens

Conforme o contexto de utilização, as constantes podem ser classificadas como:

Numéricas: 1, 0, 3, 9, 5, 4, 6, 8, 2
Literais: “”, “ ”
Lógicas: ✓, ✗

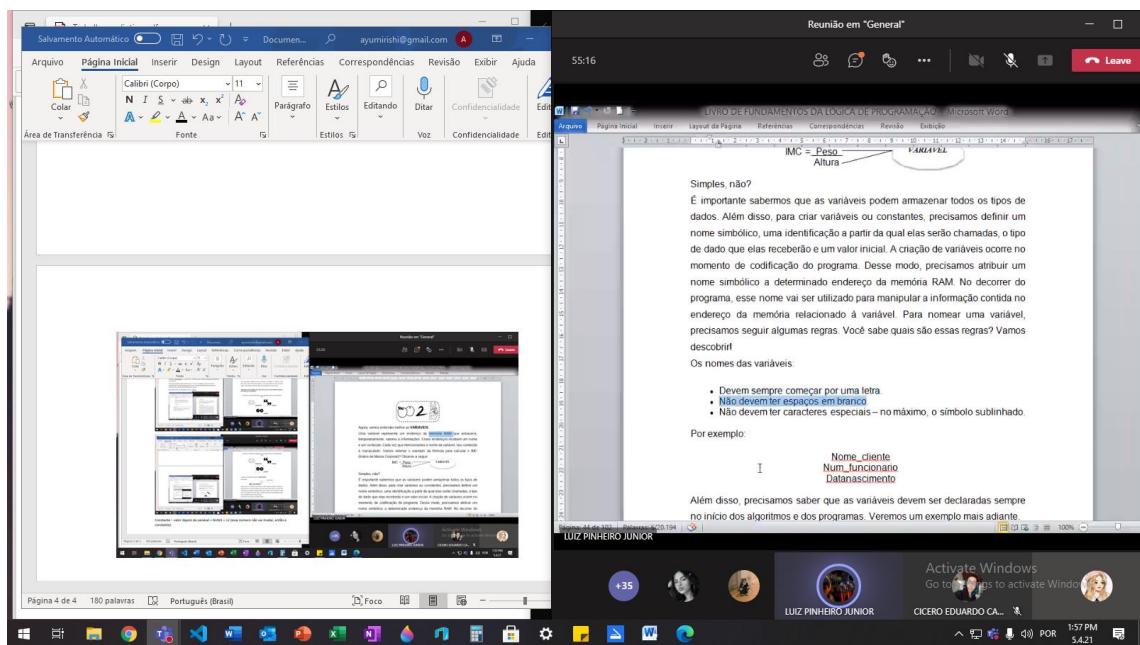
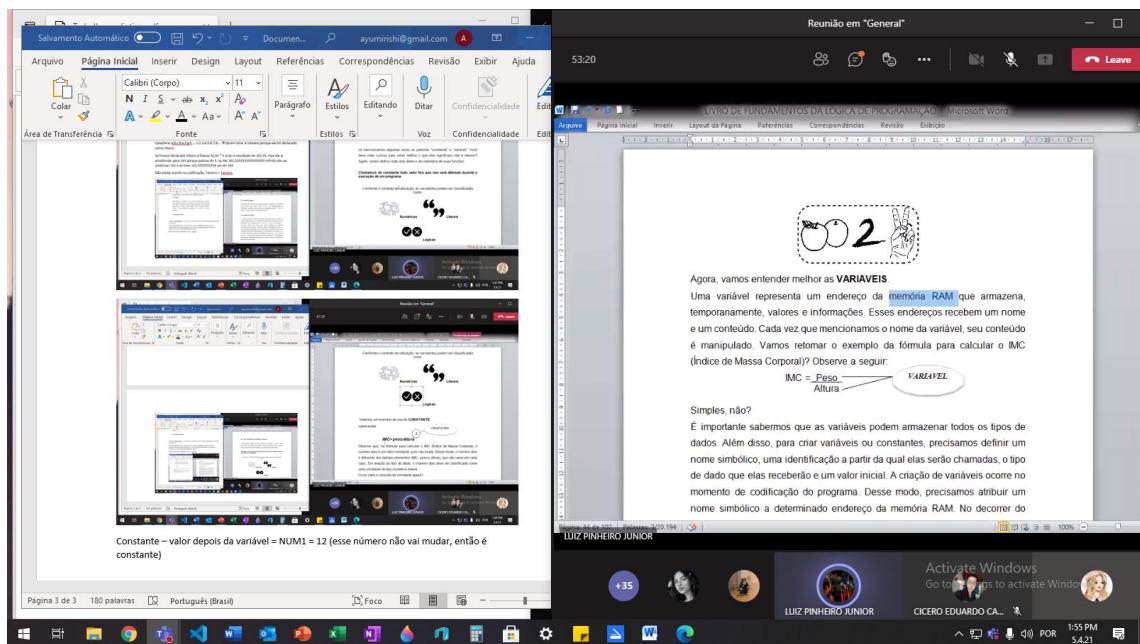
Vejamos um exemplo de uso de **CONSTANTE**

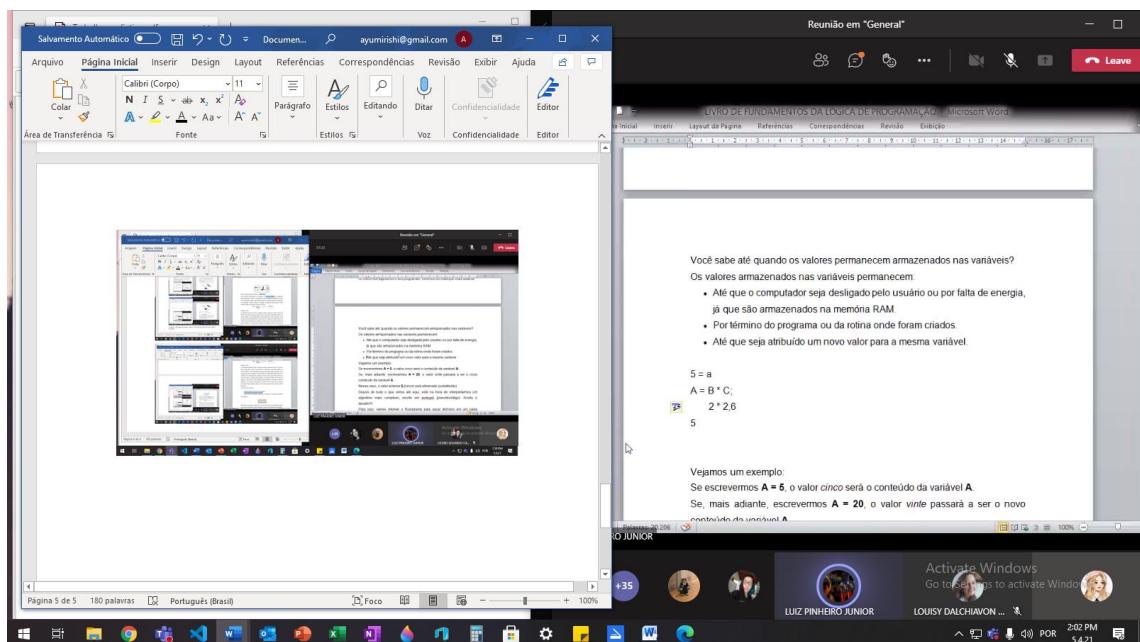
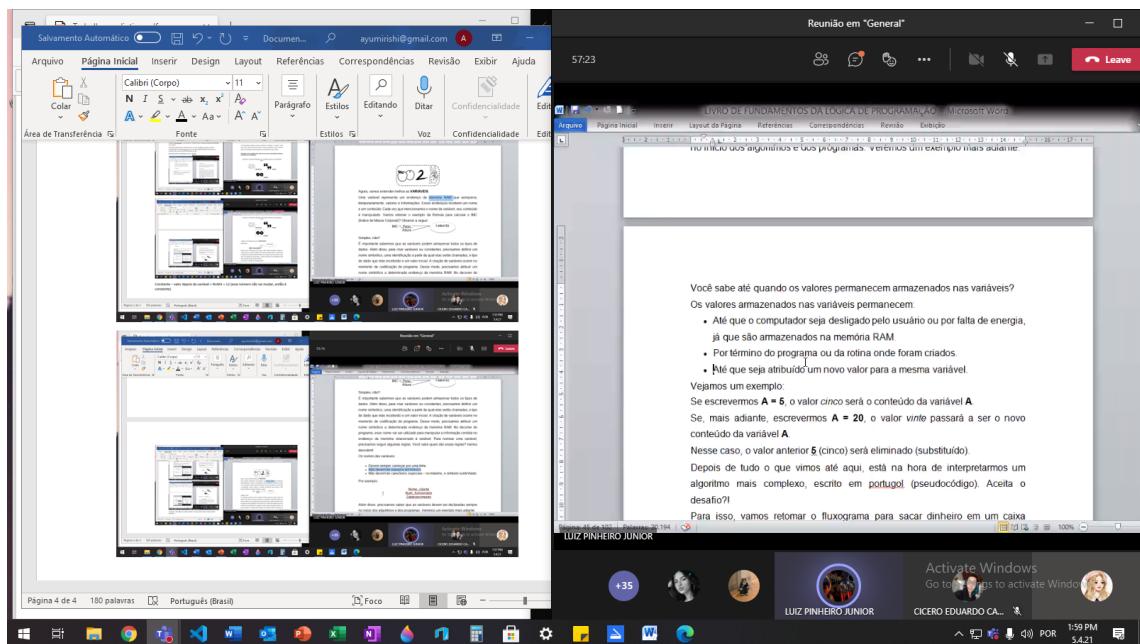
CONSTANTES: IMC = peso / altura
CONSTANTE: 2

Observe que, na fórmula para calcular o IMC (Índice de Massa Corporal), o número dois é um valor constante, pois não muda. Desse modo, o número dois é diferente dos demais elementos (IMC, peso e altura), que vão variar em cada caso. Em relação ao tipo do dado, o número dois deve ser classificado como uma constante do tipo numérica inteira.

Ficou claro o conceito de constante agora?

Constante – valor depois da variável = NUM1 = 12 (esse número não vai mudar, então é constante)





Divertimentos:
Circo: extremamente afeto aos divertimentos, foi de Roma que se originou o jogos circenses e corridas de carros, jogos de Tróia, torneios a cavalo entre cavaleiros conduzidos por escravos

Dois círcos romanos, o mais célebre é o "Circus Maximus"



Clique para adicionar anotações

Reunião em "General" 01:00:56

VEJAMOS UM EXEMPLO:
Se escrevemos **A = 5**, o valor cinco será o conteúdo da variável **A**. Se, mais adiante, escrevemos **A = 20**, o valor vinte passará a ser o novo conteúdo da variável **A**. Nesse caso, o valor anterior **5** (cinco) será eliminado (substituído). Depois de tudo o que vimos até aqui, está na hora de interpretarmos um algoritmo mais complexo, escrito em português (pseudocódigo). Aceita o desafio?!

Para isso, vamos retomar o **fluxograma** para sacar dinheiro em um caixa eletrônico. Agora, veja como essa sequência poderia ficar, com alguns ajustes, em português. Veja a sequência lógica representada de três formas diferentes:

- Descrição narrativa
- Fluxograma
- Pseudocódigo

Descrição narrativa versus fluxograma versus pseudo

Português

```

Algoritmo Saque_dinheiro_caxa_eletronico
// rotina para sacar dinheiro em um caixa eletrônico
Vai
VALOR_SAQUE SALDO real
SENHA_SAQUE NUM_AGENCIA_NUN_CONTA inteiro

```

LUIZ PINHEIRO JUNIOR CICERO EDUARDO CA... 2:03 PM 5421

Salvamento Automático Salvo automaticamente em: ayumirishi@gmail.com

Área de Transferência

Área de Transferência

Divertimentos:
Circo: extremamente afeto aos divertimentos, foi de Roma que se originou o jogos circenses e corridas de carros, jogos de Tróia, torneios a cavalo entre cavaleiros conduzidos por escravos

Dois círcos romanos, o mais célebre é o "Circus Maximus"



Clique para adicionar anotações

Reunião em "General" 01:01:15

VEJAMOS UM EXEMPLO:
Se escrevemos **A = 5**, o valor cinco será o conteúdo da variável **A**. Se, mais adiante, escrevemos **A = 20**, o valor vinte passará a ser o novo conteúdo da variável **A**. Nesse caso, o valor anterior **5** (cinco) será eliminado (substituído). Depois de tudo o que vimos até aqui, está na hora de interpretarmos um algoritmo mais complexo, escrito em português (pseudocódigo). Aceita o desafio?!

Para isso, vamos retomar o **fluxograma** para sacar dinheiro em um caixa eletrônico. Agora, veja como essa sequência poderia ficar, com alguns ajustes, em português. Veja a sequência lógica representada de três formas diferentes:

- Fluxograma
- Pseudocódigo

Descrição narrativa versus fluxograma versus pseudo

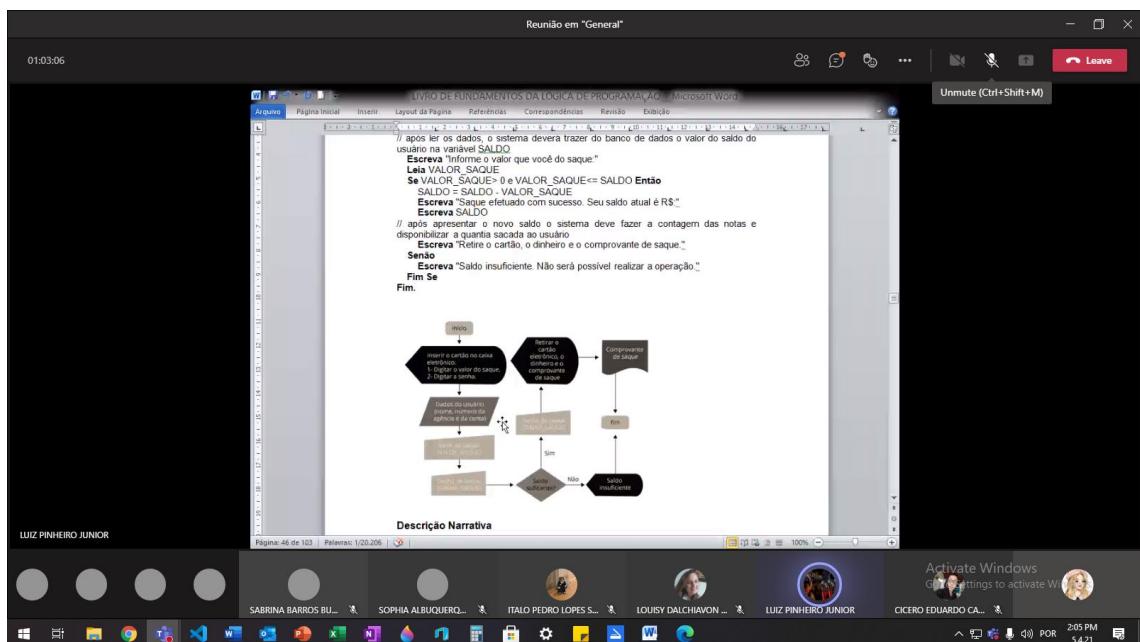
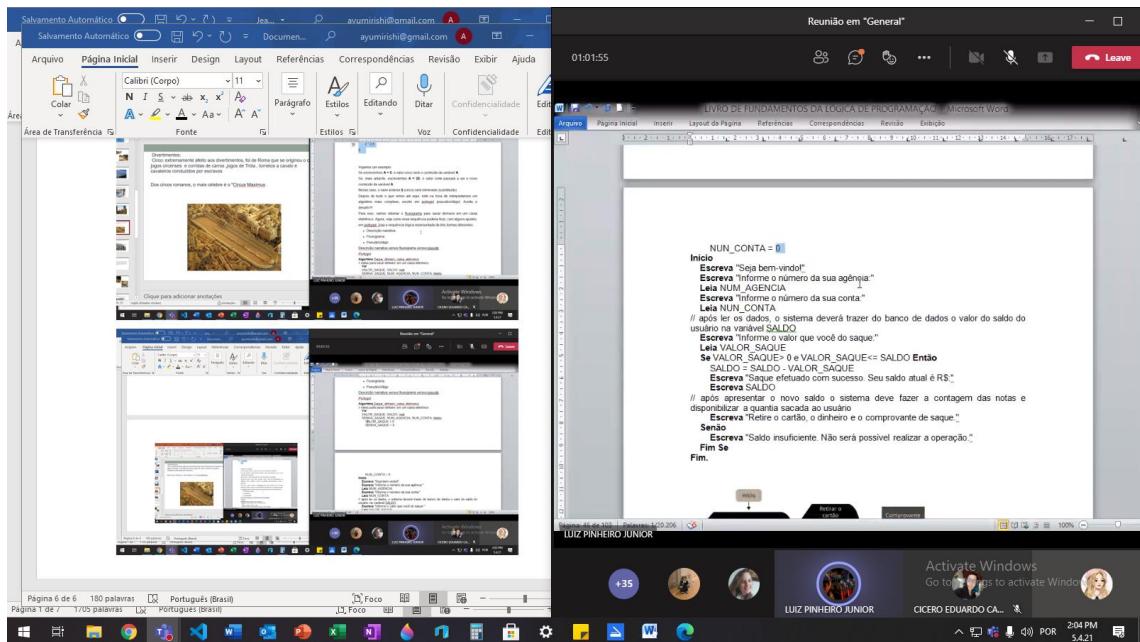
Português

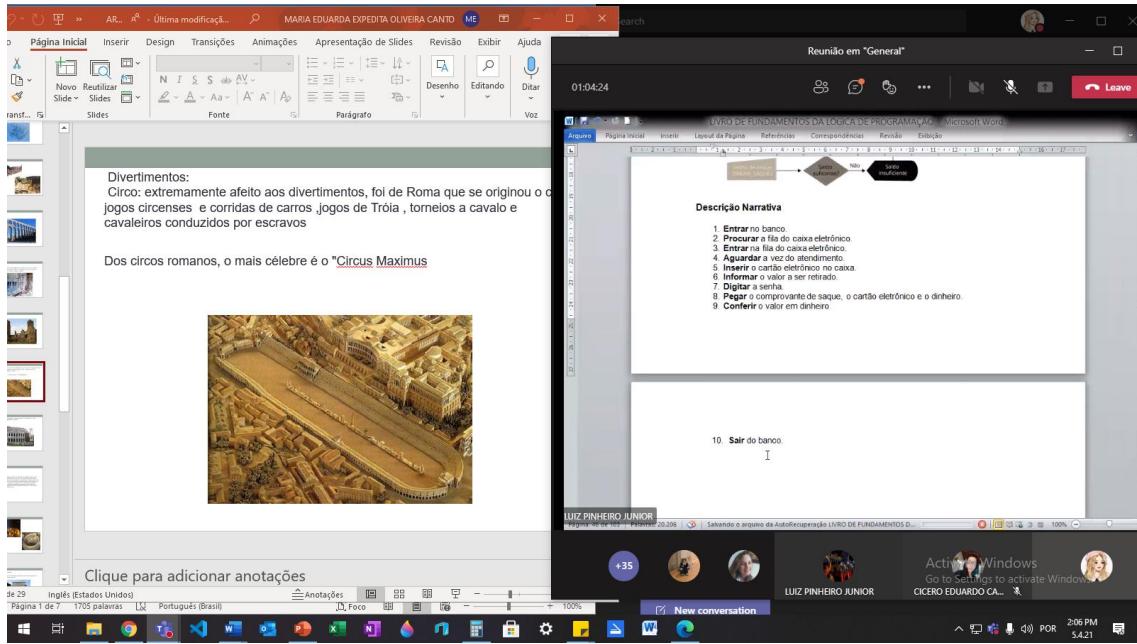
```

Algoritmo Saque_dinheiro_caxa_eletronico
// rotina para sacar dinheiro em um caixa eletrônico
Vai
VALOR_SAQUE SALDO real
SENHA_SAQUE NUM_AGENCIA_NUN_CONTA inteiro
VALOR_SAQUE = 0
SENHA_SAQUE = 0

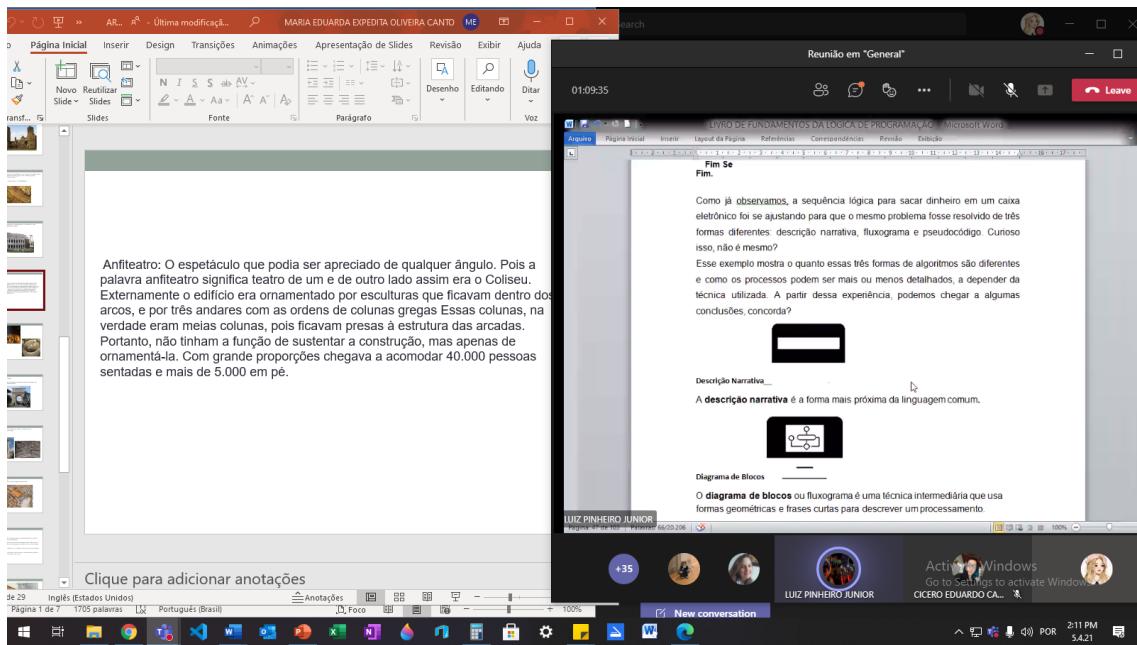
```

LUIZ PINHEIRO JUNIOR CICERO EDUARDO CA... 2:03 PM 5421





A narrativa é mais proxima do nosso entendimento e o algoritmo é mais próximo da linguagem de máquina



Anfiteatro: O espetáculo que podia ser apreciado de qualquer ângulo. Pois a palavra anfiteatro significa teatro de um é de outro lado assim era o Coliseu. Externamente o edifício era ornamentado por esculturas que ficavam dentro dos arcos, e por três andares com as ordens de colunas gregas. Essas colunas, na verdade eram meias colunas, pois ficavam presas à estrutura das arcadas. Portanto, não tinham a função de sustentar a construção, mas apenas de ornamentá-la. Com grande proporções chegava a acomodar 40.000 pessoas sentadas e mais de 5.000 em pé.

Descrição Narrativa
A descrição narrativa é a forma mais próxima da linguagem comum.

Diagrama de Blocos
O diagrama de blocos ou fluxograma é uma técnica intermediária que usa formas geométricas e frases curtas para descrever um processamento.

Pseudocódigo
O português estruturado ou pseudocódigo é a forma mais detalhada e mais próxima dos códigos de programação.

As três técnicas são úteis e importantes. Dessa forma, caberá a você definir qual é a mais apropriada, tendo em vista o contexto em que estiver planejando um programa. Boa escolha!

De acordo com Oliveira (2004), "há diversas formas de representação de algoritmos que diferem entre si pela quantidade de detalhes de implementação que fornecem ou, inversamente, pelo grau de abstração que possibilitam com relação à implementação do algoritmo em termos de uma linguagem de programação específica".

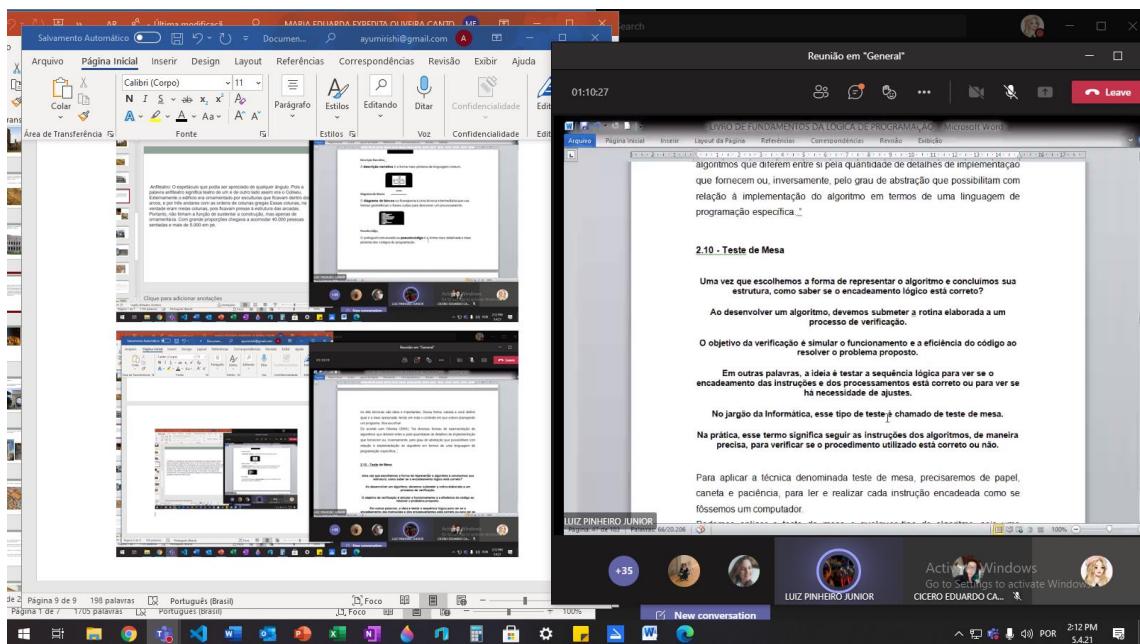
2.10 - Teste de Mesa

Uma vez que escolhemos a forma de representar o algoritmo e concluímos sua estrutura, como saber se o encadeamento lógico está correto?

Ao desenvolver um algoritmo, devemos submeter a rotina elaborada a um processo de verificação.

O objetivo da verificação é simular o funcionamento e a eficiência do código ao resolver o problema proposto.

Em outras palavras, a ideia é testar a sequência lógica para ver se o encadeamento das instruções e dos processamentos está correto ou não. Ver se



1204

2.10 - Teste de Mesa

Uma vez que escolhemos a forma de representar o algoritmo e concluímos sua estrutura, como saber se o encadeamento lógico está correto?

Ao desenvolver um algoritmo, devemos submeter a rotina elaborada a um processo de verificação.

O objetivo da verificação é simular o funcionamento e a eficiência do código ao resolver o problema proposto.

Em outras palavras, a ideia é testar a sequência lógica para ver se o encadeamento das instruções e dos processamentos está correto ou para ver se há necessidade de ajustes.

No jargão da Informática, esse tipo de teste é chamado de teste de mesa.

Na prática, esse termo significa seguir as instruções dos algoritmos, de maneira precisa, para verificar se o procedimento utilizado está correto ou não.

Para aplicar a técnica denominada teste de mesa, precisaremos de papel, caneta e paciência, para ler e realizar cada instrução encadeada como se fôssemos um computador.

Podemos aplicar o teste de mesa a qualquer tipo de algoritmo, seja uma descrição narrativa, um fluxograma ou um pseudocódigo.

Vamos descobrir como aplicar esse teste na prática? Mas antes, leia a tirinha para descontrair um pouco. Para demonstrar o teste de mesa, partiremos do algoritmo **Quatro_operações_básicas**, representado no pseudocódigo a seguir. Esse algoritmo foi construído com o objetivo de solicitar ao usuário a inserção de dois números quaisquer para a realização das quatro operações matemáticas básicas. Preparado? Então vamos lá!

Para testar algoritmos como esse, que conta com a inserção de dados, processamentos e exibição de resultados, o primeiro passo é criar uma tabela, colocando, em cada coluna, o nome das variáveis declaradas.

Veja, a seguir, a tabela criada para nosso exemplo:

TESTE	NUM_A	NUM_B	SOMA	SUBTRACAO	MULTIPLICACAO	DIVISAO
1º	(30)	(60)	90	-30	1800	0,5
2º	(-20)	(40)	20	-60	-800	0,5
3º	(20)	(-40)	-20	60	-800	0,5

Observe que tabela prevê a realização de três testes. No entanto, esse número pode variar de acordo com o contexto e a complexidade do algoritmo. Depois de montada a tabela, é preciso ler atentamente cada instrução a fim de verificar o encadeamento das ações, a clareza das instruções e a escrita dos processamentos.

Lembre-se de que a proposta do teste de mesa é simular a realização da rotina do algoritmo por meio do preenchimento da tabela.

Tabela pronta? Vamos ver como ficaria o teste do algoritmo das quatro operações?

Algoritmo Quatro_operacoes_basicas

// rotina que recebe dois números e que apresenta o resultado das quatro operações feitas com eles

```
Var
  NUM_A, NUM_B, SOMA, SUBTRACAO, MULTIPLICACAO, DIVISAO: real
  NUM_A = 0
  NUM_B = 0
  SOMA = 0
  SUBTRACAO = 0
  MULTIPLICACAO = 0
  DIVISAO = 0
Início
  Escreva "Programa que executa as quatro operações básicas"
  Escreva "Informe o primeiro número"
  Leia NUM_A
  Escreva "Informe o segundo número."
  Leia NUM_B
  SOMA = NUM_A + NUM_B
```

```
SUBTRACAO = NUM_A - NUM_B
MULTIPLICACAO = NUM_A * NUM_B
DIVISAO = NUM_A / NUM_B
Escreva "A soma dos dois números é"
Escreva SOMA
Escreva "A subtração dos dois números é"
Escreva SUBTRACAO
Escreva "A multiplicação entre os dois números é"
Escreva MULTIPLICACAO
Escreva "A divisão entre os dois números é"
Escreva DIVISAO
```

Fim.

1º TESTE

Considere os valores 30 e 60 como os dois números (NUM_A e NUM_B, respectivamente) solicitados pelo algoritmo. A partir desses dois valores, siga o encadeamento de ações proposto nas linhas do pseudocódigo.

Quando o algoritmo solicitar a ação Escreva, preencha o campo correspondente na tabela de caneta azul. Para marcar as ações Leia, coloque o valor lido entre parênteses de caneta vermelha. Essas regras simples são importantes para registrar a verificação de todas as ações previstas no algoritmo.

2º TESTE

Repita o procedimento realizado no primeiro teste, considerando os valores -20 e 40 como NUM_A e NUM_B, respectivamente.

3º TESTE

Repita o procedimento, considerando, agora, os valores 20 e -40.

Como foi a realização do teste de mesa? O que você achou da técnica?

Encontrou alguma oportunidade de melhoria no pseudocódigo testado?

Utilização de Pseudocódigos

Como você deve ter percebido, a criação de algoritmos em pseudocódigo é relativamente simples, não é mesmo?

O pseudocódigo é um método que não usa estruturas complexas e rígidas – como as sintaxes de linguagens de programação –, nem exige ambientes especiais para edição.

Em outras palavras, o pseudocódigo pode ser encarado com uma versão similar e simplificada dos códigos de programação, com o benefício de poder ser escrito em qualquer editor de texto.

definir os tipos de variáveis relacionadas ao cadastro de um livro, receber os dados do usuário e exibi-los na tela

ALGORITMO Livro (definição do nome do programa)

VAR

CódigoDoLivro: inteiro [Nome da Variável: Tipo da variável]
I Título, Autor, Editora: literal [declaração de variáveis, todas como literais]

CódigoDoLivro = 0

INICIO

Escreva "Este é um programa em pseudocódigo que exibe na tela os dados de um livro"

Escreva "Digite o código do livro"

Leia CódigoDoLivro

Escreva "Digite o título do livro"

Leia Título

Escreva "Digite o autor do livro"

Leia Autor

Escreva "Digite a editora do livro"

Leia Editora

Escreva "O código do livro é", CódigoDoLivro

Escreva "O título do livro é", Título

Escreva "O autor do livro é", Autor

Escreva "A Editora do livro é", Editora

FIM.

Você percebeu como a descrição é clara e de fácil compreensão?

Uma das principais vantagens desse método é que ele pode ser entendido por "não programadores". Com isso, os desenvolvedores podem apresentar suas ideias e validar sequências instrucionais com profissionais que não são da área de programação. Trabalhar com pseudocódigo parece ser muito eficiente, não acha?

1. Todo algoritmo em pseudocódigo deve ser iniciado com **Algoritmo**: **NomeDoAlgoritmo**.
2. O início e fim do programa são limitados pelos marcadores **Início** e **Fim**.
3. As variáveis são declaradas no início do algoritmo, abaixo do marcador **Var**, da seguinte forma: **NomeDaVariável**: Tipo da variável.
4. Os nomes das variáveis **NÃO** podem:
 - Iniciar por número (erro: 1nome)
 - Ter espaço (erro: nome completo)
 - Ter caracteres especiais (., ~, ¢, - e outros).
5. As palavras-reservadas devem ser evitadas: **Início**, **Fim**, **Var**, **Se** e **Senão**.
6. Os nomes das variáveis são **case sensitive**. Considera a diferença no uso de letras maiúsculas e minúsculas. Dessa forma, ao manipularmos variáveis, devemos usar o mesmo nome declarado no início, considerando o uso de letras maiúsculas e minúsculas.
7. O comando **Leia** deve ser usado para receber (capturar) dados do usuário, fase do processamento conhecida como "Entrada de Dados".
8. O comando **Escreva** deve ser usado para exibir (apresentar, mostrar) dados ao usuário, fase do processamento conhecida como "Saída de Dados".
9. Os textos a serem exibidos na tela ou que devam ser inseridos como caractere são colocados entre "aspas" (representação universal de um valor literal ou **string** – "Sistemas").

10. Os comentários sobre o código podem ser inseridos (entre chaves) (incomum por confundir com agrupamentos) ou inseridos utilizando `//` (mais comum) no início da linha de instruções. O comentário não altera a execução do código. Contudo, ele é de fundamental importância para documentar e tornar inteligíveis as escolhas de programação realizadas no código para outros programadores.
11. As regras são muito importantes, já que todos os elementos do pseudocódigo – e também dos códigos – são fundamentais para o funcionamento correto do programa. Desse modo, entender o rigor da padronização e aplicá-la é essencial a quem deseja atuar com programação de computadores.

2.12 - Operadores: Tipos e Usos

Com tudo o que vimos até aqui, podemos afirmar que programar computadores é lidar com informações e instruções para manipular dados e chegar a resultados. Nesse sentido, os programas podem realizar cálculos simples e complexos, comparar dados inseridos pelos usuários e dar diferentes encaminhamentos a esses dados, dependendo do processamento.

O que o computador fará dependerá do uso dos chamados operadores. E o que são operadores?

Operadores são elementos que atuam sobre os operandos (variáveis ou constantes), ou seja, são símbolos ou palavras reservadas (`true/false`) por meio dos quais é possível fazer cálculos, comparar resultados e atribuir valores às variáveis.

Com isso, os operadores são a chave para que possamos incrementar, decrementar, comparar e avaliar dados dentro do computador.

Percebeu como é importante conhecer e saber utilizar corretamente os operadores?

Agora que já sabemos o que é um operador, vamos conhecer os tipos de operadores. Existem três tipos diferentes de operadores: aritméticos, relacionais e lógicos. Clique em cada um deles a seguir para obter mais detalhes.

Operadores aritméticos

Possibilitam realizar cálculos em expressões matemáticas ou aritméticas. Suas palavras ou símbolos reservados são:

Operação	Símbolo
Soma	+
Subtração	-

Operação	Símbolo
Multiplicação	*
Divisão	/
Exponenciação	** ou ^

Operadores relacionais

Possibilitam fazer comparações ou relacionamentos entre dados numéricos e literais (caracteres). Suas palavras ou símbolos reservados são:

Operação	Símbolo
Maior que	>
Menor que	<
Maior ou igual a	\geq
Menor ou igual a	\leq
Igual a	=
Diferente de	\neq ou # ou !=

Os operadores relacionais ou de comparação são muito utilizados nas codificações pelas Linguagens de Programação. Entretanto, alguns operadores

relacionais possuem outros símbolos para representá-los, mas com as mesmas finalidades. Por exemplo:

= (igual a), que representa uma igualdade na lógica, normalmente quando se deseja fazer uma comparação em uma decisão condicional, é simbolizado nas linguagens de programação com duplo sinal de igual ==, x==2 (SE x é igual a

2). Apenas = é utilizado como atribuição, ou seja, x = 2 (atribuição do valor 2 para x), o que não é uma comparação.

\neq (diferente de), que representa uma desigualdade na lógica, é simbolizado de diferentes maneiras nas linguagens de programação. Para utilizar o \neq (diferente de) nas linguagens, você encontrará outros símbolos, como #, != e \neq entre outros. Esse detalhamento você aprenderá quando for estudar as linguagens de programação.

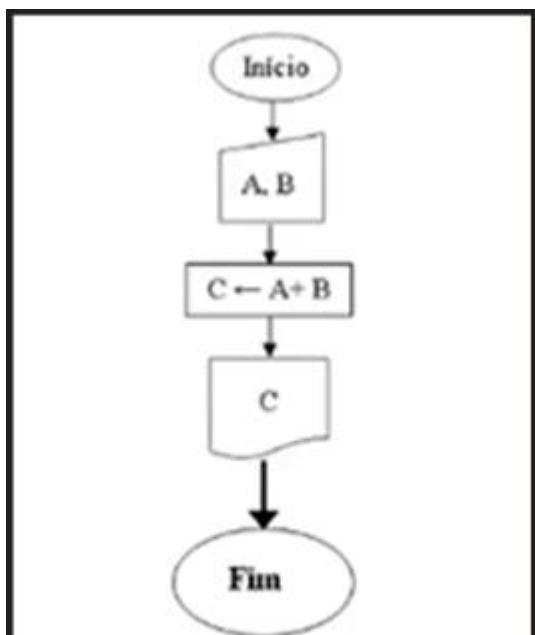
Operadores lógicos

Também chamados de booleanos. Permitem comparar ou relacionar resultados de expressões. Suas palavras ou símbolos reservados são:

Operação	Operador
Conjunção	E
Disjunção (não exclusiva)	OU
Negação	NÃO
Disjunção (exclusiva)	XOU (é-se "ou exclusivo")

Os operadores lógicos ou booleanos também são muito utilizados nas codificações pelas Linguagens de Programação. Quando utilizamos operadores lógicos, obrigatoriamente, estamos nos colocando em uma

E = +, OU = -, e NÃO é inversão.

**Figura 2:** Exemplo de Fluxograma**2.15 - Português Estruturado**

O Português Estruturado é uma forma especial de linguagem bem mais restrita que a Língua Portuguesa e com significados bem definidos para todos os termos utilizados nas instruções (comandos).

Essa linguagem também é conhecida como Portugol (junção de Português com Algol, Pseudocódigo ou Pseudolínguagem). O

Português Estruturado na verdade é uma simplificação extrema da língua portuguesa, limitada a pouquíssimas palavras e estruturas que têm significado pré-definido, pois deve-se seguir um padrão. Emprega uma linguagem intermediária entre a linguagem natural e uma linguagem de programação, para descrever os algoritmos.

A sintaxe do Português Estruturado não precisa ser seguida tão rigorosamente quanto a sintaxe de uma linguagem de programação, já que o algoritmo não será executado como um programa.

Embora o Português Estruturado seja uma linguagem bastante simplificada, ela possui todos os elementos básicos e uma estrutura semelhante à de uma linguagem de programação de computadores.

Portanto, resolver problemas com português estruturado pode ser uma tarefa tão complexa quanto a de escrever um programa em uma linguagem de programação qualquer só não tão rígida quanto a sua sintaxe, ou seja, o algoritmo não deixa de funcionar porque esquecemos de colocar um ';' (ponto-e-virgula) por exemplo, já um programa não funcionaria.

A Figura 3 apresenta um exemplo de algoritmo na forma de representação de português estruturado.

3 - Conceitos Importantes

Neste capítulo são apresentados e explicados três conceitos fundamentais para a construção de algoritmos, são eles: Constante, Variável e Atribuição.



3.1 - Constantes

São chamadas de constantes, as informações (dados) que não variam com o tempo, ou seja, permanecem sempre com o mesmo conteúdo, é um valor fixo (invariável). Como exemplos de constantes pode-se citar: números, letras, palavras etc.

3.2 - Variáveis

O bom entendimento do conceito de variável é fundamental para elaboração de algoritmos e, consequentemente de programas. Uma variável é um espaço da memória do computador que "reservamos" para guardar informações (dados). Como o próprio nome sugere as variáveis, podem conter valores diferentes a cada instante de tempo, ou seja, seu conteúdo pode variar de acordo com as instruções do algoritmo.

As variáveis são referenciadas através de um nome (identificador) criado por você durante o desenvolvimento do algoritmo. Exemplos de nomes de variáveis: produto, idade, a, x, nota1, peso, preço, etc. O conteúdo de uma variável pode ser alterado, consultado ou apagado quantas vezes forem necessárias durante o algoritmo.

Mas, ao alterar o conteúdo da variável, a informação anterior é perdida, ou seja, sempre "vale" a última informação armazenada na variável. Uma variável armazena 'apenas' um conteúdo de cada vez.

como FATOR, contém o valor 5. Em outro momento essa caixa poderá conter qualquer outro valor numérico. Entretanto, a cada instante, ela conterá um, e somente um valor.

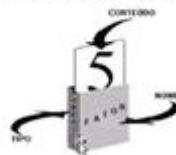


Figura 4: Ilustração de Variável

3.3 - Atribuição

A atribuição é uma notação utilizada para atribuir um valor a uma variável, ou seja, para armazenar um determinado conteúdo em uma variável. A operação de atribuição, normalmente, é representada por uma seta apontando para a esquerda, mas existem outros símbolos para representar a atribuição, depende da forma de representação do algoritmo. Na Tabela 2 a seguir, são apresentados alguns exemplos de atribuições possíveis:

Atribuições Possíveis	Exemplos
<code>variável</code> \leftarrow constante;	<code>idade</code> \leftarrow 12 (lê-se: idade recebe 12)
<code>variável</code> \leftarrow variável;	<code>preço</code> \leftarrow <code>valor</code>
<code>variável</code> \leftarrow expressão;	<code>A</code> \leftarrow <code>B + C</code>

Tabela 2: Exemplos de Atribuições

4-Instrução Escrever

Existem basicamente duas instruções principais em algoritmos (e em programação em geral) que são:

Escrever e Ler. Neste capítulo veremos como funciona a instrução *Escrever*.

I

A instrução *Escrever* é utilizada quando deseja-se mostrar informações na tela do computador, ou seja, é um comando de **saida de dados**. Para simplificar, usa-se a instrução *Escrever*, quando se necessita **mostrar algum dado para o usuário** do algoritmo (e posteriormente do programa).

Tanto no Diagrama de Chapin quanto em Português Estruturado representa-se a saída de dados através da palavra *Escrever* (ou *Escreva*). Já em Fluxogramas a representação da saída de dados é feita através de uma forma geométrica específica.

A seguir, vamos nos aprofundar em alguns aspectos da declaração de variáveis. Preparado?

ALGORITMO Livro {definição do nome do programa}
VAR
CódigoDoLivro: inteiro
Título, autor, editora: **caractere** {declaração de variáveis}
INICIO
Escreva "este é um programa em pseudocódigo que exibe na tela os dados de um livro"
CódigoDoLivro = 1
Título = "Inferno"
Autor = "Dan Brown"
Editora = "Editora Arqueiro"
Escreva "o código do livro é", **CódigoDoLivro** {irá exibir 1}
Escreva "o título do livro é", Título {Inferno}
Escreva "o autor do livro é", Autor {irá exibir Dan Brown}
Escreva "a editora do livro é", Editora {irá exibir Editora Arqueiro}
FIM.

Note que quando queremos escrever *alguma mensagem na tela* (letra, frase, número etc.) literalmente, devemos utilizar *aspas* para identificar o que será escrito, pois *o que estiver entre aspas no algoritmo, será exatamente o que aparecerá na tela do computador*. Diferente de quando queremos escrever o conteúdo de uma variável, pois neste caso não se utiliza aspas.

5 Operadores Aritméticos

Muitas vezes, ao desenvolvemos algoritmos, é comum utilizarmos expressões matemáticas para a resolução de cálculos. Neste capítulo são apresentados os operadores aritméticos necessários para determinadas expressões.

Operação	Símbolo	Prioridade
Multiplicação (Produto)	*	1a.
Divisão	/	1a.
Adição (Soma)	+	2a.
Subtração (Diferença)	-	2a.

Tabela 3: Operadores Aritméticos

Nas linguagens de programação e, portanto, nos exercícios de algoritmos que iremos desenvolver, as expressões matemáticas sempre obedecem às regras matemáticas comuns, ou seja:

- As expressões dentro de parênteses são sempre resolvidas antes das expressões fora dos parênteses. Quando existem vários níveis de parênteses, ou seja, 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.

Desta forma, veja os seguintes exemplos e os respectivos resultados:

ExemploA: $2 + (6 * (3 + 2)) = 32$

ExemploB: $2 + 6 * (3 + 2) = 32$

6 - Instrução Ler

Como vimos no capítulo 4 *InSTRUÇÃO Escrever*, existe basicamente duas instruções principais em algoritmos (e em programação em geral) que são: Escrever e Ler. No capítulo 4, foi apresentada a

Estruturado representa-se a entrada de dados através da palavra *Ler* (ou *Leia*). Já em Fluxogramas a representação da entrada de dados é feita através de uma forma geométrica específica.

Exemplo 3:

Escreva um algoritmo para ler dois valores e armazenar cada um em uma variável. A seguir, armazenar a soma dos dois valores lidos em uma terceira variável. Escrever o resultado da soma efetuada.

7 - Horizontalização

Para o desenvolvimento de algoritmos que possuam cálculos matemáticos, as expressões aritméticas devem estar horizontalizadas, ou seja, linearizadas e também não se esquecendo de utilizar os operadores corretamente. Na Tabela 4 a seguir, é apresentado um exemplo de uma expressão aritmética na forma tradicional e como deve ser utilizada nos algoritmos e em programação em geral (linearmente).

Matemática Tradicional	Algoritmo
$M = \frac{N1+N2}{2}$	$M \leftarrow (N1 + N2) / 2$

Tabela 4: Exemplo de Horizontalização de Expressões Aritméticas

As expressões matemáticas na forma horizontalizada não são apenas utilizadas em algoritmos, mas também na maioria das linguagens de programação.

8 - Estruturas de Controle

Você aprendeu muita coisa no INÍCIO, não é mesmo? Compreendeu o que é um programa e como funciona como armazenar dados na memória do computador usando variáveis e como realizar operações de diversos tipos utilizando operadores. Agora, você estudará instruções que permitem controlar o fluxo de um programa a fim de dar um pouco mais de complexidade ao código. Essas estruturas de controle servem para **dirigir o fluxo de processamento dos dados**. Para que você não fique com nenhuma dúvida, preparamos um esquema com os tipos de estrutura de controle.

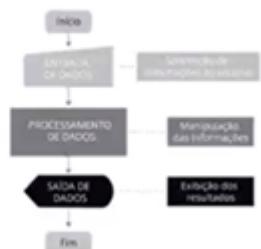
Veja no monitor para entender melhor.



8.1 - Estruturas Sequenciais

A estrutura sequencial é uma sequência de instruções que acontecem uma após a outra, sem desvios ou interrupções. Esse tipo de estrutura é composto somente de linhas de comando sucessivas (que compõe um único fluxo possível de processamento) e está sempre limitado pelos marcadores INÍCIO e FIM do algoritmo ou de uma outra estrutura. As estruturas sequenciais podem ser representadas por fluxogramas e pseudocódigos. Vamos ver exemplos?

Fluxograma



8.2 - Estruturas de Seleção

As estruturas de seleção são usadas para que sejam estabelecidos caminhos diferentes de instruções, a serem percorridos a partir de tomadas de decisão. Justamente por isso, esses recursos podem ser chamados de estruturas de seleção ou estruturas de decisão.

Como programador, você deverá utilizar os recursos de estruturas de seleção sempre que tiver de estruturar sequências de ações que poderão ser executadas ou não, a depender de um resultado frente a uma ou mais condições. Existem três tipos de estruturas de seleção, cuja aplicação irá depender do contexto de utilização. Clique em cada estrutura para conhecê-la.

A estrutura de seleção simples faz uso da instrução **SE (IF)**. Ela é utilizada quando queremos testar uma condição antes de executarmos uma ou mais instruções.

Nessa estrutura, somente teremos uma ação SE o resultado da condição for **verdadeiro**. Desse modo, não há ação a ser executada caso o resultado seja falso.

SE CONDIÇÃO = VERDADEIRO → AÇÃO
SE CONDIÇÃO = FALSO → NÃO HÁ AÇÃO A SER EXECUTADA

Para compreender melhor a utilização desse tipo de estrutura de seleção, veja o fluxograma a seguir.

8.3 - Seleção Simples

SE <CONDIÇÃO> ENTÃO

Comandos a serem executados somente se a condição for verdadeira

FIM SE

Veja um exemplo para entender melhor como funciona a estrutura de seleção simples. Em nosso exemplo, vamos desenvolver um algoritmo em pseudocódigo para ler o nome de um aluno, a nota de duas avaliações e calcular a média final dele.

8.4 - Seleção Composta

Na estrutura de seleção composta existem **dois caminhos diferentes predeterminados**. Dessa maneira, haverá a execução de um comando ou grupo de comandos caso o resultado da condição seja **verdadeiro**, OU a execução de outro comando ou grupo de comandos diferente caso o resultado da condição seja **falso**.

SE CONDIÇÃO = VERDADEIRO → AÇÃO 1
SE CONDIÇÃO = FALSO → AÇÃO 2

Para compreender melhor a utilização desse tipo de estrutura de seleção composta, veja o fluxograma a seguir.

Agora, vamos conhecer as palavras-chave (marcadores) de uma **estrutura de seleção composta**:

SE <CONDIÇÃO> ENTÃO

Comandos a serem executados somente se a condição for verdadeira

SENAO

Comandos a serem executados somente se a condição for falsa

FIM SE

Vejamos um exemplo para entender como fica a utilização prática

Estrutura de Seleção Composta Encadeada

```
ALGORITIMO ESTADOCIVIL.  
VAR  
EC: inteiro  
EC = 0  
INICIO  
    Escreva "Digite o estado civil (1 – solteiro, 2 – casado, 3 –  
outros)"  
    Leia EC  
    SE EC = 1 ENTAO  
        Escreva "Solteiro"  
    SENAO  
        SE EC = 2 ENTAO  
            Escreva "Casado"  
        SENAO  
            SE EC = 3 ENTAO  
                Escreva "Outros"  
            SENAO  
                Escreva "Valor inválido"  
            FIM SE  
        FIM SE  
    FIM SE  
FIM
```

(O FIM SE é de trás para frente e tem que fechar todos os SE)

```
EC: inteiro  
EC = 0  
INICIO  
    Escreva "Digite o estado civil (1 – solteiro, 2 – casado, 3 –  
outros)"  
    Leia EC  
    SE EC = 1 ENTAO  
        Escreva "Solteiro"  
    SENAO  
        SE EC = 2 ENTAO  
            Escreva "Casado"  
        SENAO  
            SE EC = 3 ENTAO  
                Escreva "Outros"  
            SENAO  
                Escreva "Valor inválido"  
            FIM SE  
        FIM SE  
    FIM SE  
FIM
```

2604

Escreva aqui um programa em Portugol e utilizando o Word ou no papel faça um Fluxograma,
que leia dois valores A e B, calcule a soma entre eles e exiba para o usuário.

Todo algoritmo tem que ter um nome. Os parênteses são parâmetros.

```
Nome do algoritmo - Calcular( )  
Variaveis - var  
A,B,R : INTEIRO  
  
A: inteiro;  
B: inteiro;  
R : inteiro;  
Inicio  
    Leia A,B;
```

Tanto faz, o primeiro só está ocupando menos espaço.

Calcula a + b; errado

Calcula a + b é o pensamento, não pode escrever assim.

a + b = calcule; **ERRADO**
R ← A+B; correto

A,B,R : INTEIRO;
Inicio
Leia A,B;
Calcula a + b; errado
a + b = calcule; ERRADO
SOMA A + B; ERRADO
SOMA A E B; ERRADO
R ← A+B; correto

INICIO
MOSTRAR "ENTRA COM OS NUMEROS"
SOMA A E B
ERRADO

INICIO
eSQR[REVA]"ENTRA COM OS NUMEROS", A,B;
CORRETO
Inicio
Mostrar "entra com o numero";
A;
B;
CORRETO

Nome do algoritmo - Calcular()
Variaveis - var
A,B,R : INTEIRO;
Inicio
Leia A,B;
Calcula a + b; errado
a + b = calcule; ERRADO
SOMA A + B; ERRADO
SOMA A E B; ERRADO
R ← A+B; correto
Exiba R;
MOSTRE_r;
Escreva R;
Exibir r
Mostrar r
CORRETO

INICIO
MOSTRAR "ENTRA COM OS NUMEROS"
SOMA A E B
ERRADO

INICIO
MOSTRAR "ENTRA COM OS NUMEROS", A,B;
CORRETO

Bonitinho:

Escreva "Mostre o resultado=" R;

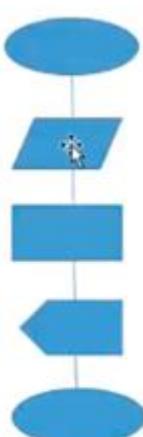
Algoritmo pronto:

```
Nome do algoritmo - Calcular( )
Variaveis - var
A,B,R : INTEIRO;
Início
    Leia A,B;
    Calcula a + b; errado
    a + b = calcule; ERRADO
    SOMA A + B; ERRADO
    SOMA A E B; ERRADO
    R ← A+B; correto
    Exiba R;
    Escreva "Mostre o resultado=" R;
    MOSTRE "Resultado=";
    r;
    Escreva R;
    Exibir r
    Mostrar r;
    CORRETO
Fim.
```

Ao invés da setinha, pode colocar o sinal de igual: $R = A + B$, mas não é igualdade, é a variável.

Início, entrada de dados, processamento, exibir na tela e fim – fluxograma.

```
Nome do algoritmo - Calcular()
Variaveis - var
A,B,R : INTEIRO;
Início
    Leia A,B;
    Calcula a + b; errado
    a + b = calcule; ERRADO
    SOMA A + B; ERRADO
    SOMA A E B; ERRADO
    R ← ou = A+B; correto
    Exiba R;
    Escreva "Mostre o resultado=" R;
    MOSTRE "Resultado=";
    r;
    Escreva R;
    Exibir r
    Mostrar r;
    CORRETO
Fim.
```



2º algoritmo:

Respostas

```
1(1,5,6,1)
Variáveis
    Dias,mês,ano,total: inteiro;
Inicio
    Leia dia,mês,ano:;
    mês = mês * 30;
Se(ano mod 400 = 0) ou ((ano mod 4 = 0) e (ano mod 100 <> 0)) então
    escreva("ano, é bissexto!");
    ano = ano*366
    senão
        ano = ano * 365;
    fim-se
    total = (dia *mês*ano);
    escreva total;
Fim
```

mod = divisão pelo resto, não exata

```
23
23/2           11,5 / 2 = isto significa MOD["função"] divisão pelo resto
Resultado / 2
Resultado / 2
    0 ou 1
2 elevado 0 + 2 elevado 1 + 2 elevado 2 + 2 Eleveda 3 + 2 elevado 4.....
10111
1+0+4+8+16= 23
```

- Divisão com número binário

Estrutura de desvio/controle/alternativa:

```
Se(ano mod 400 = 0) ou ((ano mod 4 = 0) e (ano mod 100 <> 0)) então
    escreva("ano, é bissexto!")
    ano = ano*366
    senão
        ano = ano * 365;
    fim-se
```

Resto = a mod b

$$\begin{array}{r} 2020 \text{ / } 4 \\ 20 \quad \quad 55 \\ \hline 020 \\ \quad 20 \\ \hline 0 \\ 23 \end{array}$$

Na verdade, dá 505.

Execução no Java:

```

// Exercício 2.

package ex2;

import javax.swing.JOptionPane;

public class Ex2 {

    public static void main(String[] args) {
        int dia, mês, ano, total;

        dia = Integer.valueOf(JOptionPane.showInputDialog("Digite o Dia:"));

        mês = Integer.valueOf(JOptionPane.showInputDialog("Digite o Mês:"));

        ano = Integer.valueOf(JOptionPane.showInputDialog("Digite o Ano:"));

        mês = mês * 30;

        if ((ano % 400 == 0) || ((ano % 4 == 0) & (ano % 100 != 0))) {
            System.out.println("O ano é bissexto!");

            ano = ano * 366;
        } else {
            ano = ano * 365;
        }

        total = (dia * mês * ano);

        System.out.println(total);
    }
}

```

0305



Pseudo ambiente: NetBeans



Java é uma linguagem orientada a objetos¹ que foi desenvolvida nos anos de 1990 pela Sun Microsystems, projetada para ser pequena, simples e portável a todas as plataformas e sistemas operacionais. É utilizada para desenvolver aplicativos corporativos, páginas web com conteúdo dinâmico e interativo, aprimorar funcionalidades de servidores www e está cada vez mais sendo utilizada para desenvolvimento de aplicativos móveis para telefones celulares, pagers e PDAs.



Muito provavelmente você já deve ter ouvido falar sobre a linguagem Java. Basta ler uma revista de informática ou matérias sobre desenvolvimento de softwares que logo encontra alguma informação sobre Java, dado o sucesso que esta linguagem tem tido no mercado, principalmente pelo fato dos programas escritos em Java poderem ser executados virtualmente em qualquer plataforma e aceitos em qualquer tipo de computador ou outros aparelhos, uma característica marcante da Internet. Por isso, a linguagem Java tem se destacado muito no mercado e aprendê-la é importante para você e para a sua profissão!



Java é uma Linguagem de Programação Orientada a Objetos, porém para que você compreenda melhor a Lógica utilizando esta Linguagem, iremos utilizar o Console (uma forma estruturada) para exercitar os comandos.

Para escrevermos um programa desenvolvido em Pseudocódigo de forma que o computador possa compilá-lo³ e executá-lo, precisamos utilizar uma Linguagem de Programação que nada mais é do que o “índia” necessário para conversar com o computador. No desenvolvimento de nossos estudos em Lógica de Programação utilizaremos a linguagem de programação Java.

O Java atualmente é uma das linguagens de programação mais utilizadas no mercado de trabalho. É também capaz de fornecer uma portabilidade muito grande, sendo compatível com a maioria dos Sistemas Operacionais disponíveis para usuários gerais.

O mesmo programa escrito com a Linguagem de Programação Java, poderá ser utilizado em um computador com os Sistemas Operacionais Windows, Linux ou Mac OS sem que nenhuma linha do código fonte³ necessite ser alterada, ganhando tempo no desenvolvimento de novas aplicações.

1 - Linguagem de Programação Orientada a Objetos (P.O.O) significa que esta Linguagem de Programação utilizará os conceitos de Orientação a Objetos (O.O.) a qual veremos no módulo seguinte com mais profundidade.

2 - Compilação é a ação de transformar um código amigável escrito com uma Linguagem de Programação em um programa executável baseado em Código de Máquina.

3 - Código Fonte é o conjunto de instruções criadas pelo programador utilizando uma Linguagem de Programação.

Além disto, Java é uma linguagem de programação que pode ser utilizada para desenvolver páginas da Internet, por meio de um Servidor Web configurado para executar páginas do tipo Java Server Pages (JSP). Com ele também podemos desenvolver aplicativos para celulares que utilizam o Sistema Operacional Android.

Pela sua vasta lista de plataformas suportadas, utilizaremos o Java como Linguagem de Programação de apoio no desenvolvimento da Lógica de Programação.



Você sabia que o Java não é a única linguagem de Programação que funciona em todos os principais Sistemas Operacionais do mercado?
Pesquise outras linguagens existentes no mercado e reflita sobre a eficiência delas em relação à Linguagem Java.

Conhecendo as ferramentas Java

Para iniciar o estudo, primeiramente é necessário conhecer as diferentes ferramentas para desenvolvimento que o Java oferece (JRE, JVM, JSE, JEE, IME, JDK...) parece uma sopa de letrinha, não é mesmo? Em seguida, com base na aplicação que pretende desenvolver, deve selecionar as tecnologias e ferramentas necessárias para este fim.

É mais ou menos assim: Imagine que você precisa fazer um brigadeiro! Com a receita em mãos, você vai até o supermercado, compra os ingredientes: leite condensado, chocolate em pó, manteiga e chocolate granulado e ainda providencia as ferramentas necessárias para a confecção do doce: panela e colher de pau.



A mesma coisa acontece com o desenvolvimento de um aplicativo em Java! Conhecendo o tipo de aplicação que deseja desenvolver, você seleciona as ferramentas necessárias para tal finalidade. Vamos lá:

USA O JDK E JVM

Java Virtual Machine – JVM

O que faz com que a portabilidade da linguagem Java seja eficiente é uma aplicação responsável por executar programas desenvolvidos na linguagem. Sua função é simular a um computador permitindo a execução do código fonte, por isto recebe o nome de Máquina Virtual.

Na prática, basta instalar em seu computador a JVM desenvolvida para o Sistema Operacional e você estará pronto para executar programas desenvolvidos em Java.

O Java Development Kit (JDK) e a Integrated Development Environment (IDE) Eclipse



Como em breve seremos desenvolvedores e não usuários, também precisaremos instalar em nosso computador o kit para desenvolvimento de programas feitos em Java, o que inclui o compilador⁴ da linguagem de programação e a Máquina Virtual (JVM). Sem ele não é possível finalizar um programa desenvolvido em Java, mesmo que você escreva o código fonte completo.

Apesar de já ser possível criar programas apenas com o JDK, utilizaremos uma interface de desenvolvimento integrada (IDE) para auxiliar na codificação, compilação e testes dos nossos programas. O Java tem como principais IDEs de desenvolvimento o NetBeans e o Eclipse. Independente da IDE escolhida, os comandos sempre serão os mesmos, logo, se você aprende a programar em Java, conseguirá utilizar qualquer uma das IDEs sem maiores problemas.

I

Durante o desenvolvimento das atividades utilizaremos a IDE Eclipse, porém você pode utilizar qualquer outra IDE desde que siga fielmente as estruturas listadas neste material.

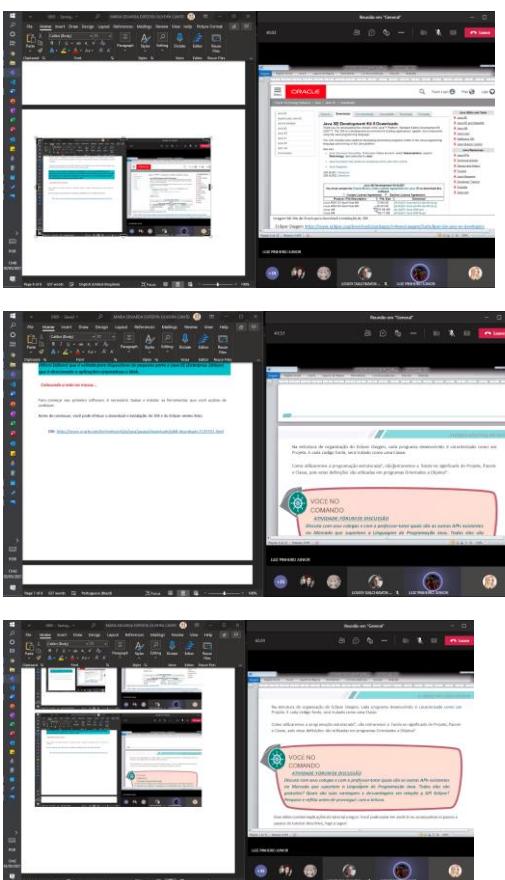
Durante os nossos estudos, utilizaremos a distribuição do Java SE (Standard Edition), que é voltada para o Desenvolvimento de Sistemas Desktop. Existem também as Distribuições Java ME (Micro Edition) que é voltada para dispositivos de pequeno porte e Java EE (Enterprise Edition) que é direcionado a aplicações corporativas e Web.

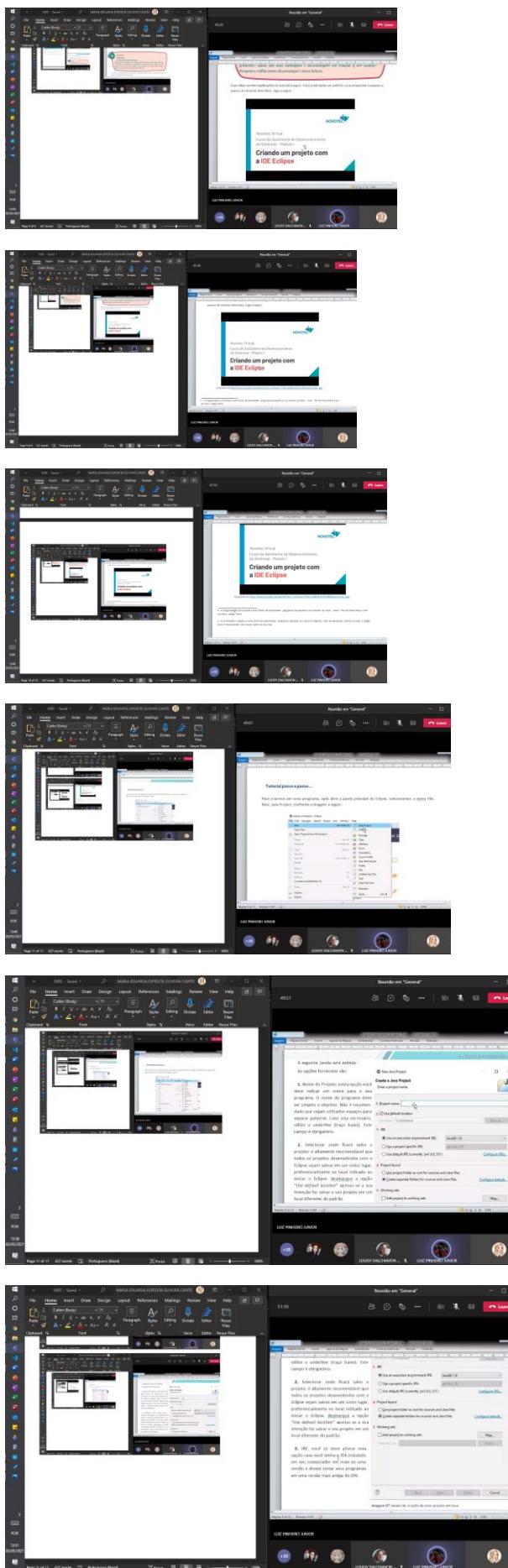
Colocando a mão na massa...

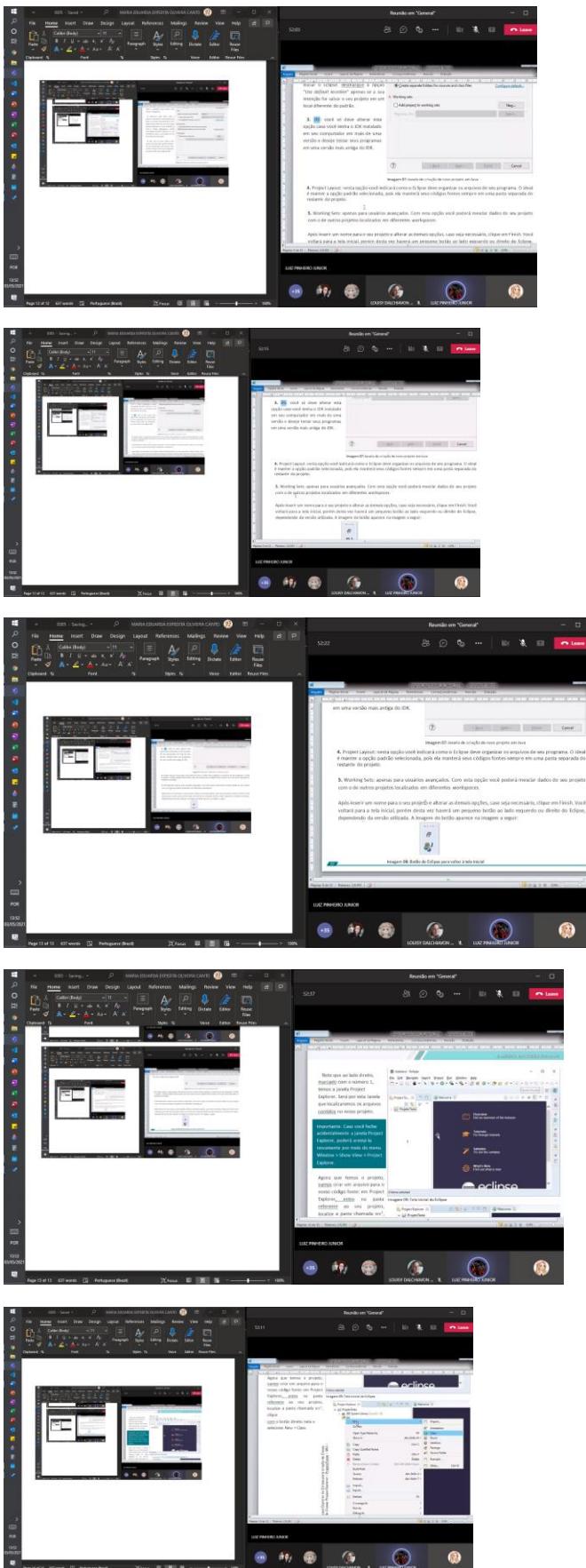
Para começar seu primeiro software, é necessário baixar e instalar as ferramentas que você acabou de conhecer.

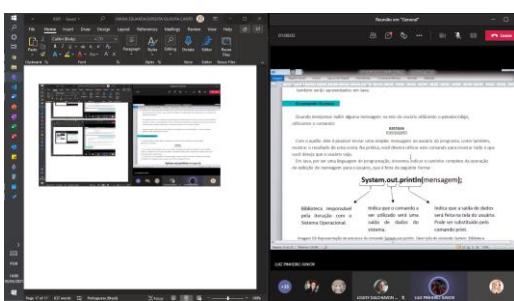
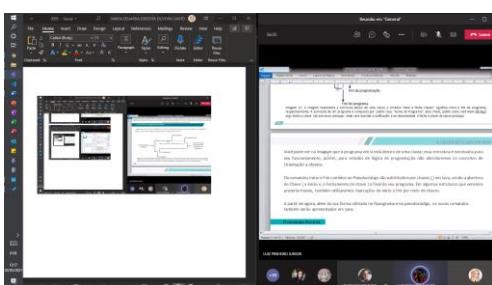
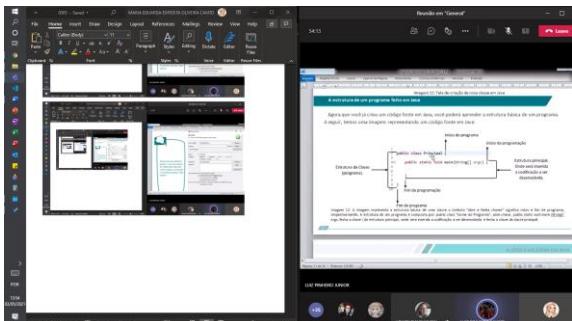
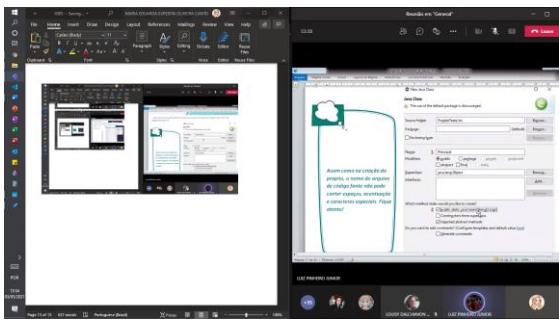
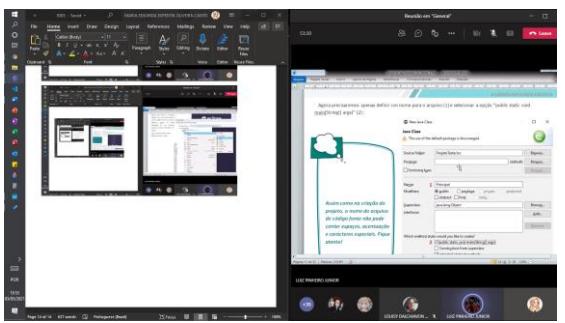
Antes de continuar, você pode efetuar o download e instalação do JDK e do Eclipse nestes links:

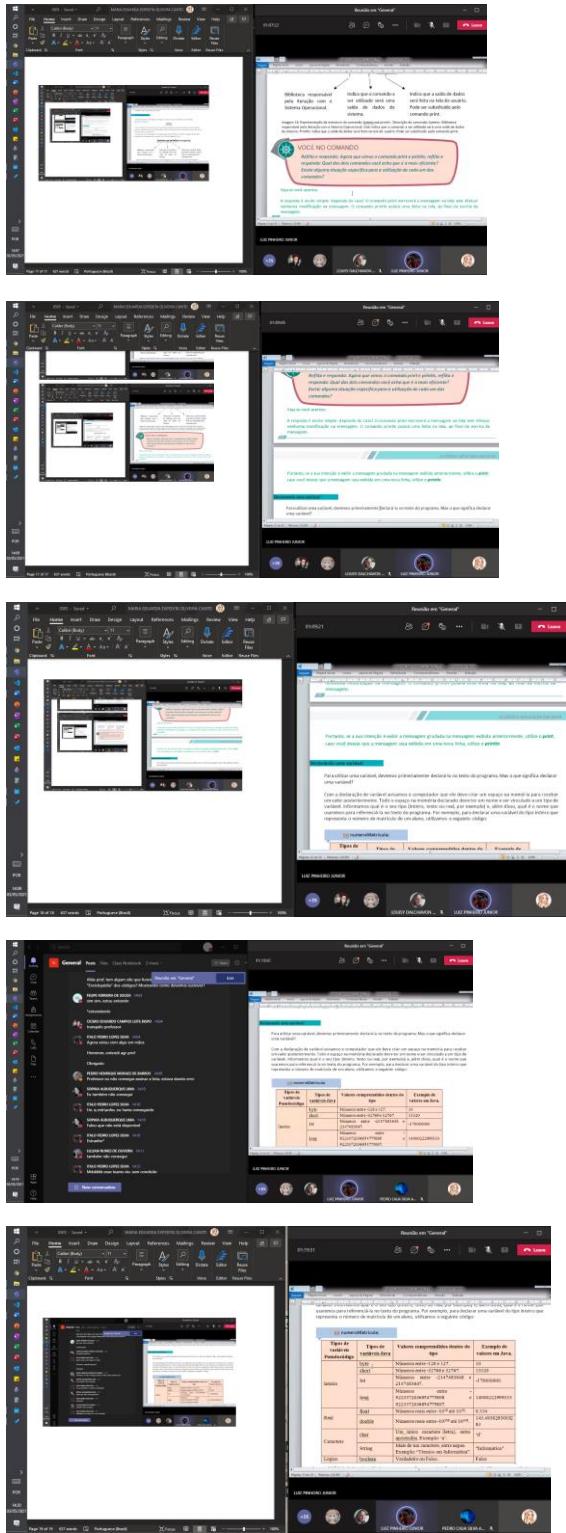
JDK: <https://www.oracle.com/technetwork/pt/java/javase/downloads/jdk8-downloads-2133151.html>











1005

The screenshot shows a Microsoft Word document titled "Reunião em "General"" with the file path "logica_aplicada_a_linguagem_JAVA.docx". The document contains a table comparing Java variable types with their representations in pseudocode and Java code.

Tipo	Pseudocódigo	Java
Real	float	Números reais entre -10 ⁻⁴ até 10 ⁴ . double
Caractere	char	Um único caractere (letra), entre aspas. Exemplo: "a".
String	String	Mais de um caractere, entre aspas. Exemplo: "Técnico em Informática".
Lógico	boolean	Verdadeiro ou Falso. Exemplo: true ou false.

Below the table, there is a note in Portuguese:

Imagen 14: Tabela de tipos de variáveis e suas representações em pseudocódigo e Java – no pseudocódigo temos as variáveis inteiro, real, caractere e lógico em Java o inteiro é representado pelos tipos byte, short, int, long, o real por float e double, o tipo caractere por char e String e o lógico como boolean, lembrando que cada tipo de variável numérica é representada pela sua faixa de valores.

Note que em Java, o único tipo de dado que se inicia com letra maiúscula é o **String**. Vale lembrar também que os números decimais são separados por ponto (.) ao invés de vírgula.

Voltando ao caso da Agenda Telefônica, qual seria o tipo de dado a ser aplicado para indicar o número de telefone informado por Juvenal?

The Trello board "Review Schedule" has three columns: "7 Days", "15 Days", and "23 Days". Under the first column, there are cards for "Geography" (due 16 May, 0/12), "P. Education" (due 16 May, 0/6), and "TPA" (due 16 May, 0/16). A tooltip "Add another card" is visible over the third column.

Geografia | Google | trello.com | +

trello.com/c/AxRf4AIB/1-g... Apps YouTube Gmail Classroom Grammarly

Description
Add a more detailed description...

Topics
Orientation and Location 😊
Geographic Coordinates
Scales
Graphics - Lines and Colors
Timezones
Book Exercises
Exercise Correction
Formation and Structure of the Earth
Tectonic Plates
Movie Report (Earth: Making of a Planet)
Consequences of Tectonism (Folds, Faults and Volcanism)
Consequences of Tectonism (Earthquakes and Tsunamis)

Add an item

Activity

Review Schedule | Trello

Board Review Schedule Trello workspace

7 Days Geography Orientation and Location
P. Education
TPA

Geography
Orientation and Location
Geographic Coordinates
Scales
Graphics - Lines and Colors
Timezones
Book Exercises
Exercise Correction
Formation and Structure of the Earth
Tectonic Plates
Movie Report (

Add another card

Geography

Orientation and Location

Geographic Coordinates

Scales

Graphics - Lines and Colors

Timezones

Book Exercises

Exercise Correction

Formation and Structure of the Earth

Tectonic Plates

Movie Report (

Add card

Reunião em "General"

25:29

Arquivo Página Inicial Inserir Layout da Página Páginas Iniciais Correspondências Revisão Edição

A LÓGICA APLICADA EM JAVA

O número de telefone seria uma variável do tipo String, pois apesar de ser um número de telefone, não efetuamos nenhum tipo de cálculo com este número, sendo assim, não é necessário definir este tipo de dado como inteiro.

Após identificar o tipo da variável que você utilizará, basta declará-la® em seu programa, seguindo o padrão para a cada linguagem de programação, sendo:

PSEUDOCÓDIGO	JAVA
declare nome da variável como tipodedado	declare tipodedado nome da variável;

A seguir, temos um exemplo de sua aplicação utilizando o pseudocódigo e o Java:

Pseudocódigo	Java
Exemplo declare nome como caractere idade como inteiro preco como real	String nome; int idade; double preco;

Imagem 15: Tabela de representação de declaração de variáveis em fluxograma, pseudocódigo e codificação em Java – deve-se seguir o padrão de cada linguagem, ambos fluxograma e pseudocódigo as variáveis são declaradas da mesma forma, nome da variável e o seu tipo, em Java primeiro declara-se o tipo de dado e depois o nome da variável.

LUIZ PINHEIRO JUNIOR +35 HENRIQUE SANTANA ...

School Review | Google | trello.com | +

trello.com/_eduardoalcantao_boards | Boards | trello.com

Boards | Trello

TPA in list 7 Days DUE DATE 16 May at 15:30

Description Add a more detailed description...

Topics
Logic in Programming
Reasoning Logic
Programming Logic
Algorithm Programming
Programs, applications or
Object Oriented Program
Ways of Representing the
Flowchart and Use of Ge
Narrative Flowchart and
Pseudocode

Add board title Trello workspace Workspace visible

Create board Start with a template

Boards Templates Home WORKSPACES Trello workspace Getting started Highlights Workspace table Members Settings

Upgraded this Workspace Unlimited boards, automation, Power-Ups and so much more! Try for 14 days

Recently viewed School Review Schedule TEMPLATE Remote Class Template Hub

Browse the full template gallery

nome do aluno como caractere nomeAluno como caractere

String data_nascimento; String data_nascimento;

Imagem 16: Nomeação de variáveis – exemplos de nomes de variáveis corretas e incorretas, variáveis corretas sem espaço em branco, variáveis incorretas com espaço em branco. Na imagem temos 4 quadros com declaração de variáveis. 1º quadro: "nome do aluno como caractere", 2º quadro: string data de nascimento. Esses dois quadros, posicionados à esquerda, são nomes de variáveis corretos, pois seguem as regras de nomenclatura de variáveis, ou seja, não pode ter espaço entre o nome da variável e o seu tipo, em Java primeiro declara-se o tipo de dado e depois o nome da variável. 3º e 4º quadro: string data_nascimento. Esses dois últimos quadros estão com um ticket vermelho à esquerda indicando que estão incorretos.

Reunião em "General"

28:30

Arquivo Página Inicial Inserir Layout da Página Páginas Iniciais Correspondências Revisão Edição

A LÓGICA APLICADA EM JAVA

Imagem 15: Tabela de representação de declaração de variáveis em fluxograma, pseudocódigo e codificação em Java – deve-se seguir o padrão de cada linguagem, ambos fluxograma e pseudocódigo as variáveis são declaradas da mesma forma, nome da variável e o seu tipo, em Java primeiro declara-se o tipo de dado e depois o nome da variável.

Como nomear uma variável?

Para garantir um rápido entendimento e continuidade do desenvolvimento do software, as variáveis devem ser nomeadas de forma objetiva, para esclarecer facilmente o programador sobre qual é a sua função.

Quando nomeamos as variáveis, é imprescindível também seguir algumas regras, que são:

- As variáveis nunca podem conter um espaço em seu nome

nome do aluno como caractere nomeAluno como caractere

String data_nascimento; String data_nascimento;

LUIZ PINHEIRO JUNIOR +35 HENRIQUE SANTANA ...

School R... Google T... trello.com/c/UBX9jPut/3-tpa Apps YouTube Gmail Classroom Grammaticy

Reunião em "General"

31:00

Technical Course Review Schedule

Trello workspace | Free Workspace view

TPA in list 7_Days

DUE DATE 16 May at 15:30

Description Add a more detailed description...

Topics

- Logic in Programming
- Reasoning Logic
- Programming Logic
- Algorithm Programming (Introduction)
- Programs, applications or software
- Object Oriented Programming
- Ways of Representing the Algorithm
- Flowchart and Use of Geometric Shapes
- Narrative, Flowchart and Portugol Description
- Pseudocode

+ Add another list

TPA Logic of Programming Reasoning Logic Programming Logic Algorithm Programming Programs, Applications or Softwares Object Oriented

Add card

nome do aluno como caractere ✗ nomeAluno como caractere ✓

String data_nascimento; ✗ String data_nascimento; ✓

Imagem 16: Exemplo de variáveis - exemplos de nomes de variáveis corretas e incorretas, variáveis corretas sem espaço em branco, variáveis incorretas com espaço em branco. A imagem tem 4 quadros, posicionados à esquerda: "nome do aluno como caractere", 2º quadro: string data de nascimento. Esses dois quadros, posicionados à esquerda, estão com um x vermelho indicando erro. Do lado direito temos mais dois quadros, 3º quadro: nomeAluno como caractere e o 4º quadro: string data_nascimento. Esses dois últimos quadros estão com um ticket verde à esquerda indicando que estão corretos.

Caso você necessite de mais de uma palavra para definir o nome de uma variável, junte as palavras ou então separe-as apenas com um underline.

Remova as preposições do nome da variável, assim o nome dela ficará mais objetivo e fácil de entender.

Não inclua mais do que duas palavras em um nome de variável, seja sempre objetivo.

A LÓGICA APlicada EM JAVA

Page: 14 de 31 | Palavras: 2,6,991 | 3 10 120%

LUIZ PINHEIRO JUNIOR +35 LOUISY DALCHIAVON ...

School R... Google T... trello.com/c/UBX9jPut/3-tpa Apps YouTube Gmail Classroom Grammaticy

Reunião em "General"

32:04

Technical Course Review Schedule

Trello workspace | Free Workspace view

TPA in list 7_Days

DUE DATE 16 May at 15:30

Description Add a more detailed description...

Topics

- Logic in Programming
- Reasoning Logic
- Programming Logic
- Algorithm Programming (Introduction)
- Programs, applications or software
- Object Oriented Programming
- Ways of Representing the Algorithm
- Flowchart and Use of Geometric Shapes
- Narrative,
- Pseudocode

+ Add another list

TPA Logic of Programming Reasoning Logic Programming Logic Algorithm Programming Programs, Applications or Softwares Object Oriented

Add card

masculino/feminino como caractere ✗ sexo como caractere ✓

int di@sem@n@; ✗ int diaSemana; ✓

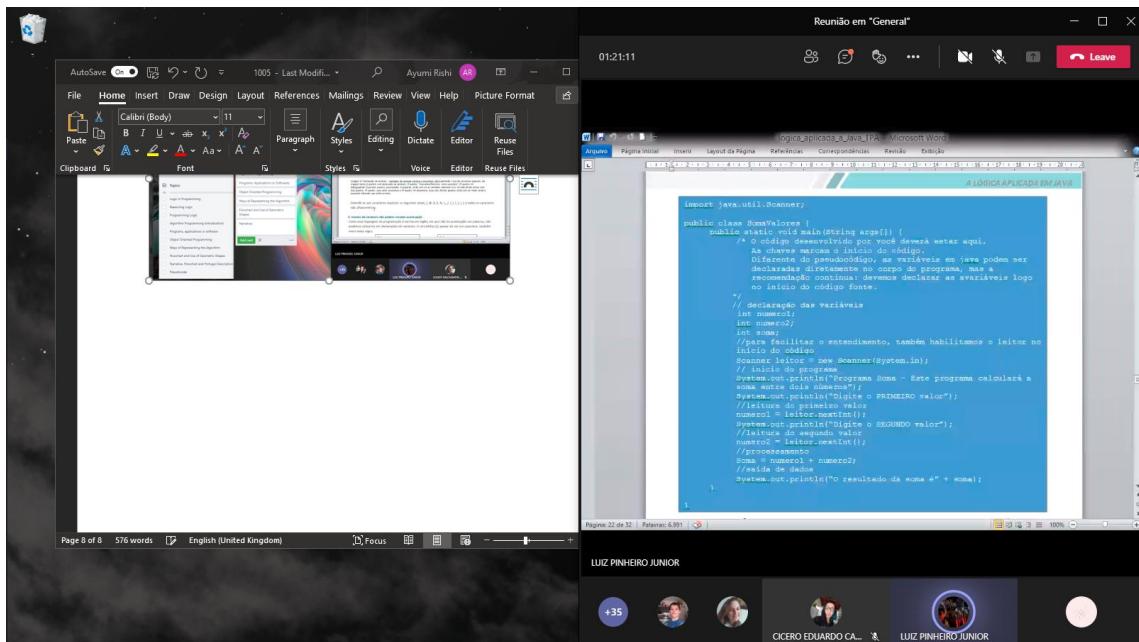
Imagem 17: Nomeação de variáveis - exemplos de variáveis corretas e incorretas, não é permitido o uso de caracteres especiais. Na imagem temos 4 quadros com exemplos de nomeação de variáveis. 1º quadro: "masculino/feminino" como caractere", 2º quadro: int di@sem@n@;. Esses dois quadros, posicionados à esquerda, estão com um x vermelho indicando erro. Do lado direito temos mais dois quadros, 3º quadro: sexo como caractere e o 4º quadro: int diaSemana;. Esses dois últimos quadros estão com um ticket verde à esquerda indicando que estão corretos.

Entende-se por caracteres especiais os seguintes símbolos: !, @, #, \$, %, ^, /, [,], { , }, e todos os caracteres não alfanuméricos.

3. Nomes de variáveis não podem receber acentuação

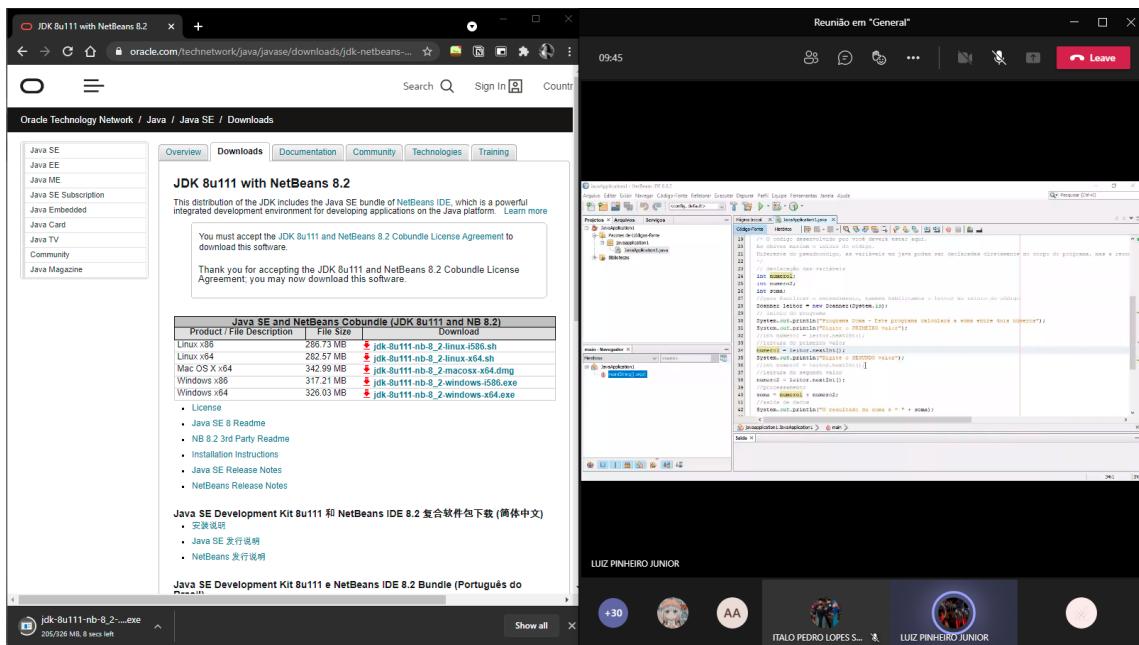
Como essa linguagem de programação é escrita em inglês, em que não há acentuação nas palavras, não podemos utilizá-las em declarações de variáveis. O círculo-cedilha (ç) apesar de ser um caractere, também entra nesta regra.

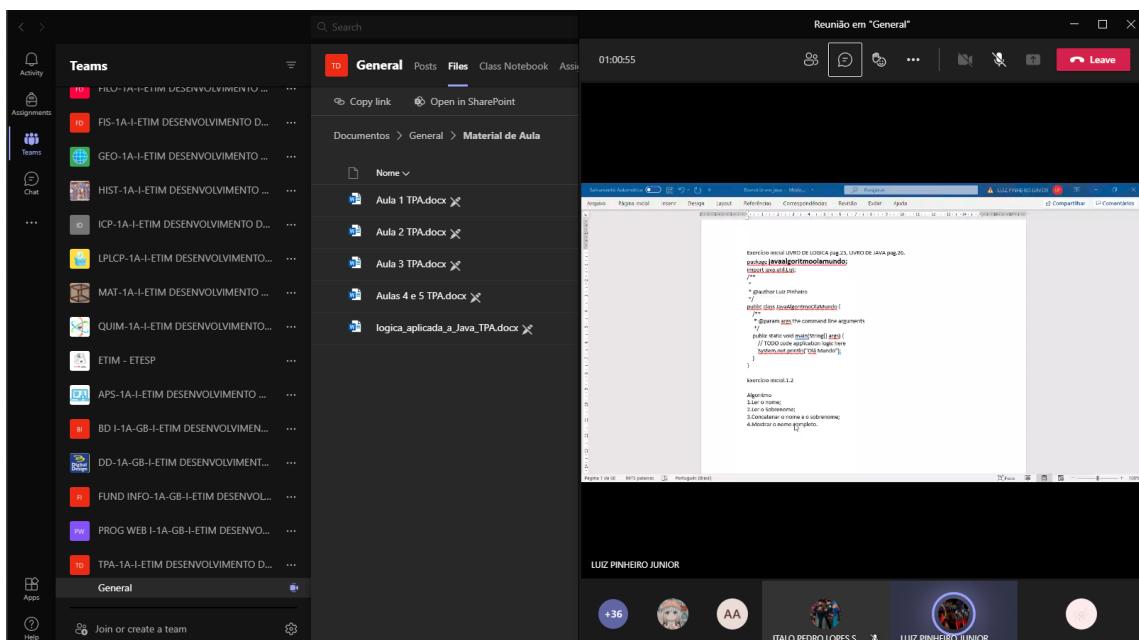
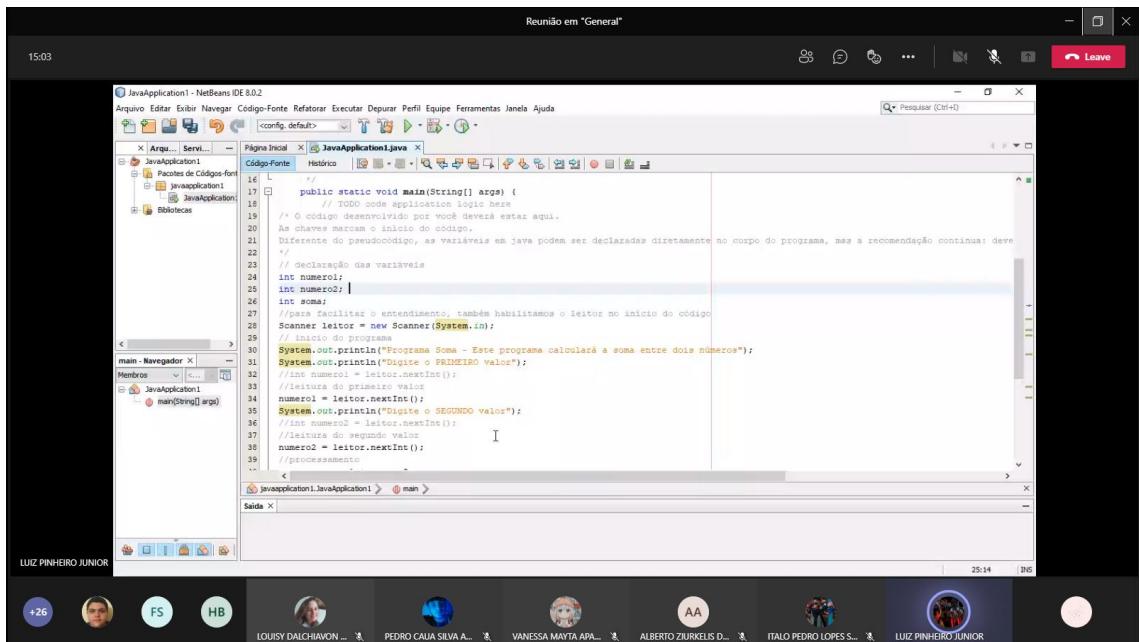
LUIZ PINHEIRO JUNIOR +35 LOUISY DALCHIAVON ...



Novo projeto no NetBeans: soma e valor, copia e cola o código e roda no NetBeans para ver se funciona.

1705





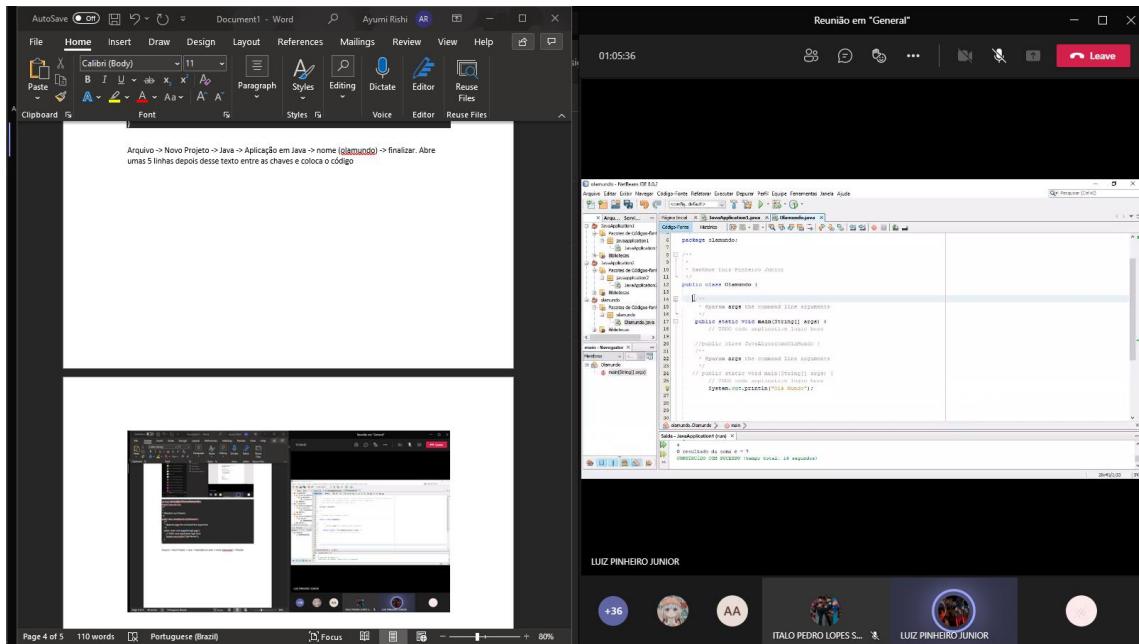
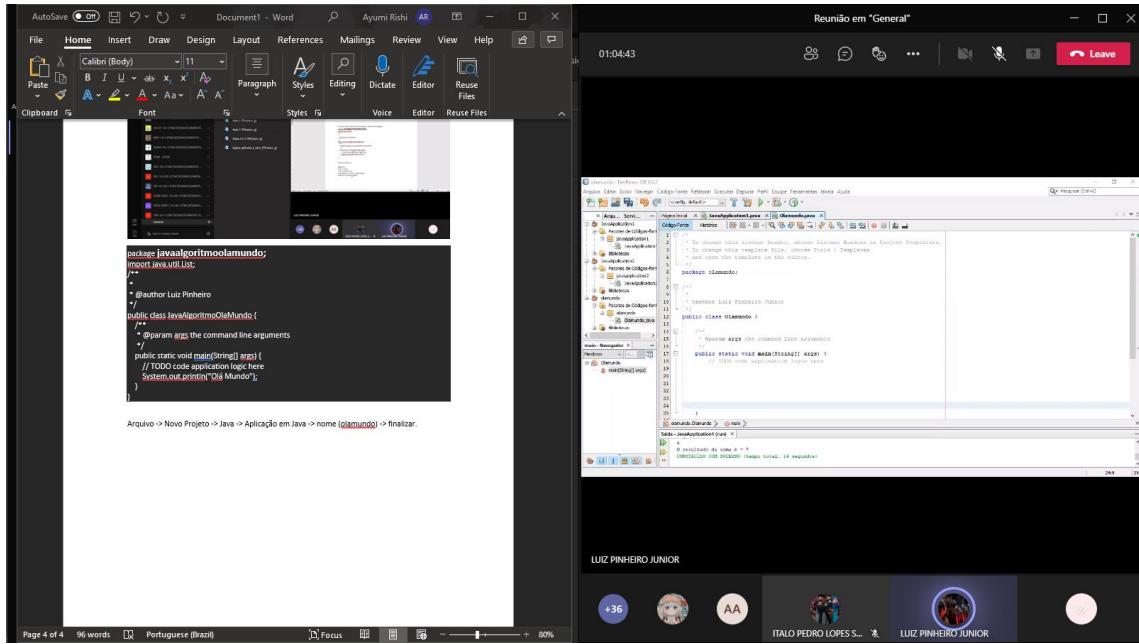
```

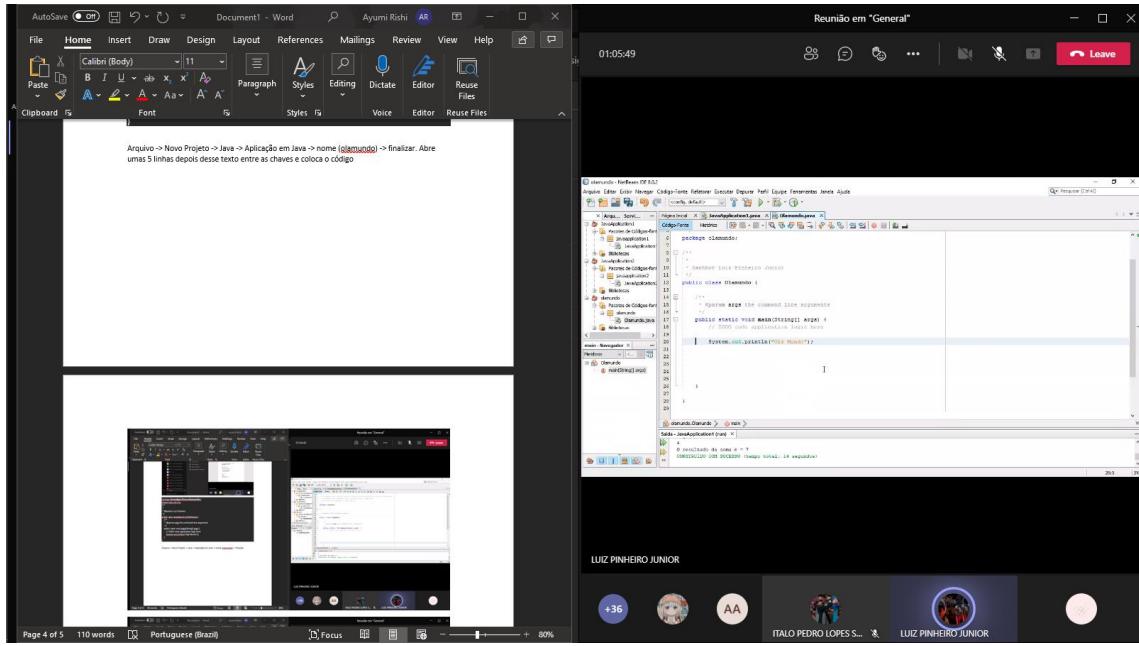
package javaalgoritmolamundo;
import java.util.List;
/**
 *
 * @author Luiz Pinheiro
 */
public class JavaAlgoritmoOlaMundo {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("Olá Mundo");
    }
}

```

}

Arquivo -> Novo Projeto -> Java -> Aplicação em Java -> nome (olamundo) -> finalizar. Abre umas 5 linhas depois desse texto entre as chaves e coloca o código





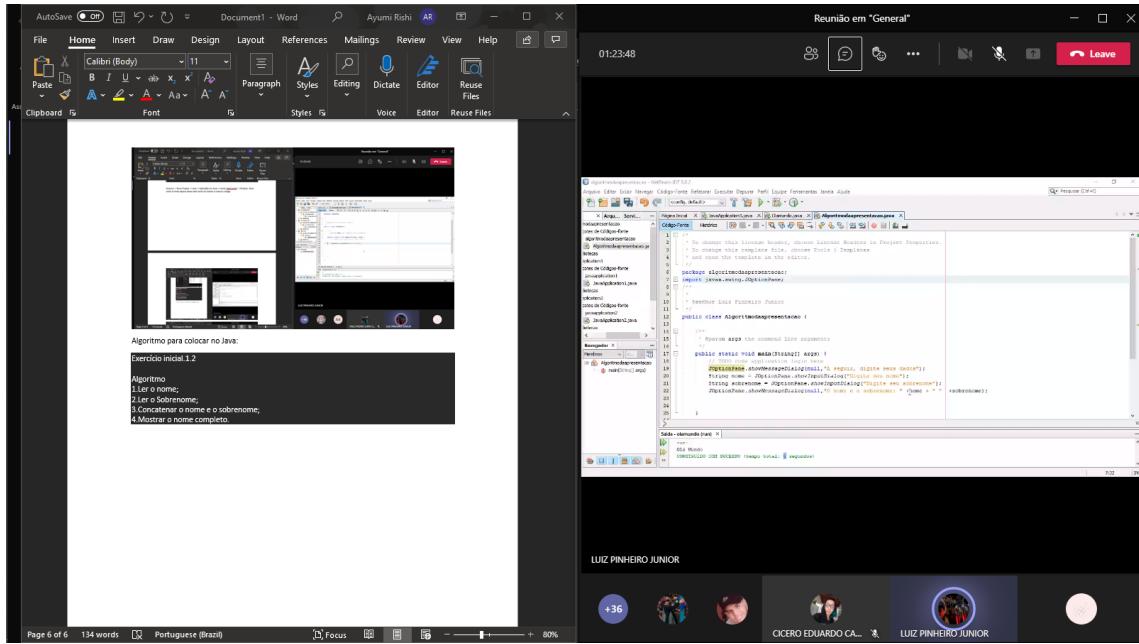
Algoritmo para colocar no Java:

Exercício inicial.1.2

Algoritmo

- 1.Ler o nome;
 - 2.Ler o Sobrenome;
 - 3.Concatenar o nome e o sobrenome;
 - 4.Mostrar o nome completo.

Outra biblioteca:



Código pronto:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package nomesobrenome;
import java.util.Scanner;
/**
 *
 * @author user
 */
public class NomeSobrenome {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        String nome;
        String sobrenome;

        Scanner leitor = new Scanner(System.in);

        System.out.println("Programa Nome e Sobrenome – Este programa concatenará seu
nome e sobrenome");
        System.out.println("Digite seu prmeiro nome");
        nome = leitor.next();
        System.out.println("Digite o seu sobrenome");
        sobrenome = leitor.next();
        System.out.println("O seu nome é " + nome + sobrenome);
    }
}
```

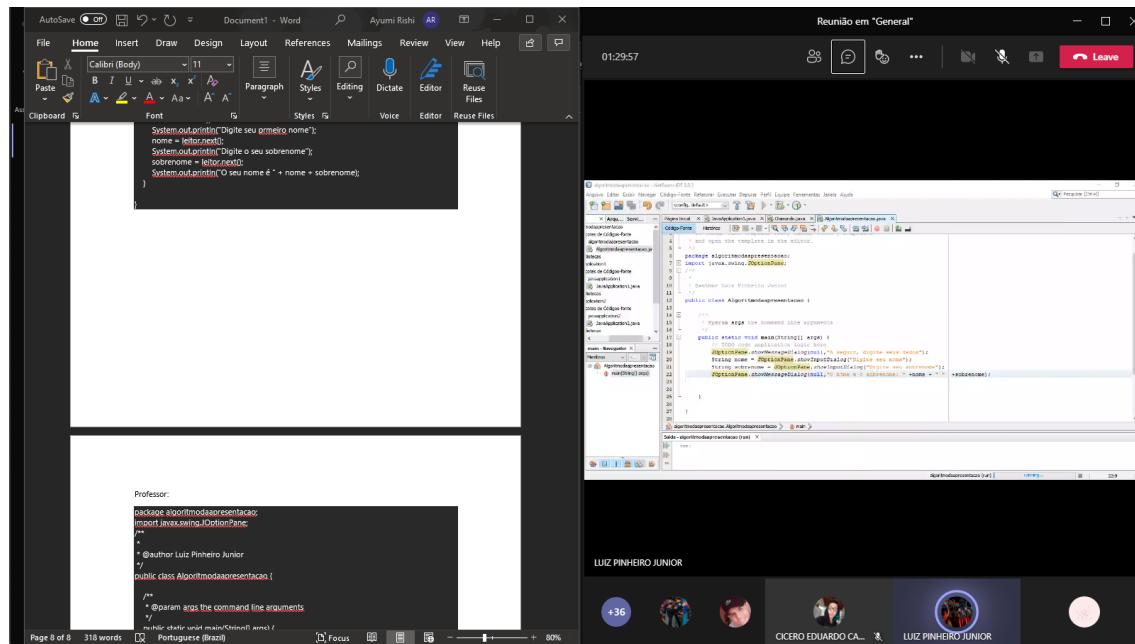
Professor:

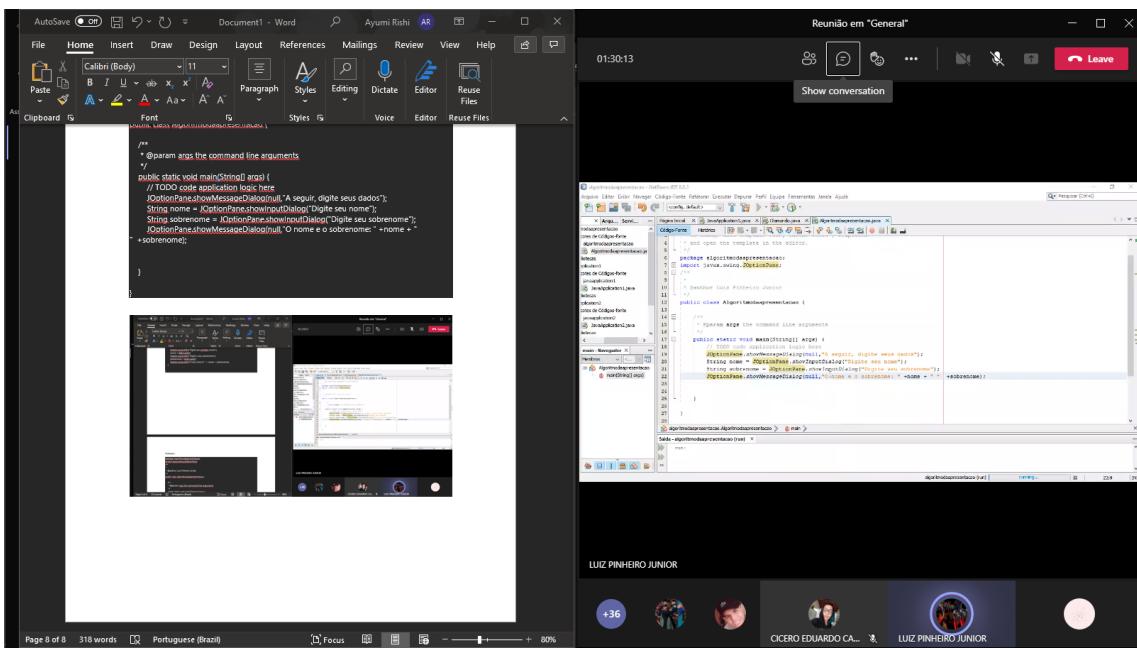
```
package algoritmodapresentacao;
import javax.swing.JOptionPane;
/**
 *
 * @author Luiz Pinheiro Junior
 */
public class Algoritmodapresentacao {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        JOptionPane.showMessageDialog(null,"A seguir, digite seus dados");
        String nome = JOptionPane.showInputDialog("Digite seu nome");
        String sobrenome = JOptionPane.showInputDialog("Digite seu sobrenome");
        JOptionPane.showMessageDialog(null,"O nome e o sobrenome: " +nome +
" "+sobrenome);

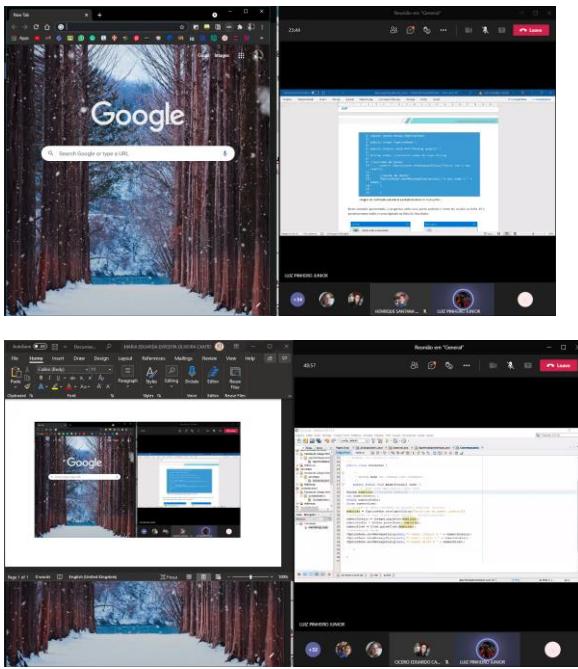
    }

}
```

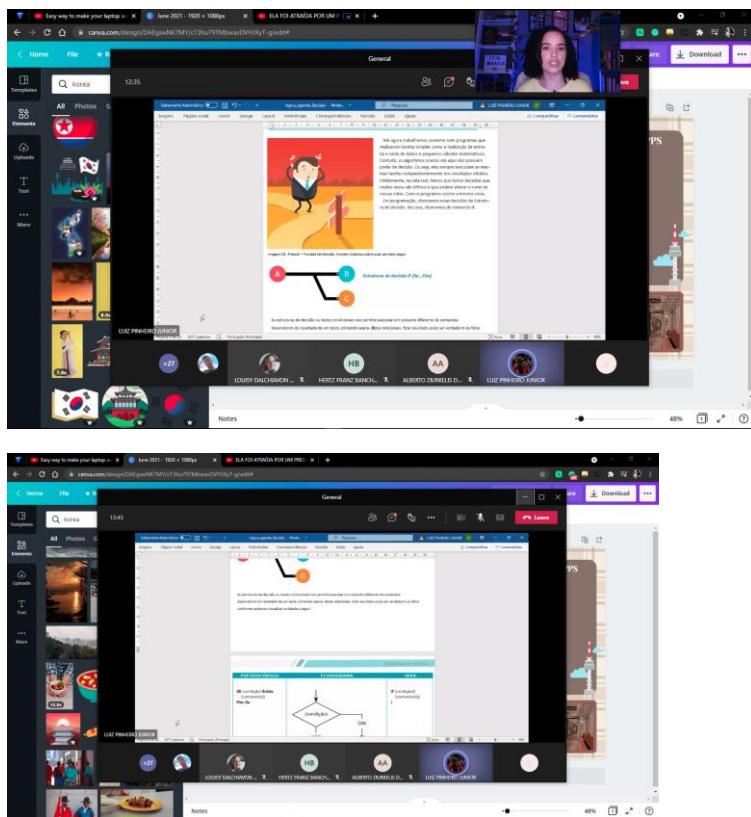




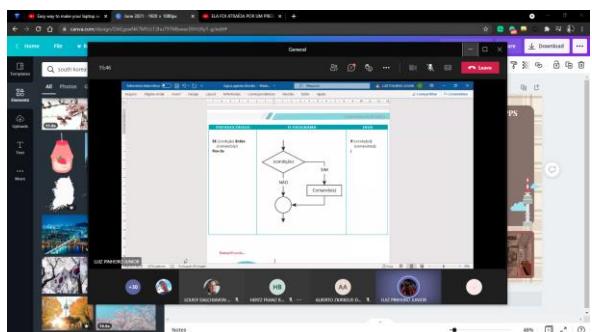
2405



0706



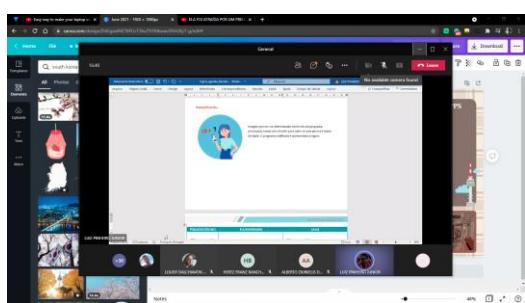
Sempre a priemria resposta n Ose É VERDADEIRA

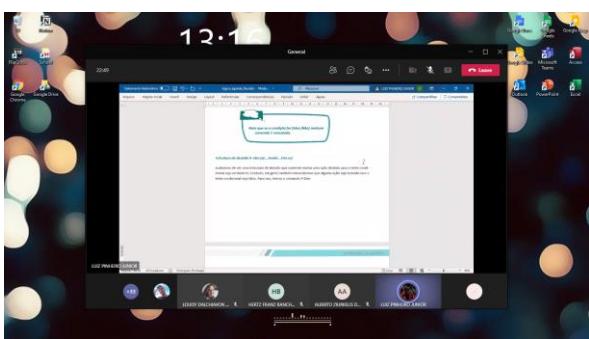
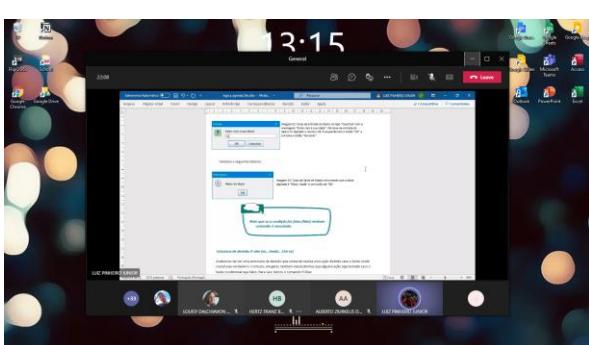
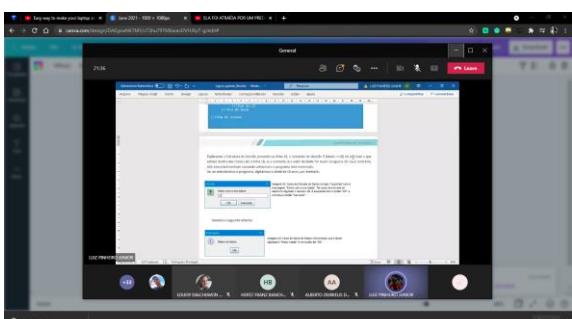
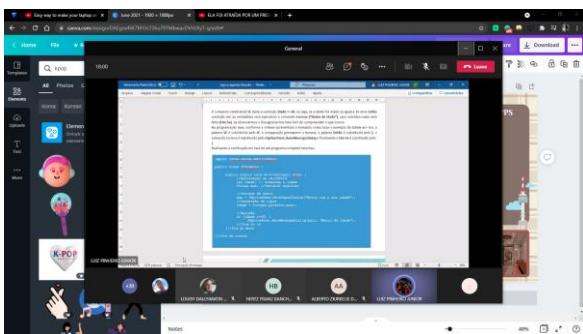
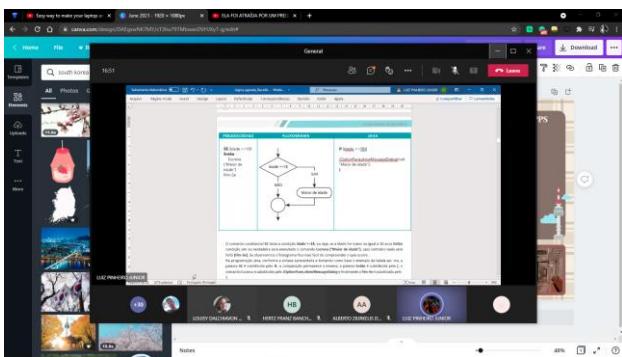


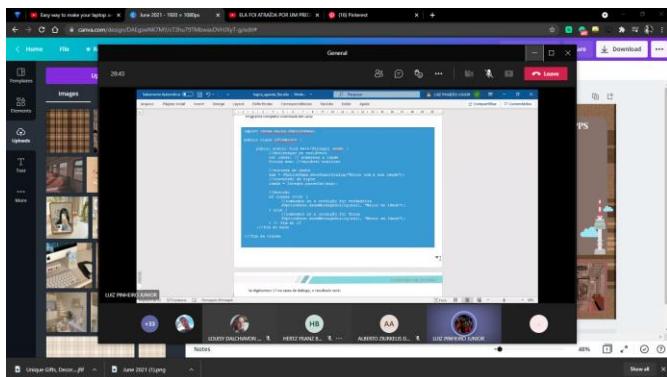
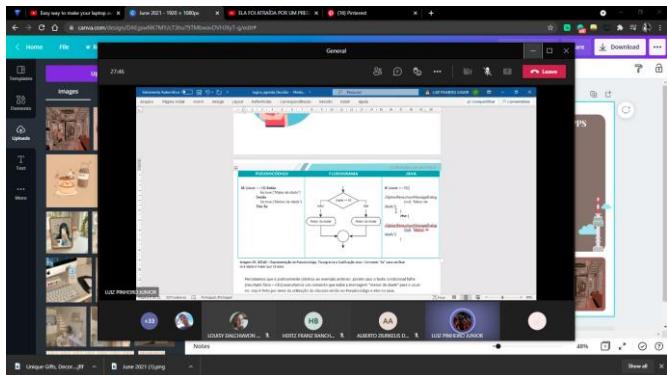
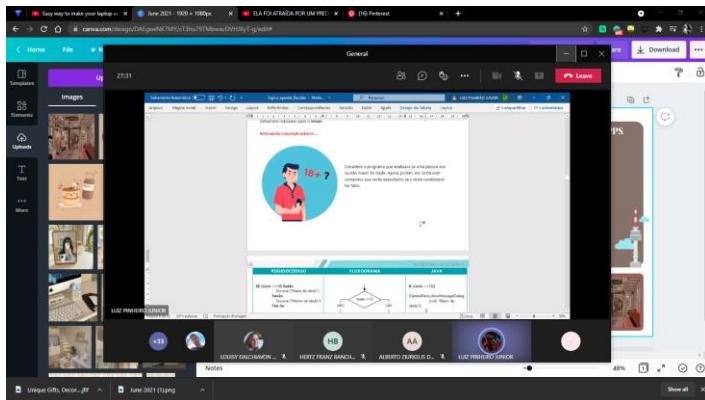
Losango – decisão

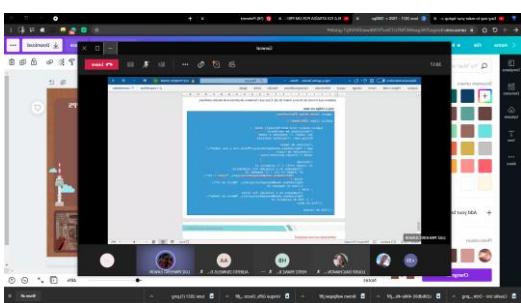
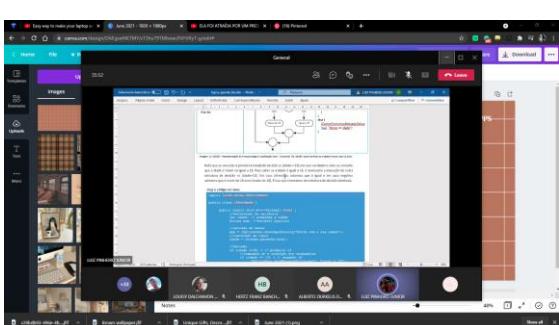
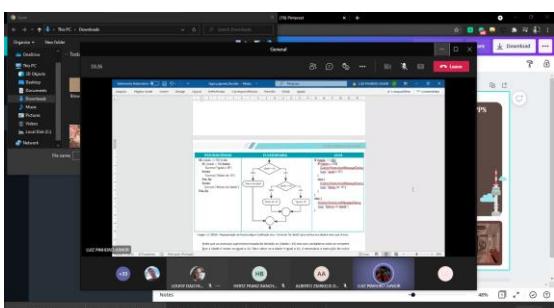
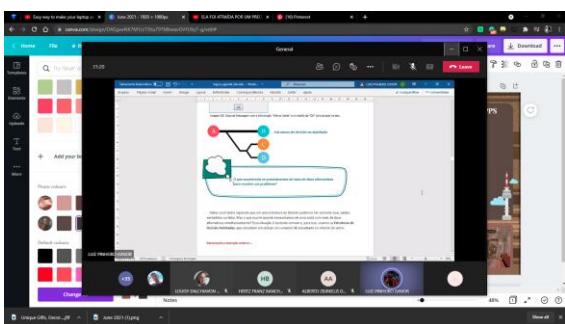
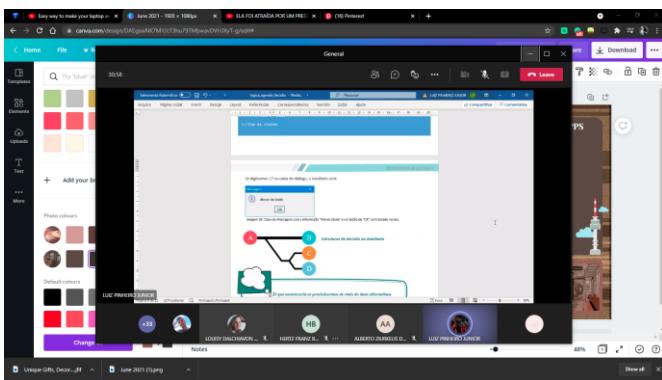
Processamento – retângulo

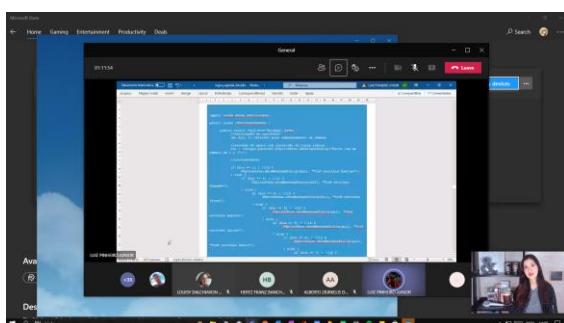
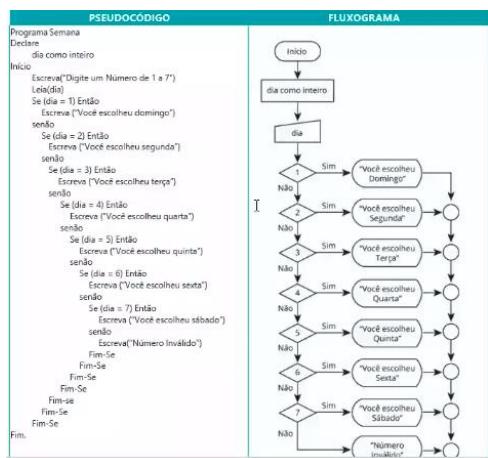
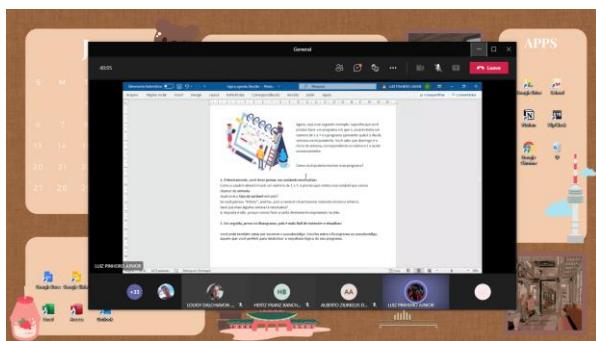
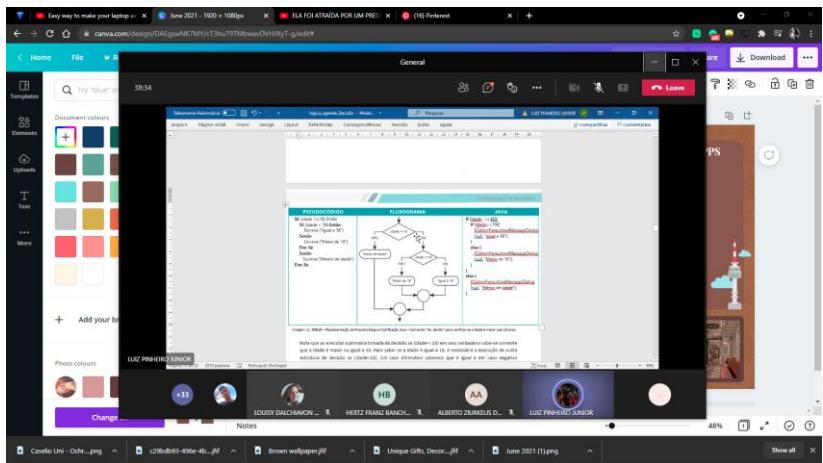
Bolinha – conexão de linha – linha não pode se cruzar na lógica

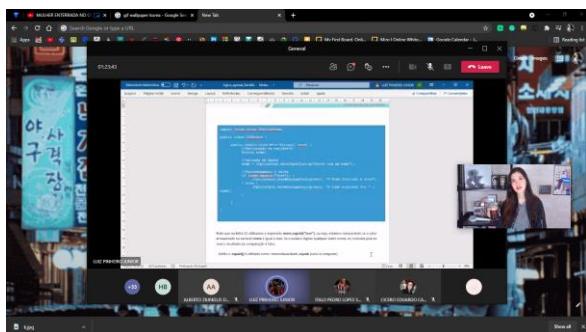
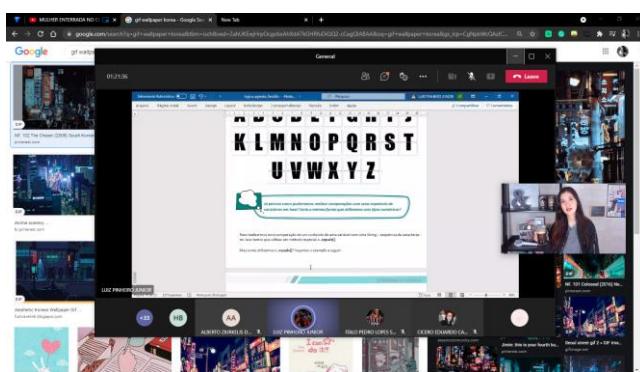
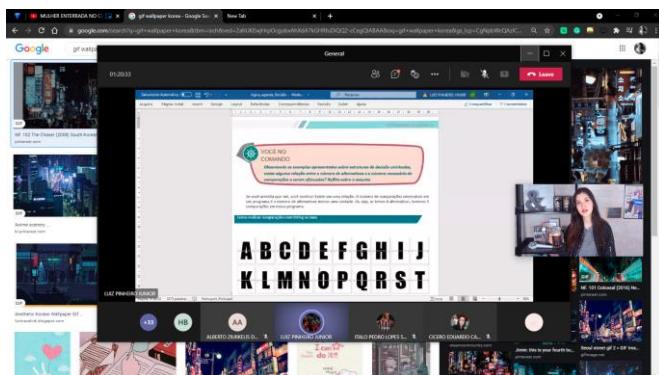
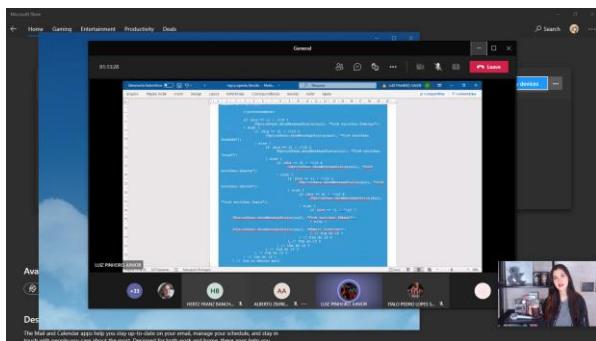
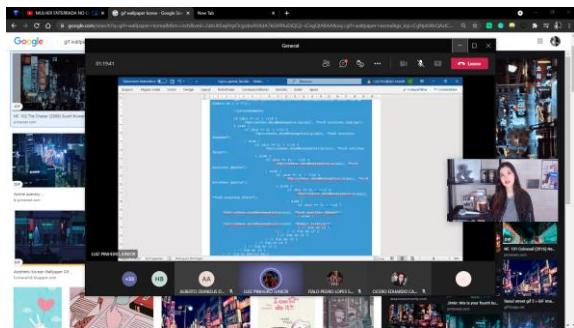




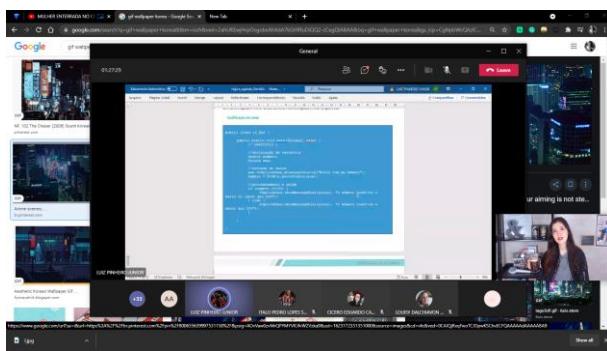
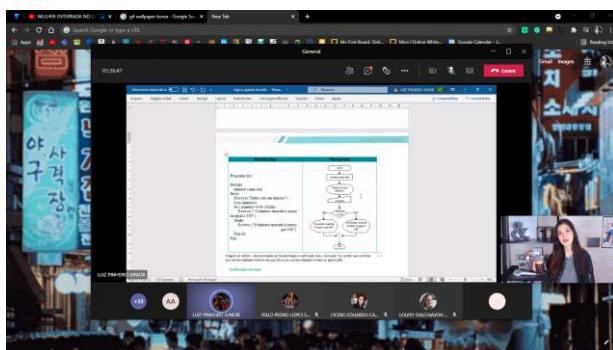
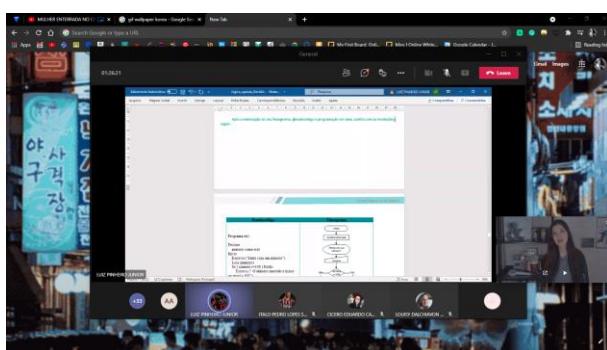
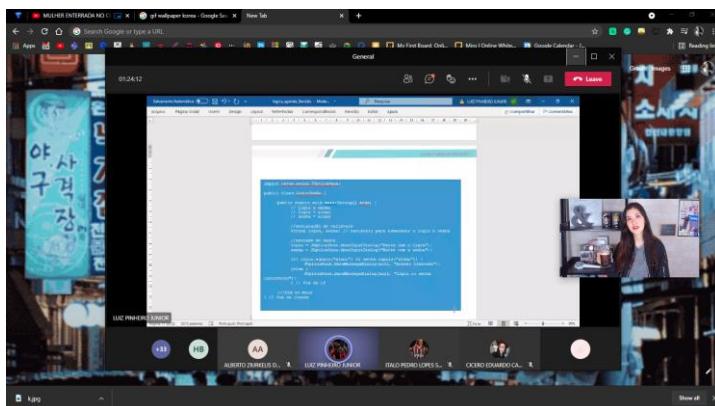


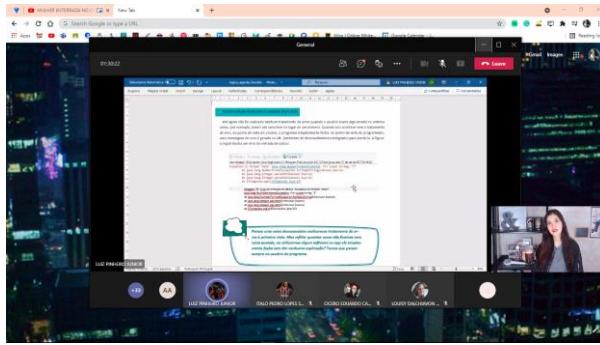






Login e senha:





1406

Relevo terrestre resulta da atuação dos agentes internos e externos. Os agentes internos são os agentes endógenos (endo = dentro) são aquelas impulsões pela energia contida dentro do planeta. Agentes externos são os agentes que originam grandes formações geológicas do planeta e continuam a agir. Os agentes externos são os agentes exógenos (exo = fora) eles atuam na modelagem na crosta terrestre, transforma as rochas... daí o relevo o aspecto que se apresenta atualmente, eles são chamados de agentes modeladores do relevo.

O relevo se transforma em três formas : intertemperismo (processo de desagregação é desintegração) - quando a rocha se desintegra e se fragmenta sem que haja movimento ou estar sujeito à erosão - o intertemperismo pode ser de três tipos: fúcsio (se dá em áreas muito secas como desertos), áreas semiáridas, chove pouco e lá o intertemperismo fílico predomina – o principal fator dele é a variação de temperatura de dia/noite, verão/inverno – calor as moléculas se expandem e frio contrai e acontece isso dia e noite, até que as moléculas se desagregam), químico (acontece em áreas chuvosas e resulta fundamentalmente da ação da água sobre as rochas, que a água vai modificando a composição química da rocha – ligado a água) e biológico (ligado a motivos como a raiz das plantas das rochas...). Ivro – pode ser por causa de algum fungo que pode destruir a rocha), transporte desse material (intertempo – os resíduos gerados pelo processo de desagregação são levados para a erosão, é só levando esse material removido modifica o relevo e pode virar solo se sedimentar) e sedimentação.

ASSOREAMENTO – um rio está assoreado que recebe muito material particulado da rocha e uma parte dos sedimentos desse transporte foi para o no.

Palavras fundamentais: a forma da terra se desgrega (com o passar do tempo a forma da terra muda – os agentes modeladores agem continuamente). EROSÃO –

The screenshot shows a Microsoft Word document with the following content:

Imagem 01: Freepik – Tomada de Decisão: Homem indeciso sobre qual caminho seguir

Estruturas de decisão IF (Se...Fim)

```
graph LR; A((A)) --> B((B)); B --> C((C))
```

As estruturas de decisão ou testes condicionais nos permitem executar um conjunto diferente de comandos dependendo do resultado de um teste utilizando operadores relacionais. Este resultado pode ser verdadeiro ou falso conforme podemos visualizar na tabela a seguir.

PSEUDOCODIGO

FLUXOGRAMA

JAVA

NHEIRO JUNIOR

ITALO PEDRO LOPES S...

LUIZ PINHEIRO JUNIOR

1406 - Saved - MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

TPA

DE DECISÃO II

PSEUDOCÓDIGO FLUXOGRAMA JAVA

```

SE (condição) Então
  Fim-SE
    ↓
    (condição)
    SIM → Comando(s)
    NÃO →
      ↓
      (condição)
      SIM → Comando(s)
      NÃO →
        ↓
        (condição)
        SIM → Comando(s)
        NÃO →
          ↓
          (condição)
          SIM → Comando(s)
          NÃO →
            ↓
            (condição)
            SIM → Comando(s)
            NÃO →
              ↓
              (condição)
              SIM → Comando(s)
              NÃO →
                ↓
                (condição)
                SIM → Comando(s)
                NÃO →
                  ↓
                  (condição)
                  SIM → Comando(s)
                  NÃO →
                    ↓
                    (condição)
                    SIM → Comando(s)
                    NÃO →
                      ↓
                      (condição)
                      SIM → Comando(s)
                      NÃO →
                        ↓
                        (condição)
                        SIM → Comando(s)
                        NÃO →
                          ↓
                          (condição)
                          SIM → Comando(s)
                          NÃO →
                            ↓
                            (condição)
                            SIM → Comando(s)
                            NÃO →
                              ↓
                              (condição)
                              SIM → Comando(s)
                              NÃO →
                                ↓
                                (condição)
                                SIM → Comando(s)
                                NÃO →
                                  ↓
                                  (condição)
                                  SIM → Comando(s)
                                  NÃO →
                                    ↓
                                    (condição)
                                    SIM → Comando(s)
                                    NÃO →
                                      ↓
                                      (condição)
                                      SIM → Comando(s)
                                      NÃO →
                                        ↓
                                        (condição)
                                        SIM → Comando(s)
                                        NÃO →
                                          ↓
                                          (condição)
                                          SIM → Comando(s)
                                          NÃO →
                                            ↓
                                            (condição)
                                            SIM → Comando(s)
                                            NÃO →
                                              ↓
                                              (condição)
                                              SIM → Comando(s)
                                              NÃO →
                                                ↓
                                                (condição)
                                                SIM → Comando(s)
                                                NÃO →
                                                  ↓
                                                  (condição)
                                                  SIM → Comando(s)
                                                  NÃO →
                                                    ↓
                                                    (condição)
                                                    SIM → Comando(s)
                                                    NÃO →
                                                      ↓
                                                      (condição)
                                                      SIM → Comando(s)
                                                      NÃO →
                                                        ↓
                                                        (condição)
                                                        SIM → Comando(s)
                                                        NÃO →
                                                          ↓
                                                          (condição)
                                                          SIM → Comando(s)
                                                          NÃO →
                                                            ↓
                                                            (condição)
                                                            SIM → Comando(s)
                                                            NÃO →
                                                              ↓
                                                              (condição)
                                                              SIM → Comando(s)
                                                              NÃO →
                                                                ↓
                                                                (condição)
                                                                SIM → Comando(s)
                                                                NÃO →
                                                                  ↓
                                                                  (condição)
                                                                  SIM → Comando(s)
                                                                  NÃO →
                                                                    ↓
                                                                    (condição)
                                                                    SIM → Comando(s)
                                                                    NÃO →
                                                                      ↓
                                                                      (condição)
                                                                      SIM → Comando(s)
                                                                      NÃO →
                                                                        ↓
                                                                        (condição)
                                                                        SIM → Comando(s)
                                                                        NÃO →
                                                                          ↓
                                                                          (condição)
                                                                          SIM → Comando(s)
                                                                          NÃO →
                                                                            ↓
                                                                            (condição)
                                                                            SIM → Comando(s)
                                                                            NÃO →
                                                                              ↓
                                                                              (condição)
                                                                              SIM → Comando(s)
                                                                              NÃO →
                                                                                ↓
                                                                                (condição)
                                                                                SIM → Comando(s)
                                                                                NÃO →
                                                                                  ↓
                                                                                  (condição)
                                                                                  SIM → Comando(s)
                                                                                  NÃO →
                                                                                    ↓
                                                                                    (condição)
                                                                                    SIM → Comando(s)
                                                                                    NÃO →
                                                                                      ↓
                                                                                      (condição)
                                                                                      SIM → Comando(s)
                                                                                      NÃO →
                        
```

NHEIRO JUNIOR

+23 HB ITALO PEDRO LOPES S... LUIZ PINHEIRO JUNIOR

1406 - Saved - MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File | C:/Users/maria%20eduarda/Downloads/2021... | Página 2 de 2 | 557 words | Portuguese (Brazil)

Reunião em "General"

2021 - LISTA DE EXERCÍCIOS 1 ANO 2 BIM.pdf

4) A função horária da velocidade de um móvel é $v = 16 - 2t$ (unidades do S.I.). Determine:

- a velocidade do móvel no instante 6 s.
- o instante em que o móvel muda o sentido de seu movimento.
- se no instante inicial o móvel estava na posição 12 m, escreva a função horária da posição desse móvel.
- em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2t + 3t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Atira-se em um poço uma pedra verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Desprezar a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

- as funções horárias do movimento.
- o tempo de subida, isto é, o tempo para atingir a altura máxima.
- A altura máxima.
- O espaço do móvel e o sentido do mesmo para o instante 6 s.
- O instante da chegada ao solo.
- A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreze a resistência do ar e determine ($g = 10 \text{ m/s}^2$):

- a altura do edifício.
- o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACELERADO

PSEUDOCÓDIGO FLUXOGRAMA JAVA

```

SE (idade >= 18)
  Então
    Escreva ("Maior de idade")
  Fim-SE
    ↓
    (idade >= 18)
    SIM → Maior de idade
    NÃO →
      ↓
      (idade >= 18)
      SIM → Maior de idade
      NÃO →
        ↓
        (idade >= 18)
        SIM → Maior de idade
        NÃO →
          ↓
          (idade >= 18)
          SIM → Maior de idade
          NÃO →
            ↓
            (idade >= 18)
            SIM → Maior de idade
            NÃO →
              ↓
              (idade >= 18)
              SIM → Maior de idade
              NÃO →
                ↓
                (idade >= 18)
                SIM → Maior de idade
                NÃO →
                  ↓
                  (idade >= 18)
                  SIM → Maior de idade
                  NÃO →
                    ↓
                    (idade >= 18)
                    SIM → Maior de idade
                    NÃO →
                      ↓
                      (idade >= 18)
                      SIM → Maior de idade
                      NÃO →
                        ↓
                        (idade >= 18)
                        SIM → Maior de idade
                        NÃO →
                          ↓
                          (idade >= 18)
                          SIM → Maior de idade
                          NÃO →
                            ↓
                            (idade >= 18)
                            SIM → Maior de idade
                            NÃO →
                              ↓
                              (idade >= 18)
                              SIM → Maior de idade
                              NÃO →
                                ↓
                                (idade >= 18)
                                SIM → Maior de idade
                                NÃO →
                                  ↓
                                  (idade >= 18)
                                  SIM → Maior de idade
                                  NÃO →
                                    ↓
                                    (idade >= 18)
                                    SIM → Maior de idade
                                    NÃO →
                                      ↓
                                      (idade >= 18)
                                      SIM → Maior de idade
                                      NÃO →
                                        ↓
                                        (idade >= 18)
                                        SIM → Maior de idade
                                        NÃO →
                                          ↓
                                          (idade >= 18)
                                          SIM → Maior de idade
                                          NÃO →
                                            ↓
                                            (idade >= 18)
                                            SIM → Maior de idade
                                            NÃO →
                                              ↓
                                              (idade >= 18)
                                              SIM → Maior de idade
                                              NÃO →
                                                ↓
                                                (idade >= 18)
                                                SIM → Maior de idade
                                                NÃO →
                                                  ↓
                                                  (idade >= 18)
                                                  SIM → Maior de idade
                                                  NÃO →
                                                    ↓
                                                    (idade >= 18)
                                                    SIM → Maior de idade
                                                    NÃO →
                                                      ↓
                                                      (idade >= 18)
                                                      SIM → Maior de idade
                                                      NÃO →
                                                        ↓
                                                        (idade >= 18)
                                                        SIM → Maior de idade
                                                        NÃO →
                                                          ↓
                                                          (idade >= 18)
                                                          SIM → Maior de idade
                                                          NÃO →
                                                            ↓
                                                            (idade >= 18)
                                                            SIM → Maior de idade
                                                            NÃO →
                                                              ↓
                                                              (idade >= 18)
                                                              SIM → Maior de idade
                                                              NÃO →
                                                                ↓
                                                                (idade >= 18)
                                                                SIM → Maior de idade
                                                                NÃO →
                                                                  ↓
                                                                  (idade >= 18)
                                                                  SIM → Maior de idade
                                                                  NÃO →
                                                                    ↓
                                                                    (idade >= 18)
                                                                    SIM → Maior de idade
                                                                    NÃO →
                                                                      ↓
                                                                      (idade >= 18)
                                                                      SIM → Maior de idade
                                                                      NÃO →
                                                                        ↓
                                                                        (idade >= 18)
                                                                        SIM → Maior de idade
                                                                        NÃO →
                                                                          ↓
                                                                          (idade >= 18)
                                                                          SIM → Maior de idade
                                                                          NÃO →
                                                                            ↓
                                                                            (idade >= 18)
                                                                            SIM → Maior de idade
                                                                            NÃO →
                                                                              ↓
                                                                              (idade >= 18)
                                                                              SIM → Maior de idade
                                                                              NÃO →
                                                                                ↓
                                                                                (idade >= 18)
                                                                                SIM → Maior de idade
                                                                                NÃO →
                                                                                  ↓
                                                                                  (idade >= 18)
                                                                                  SIM → Maior de idade
                                                                                  NÃO →
                                                                                    ↓
                                                                                    (idade >= 18)
                                                                                    SIM → Maior de idade
                                                                                    NÃO →
                                                                                      ↓
                                                                                      (idade >= 18)
                                                                                      SIM → Maior de idade
                                                                                      NÃO →
                        
```

NHEIRO JUNIOR

+24 HB ITALO PEDRO LOPES S... LUIZ PINHEIRO JUNIOR

1406 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO Reunião em "General"

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Paste A A A A A A A A A A A A

Font Calibri (Body) 11 Paragraph Styles Editing Dictate Editor Reuse Files

Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição

ESTRUTURAS DE DECISÃO

PSEUDOCÓDIGO	FLUXOGRAAMA	JAVA
<pre>SE (condição) Então [comando(s) condição verdadeira] Senão [comando(s) condição falsa] Fim-SE</pre>		<pre># (condição) [comando(s) condição verdadeira]; else { [comando(s) condição falsa]; }</pre>

Imagem 04: GEEAD – Representação de Pseudocódigo, Fluxograma e Codificação Java – Sintaxe do Comando "Se...Senão".

Observando a tabela acima, você pode notar que, caso o teste lógico condicional falte (ou seja, caso a condição não seja atendida), temos um comando ou grupo de comandos a serem executados: os comandos indicados após o **Senão**.

NHEIRO JUNIOR +25 HB ITALO PEDRO LOPES S... LUIZ PINHEIRO JÚNIOR

Page 3 of 3 557 words Portuguese (Brazil) Focus 90%

1406 - Saved MARIA EDUARDA EXPEDITA OLIVEIRA CANTO Reunião em "General"

File | C:/Users/maria%20eduarda/Downloads/2021... Focus 90%

2021 - LISTA DE EXERCÍCIOS 1 ANO 2 BIM.pdf

4) A função horária da velocidade de um móvel é $v = 16 - 2t$ (unidades do S.I.). Determine:
 a) a velocidade do móvel no instante 6 s.
 b) o instante em que o móvel muda o sentido de seu movimento.
 c) se no instante inicial o móvel estava na posição 12 m, escreva a função horária da posição desse móvel.
 d) em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2t + 3t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Atira-se em um poço verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Desprezar a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:
 a) as funções horárias do movimento.
 b) o tempo de subida, isto é, o tempo para atingir a altura máxima.
 c) A altura máxima.
 d) O espaço do móvel e o sentido do mesmo para o instante 6 s.
 e) O instante da chegada ao solo.
 f) A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreza a resistência do ar e determine ($g = 10 \text{ m/s}^2$):
 a) a altura do edifício.
 b) o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACCELERADO

ESTRUTURAS DE DECISÃO

PSEUDOCÓDIGO	FLUXOGRAAMA	JAVA
<pre>SE (idade >= 18) Então Escreva ("Maior de idade") Senão Escreva ("Menor de idade") Fim-SE</pre>		<pre># (idade >= 18){ JOptionPane.showMessageDialog (null, "Maior de idade"); } else { JOptionPane.showMessageDialog (null, "Menor de idade"); }</pre>

Imagem 05: GEEAD – Representação de Pseudocódigo, Fluxograma e Codificação Java – Comando "Se" para verificar se a idade é maior que 18 anos.

Perceberemos que é praticamente idêntico ao exemplo anterior, porém caso o teste condicional falle (resultado falso – não) executamos um comando que exibe a mensagem "menor de idade" para o usuário. Isso é feito por meio da utilização da cláusula **Senão** no Pseudocódigo, e **else** no Java.

Programa completo codificado em Java:

NHEIRO JUNIOR +25 HB ITALO PEDRO LOPES S... LUIZ PINHEIRO JÚNIOR

de 22 | Páginas: 3.272 | Focus 90%

1406 - Saved

MARIA EDUarda EXPEDITA OLIVEIRA CANTO

2021 - LISTA DE EXERCÍCIOS 1 ANO 2 BIM.pdf

4) A função horária da velocidade de um móvel é $v = 16 - 2.t$ (unidades do S.I.). Determine:

- a velocidade do móvel no instante 6 s.
- o instante em que o móvel muda o sentido de seu movimento.
- se no instante inicial o móvel estava na posição 12 m, escreva a função horária da posição deste móvel.
- em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2.4 + 3.t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Atira-se em um poço uma pedra verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Despreze a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

- as funções horárias do movimento.
- o tempo de subida, isto é, o tempo para atingir a altura máxima.
- A altura máxima.
- O espaço do móvel e o sentido do mesmo para o instante 6 s.
- O instante da chegada ao solo.
- A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreze a resistência do ar e determine ($g = 10 \text{ m/s}^2$):

- a altura do edifício.
- o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACCELERADO

Imagem 05: Caixa de Mensagem com a informação "Menor idade" e um botão de "OK" centralizado na tela.

Talvez você tenha reparado que em uma Estrutura de Decisão podemos ter somente duas saídas: verdadeiro ou falso. Mas o que ocorre quando necessitamos de uma saída com mais de duas alternativas simultaneamente? Essa situação é bastante comum e, para isso, usamos as Estruturas de Decisão Aninhadas, que consistem em utilizar um comando SE encadeado no interior de outro.

Louisy Dalchavon Tomaz 1DS - Lista Exercícios 2BIM.pdf

11:59

5) Um móvel obedece à função horária $s = -4 + 2.4 + 3.t^2$, medidos no S.I. Determine sua posição em que o móvel se encontra no instante 5 s.

$$s = -4 + 2(5) + 3(5)^2$$

$$s = -4 + 2(5) + 3(25)$$

$$s = 4 + 75$$

$$s = 81 \text{ m}$$

6) Atira-se em um poço uma pedra calmamente para baixo com uma velocidade de inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

$$s = s_0 + v_0 t + \frac{1}{2} a t^2$$

$$s = 0 + 10(2) + \frac{1}{2}(10)(2)^2$$

$$s = 20 + 20$$

$$s = 40 \text{ m}$$

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Despreze a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

PSEUDOCÓDIGO

```
SE Idade >= 18
  SE Idade < 18
    Escreve ("Idade > 18")
    Senão
      Escreve ("Idade de 18")
    Fim Se
    Senão
      Escreve ("Menor de idade")
    Fim Se
```

FLUXOGRAAMA

JAVA

```
# Idade >= 18{
  JOptionPane.showMessageDialog(
    null, "Idade > 18");
}
else {
  JOptionPane.showMessageDialog(
    null, "Idade de 18");
}
else if {
  JOptionPane.showMessageDialog(
    null, "Menor de idade");
}
```

Imagem 11: GETS& - Representação de Pseudocódigo e Codificação Java - Comando "Se, Senão" para verificar se a idade é maior que 18 anos.

Note que ao executar a primeira tomada de decisão se (idade >= 18) em caso verdadeiro sabe-se imediatamente que a idade é maior ou igual a 18. Para saber se a idade é igual a 18, é necessária a execução de outra

LUIZ PINHEIRO JUNIOR

Louisy Dalchiavon Tomaz 1DS. Lista Exercícios 2BIM.pdf

12:31

5) Um móvel desce à função horária $S = -4t^2 + 3t^2$, medidas no S.I. Determine sua posição em que o móvel se encontra no instante 5s.

$$S = -4t^2 + 3t^2$$

$$S = -4(5)^2 + 3(5)^2$$

$$S = -4(25) + 3(25)$$

$$S = -100 + 75$$

$$S = 75 \text{ m}$$

R: Ele estará na 75 m .

6) Atira-se em um poço uma pedra calmamente para baixo com uma velocidade inicial igual a 10 m/s . A pedra demora 2 segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s^2 . Qual a profundidade do poço?

$$S = 50 + V_0 t + \frac{1}{2} a t^2$$

$$S = 50 + 10(2) + \frac{1}{2}(10)(2)^2$$

$$S = 50 + 20 + 20$$

$$S = 90 \text{ m}$$

R: A profundidade do poço é de 90 m .

7) Um móvel é atirado verticalmente para cima, a partir da terra, com velocidade de 50 m/s . Desprezando a resistência do ar, $g = 10 \text{ m/s}^2$. Determine:

Louisy Dalchiavon Tomaz 1DS. Lista Exercícios 2BIM.pdf

ESTRUTURAS DE DECISÃO

PSEUDOCÓDIGO

```

Programa Semana
    Declarar
        dia como inteiro
    Início
        Escreva "Digite um Número de 1 a 7"
        Leia(dia)
        Se (dia = 1) Então
            Escreva ("Você escolheu domingo")
            senão
                Se (dia = 2) Então
                    Escreva ("Você escolheu segunda")
                    senão
                        Se (dia = 3) Então
                            Escreva ("Você escolheu terça")
                            senão
                                Se (dia = 4) Então
                                    Escreva ("Você escolheu quarta")
                                    senão
                                        Se (dia = 5) Então
                                            Escreva ("Você escolheu quinta")
                                            senão
                                                Se (dia = 6) Então
                                                    Escreva ("Você escolheu sexta")
                                                    senão
                                                        Se (dia = 7) Então
                                                            Escreva ("Número Inválido")
                                                            Fim-Se
                                                        Fim-Se
                                                    Fim-Se
                                                Fim-Se
                                            Fim-Se
                                        Fim-Se
                                    Fim-Se
                                Fim-Se
                            Fim-Se
                        Fim-Se
                    Fim-Se
                Fim-Se
            Fim-Se
        Fim-Se
    Fim-Se

```

FLUXOGRAMA

Louisy Dalchiavon Tomaz 1DS. Lista Exercícios 2BIM.pdf

12:59

5) Um móvel desce à função horária $S = -4t^2 + 3t^2$, medidas no S.I. Determine sua posição em que o móvel se encontra no instante 5s.

$$S = -4t^2 + 3t^2$$

$$S = -4(5)^2 + 3(5)^2$$

$$S = -4(25) + 3(25)$$

$$S = -100 + 75$$

$$S = 75 \text{ m}$$

R: Ele estará na 75 m .

6) Atira-se em um poço uma pedra calmamente para baixo com uma velocidade inicial igual a 10 m/s . A pedra demora 2 segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s^2 . Qual a profundidade do poço?

$$S = 50 + V_0 t + \frac{1}{2} a t^2$$

$$S = 50 + 10(2) + \frac{1}{2}(10)(2)^2$$

$$S = 50 + 20 + 20$$

$$S = 90 \text{ m}$$

R: A profundidade do poço é de 90 m .

7) Um móvel é atirado verticalmente para cima, a partir da terra, com velocidade de 50 m/s . Desprezando a resistência do ar, $g = 10 \text{ m/s}^2$. Determine:

Louisy Dalchiavon Tomaz 1DS. Lista Exercícios 2BIM.pdf

ESTRUTURAS DE DECISÃO

PSEUDOCÓDIGO

```

Programa Semana
    Declarar
        dia como inteiro
    Início
        Escreva "Digite um Número de 1 a 7"
        Leia(dia)
        Se (dia = 1) Então
            Escreva ("Você escolheu domingo")
            senão
                Se (dia = 2) Então
                    Escreva ("Você escolheu segunda")
                    senão
                        Se (dia = 3) Então
                            Escreva ("Você escolheu terça")
                            senão
                                Se (dia = 4) Então
                                    Escreva ("Você escolheu quarta")
                                    senão
                                        Se (dia = 5) Então
                                            Escreva ("Você escolheu quinta")
                                            senão
                                                Se (dia = 6) Então
                                                    Escreva ("Você escolheu sexta")
                                                    senão
                                                        Se (dia = 7) Então
                                                            Escreva ("Número Inválido")
                                                            Fim-Se
                                                        Fim-Se
                                                    Fim-Se
                                                Fim-Se
                                            Fim-Se
                                        Fim-Se
                                    Fim-Se
                                Fim-Se
                            Fim-Se
                        Fim-Se
                    Fim-Se
                Fim-Se
            Fim-Se
        Fim-Se
    Fim-Se

```

FLUXOGRAMA

4) A função horária da velocidade de um móvel é $v = 16 - 2t$ (unidades do S.I.). Determine:

- a velocidade do móvel no instante 6 s.
- o instante em que o móvel muda o sentido de seu movimento.
- se no instante o móvel estava na posição 12 m, escreva a função horária da posição deste móvel.
- em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2t + 3t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Atira-se em um poço uma pedra verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Desprezar a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

- as funções horárias do movimento.
- o tempo de subida, isto é, o tempo para atingir a altura máxima.
- A altura máxima.
- O espaço do móvel e o sentido do mesmo para o instante 6 s.
- O instante da chegada ao solo.
- A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreze a resistência do ar e determine ($g = 10 \text{ m/s}^2$):

- a altura do edifício.
- o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACERLADO

Note na linha 13 utilizamos a expressão `nome.equals("jose")`, ou seja, estamos comparando se o valor armazenado na variável nome é igual a jose. Se o usuário digitar qualquer outro nome, ou inclusive jose ou José o resultado da comparação é falso.

```
import javax.swing.JOptionPane;
public class IfEqual {
    public static void main(String[] args) {
        //Declaração da variável
        String nome;
        //Entrada de dados
        nome = JOptionPane.showInputDialog("Entre com um nome");
        //Processamento e saída
        if (nome.equals("jose")) {
            JOptionPane.showMessageDialog(null, "O Nome Digitado é Jose");
        } else {
            JOptionPane.showMessageDialog(null, "O nome digitado foi " + nome);
        }
    }
}
```

LUIZ PINHEIRO JUNIOR

+30 HB ITALO PEDRO LOPEZ S... LUIZ PINHEIRO JUNIOR

4) A função horária da velocidade de um móvel é $v = 16 - 2t$ (unidades do S.I.). Determine:

- a velocidade do móvel no instante 6 s.
- o instante em que o móvel muda o sentido de seu movimento.
- se no instante o móvel estava na posição 12 m, escreva a função horária da posição deste móvel.
- em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2t + 3t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Atira-se em um poço uma pedra verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Desprezar a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

- as funções horárias do movimento.
- o tempo de subida, isto é, o tempo para atingir a altura máxima.
- A altura máxima.
- O espaço do móvel e o sentido do mesmo para o instante 6 s.
- O instante da chegada ao solo.
- A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreze a resistência do ar e determine ($g = 10 \text{ m/s}^2$):

- a altura do edifício.
- o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACERLADO

Note na linha 13 utilizamos a expressão `nome.equals("jose")`, ou seja, estamos comparando se o valor armazenado na variável nome é igual a jose. Se o usuário digitar qualquer outro nome, ou inclusive jose ou José o resultado da comparação é falso.

VOCÊ NO COMANDO

Como poderíamos elaborar um programa que faça a leitura de um nome de usuário e uma senha e libere o acesso ao aplicativo somente se ambos estiverem corretos?

Após elaborar o seu programa, confira no quadro a seguir se você codificou a situação corretamente.

LUIZ PINHEIRO JUNIOR

+30 HB ITALO PEDRO LOPEZ S... LUIZ PINHEIRO JUNIOR

4) A função horária da velocidade de um móvel é $v = 16 - 2t$ (unidades do S.I.). Determine:

- a velocidade do móvel no instante 6 s.
- o instante em que o móvel muda o sentido de seu movimento.
- se no instante inicial o móvel estava na posição 12 m, escreva a função horária da posição deste móvel.
- em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2t + 3t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Atira-se em um poço uma pedra verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Desprezar a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

- as funções horárias do movimento.
- o tempo de subida, isto é, o tempo para atingir a altura máxima.
- A altura máxima.
- O espaço do móvel e o sentido do mesmo para o instante 6 s.
- O instante da chegada ao solo.
- A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreze a resistência do ar e determine ($g = 10 \text{ m/s}^2$):

- a altura do edifício.
- o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACERELADO

Reunião em "General"

16:08

lógica_agenda_Decisão - Microsoft Word

```
import javax.swing.JOptionPane;
public class LoginSenha {
    public static void main(String[] args) {
        // login e senha
        String login, senha; // variáveis para armazenar o login e senha
        // entrada de dados
        login = JOptionPane.showInputDialog("Entre com o login");
        senha = JOptionPane.showInputDialog("Entre com a senha");
        if( login.equals("Aluno") || senha.equals("Aluno") ) {
            JOptionPane.showMessageDialog(null, "Acesso liberado");
        } else {
            JOptionPane.showMessageDialog(null, "Login ou senha incorreta");
        }
    }
}
```

VOCÊ NO COMANDO

Elabore um fluxograma e um programa em Java que leia um número e compare se ele é maior ou igual a 100.

Página 17 de 22 | Palavras: 3.272 |

LUIZ PINHEIRO JUNIOR

+30 HB ITALO PEDRO LOPEZ S... LUIZ PINHEIRO JUNIOR

4) A função horária da velocidade de um móvel é $v = 16 - 2t$ (unidades do S.I.). Determine:

- a velocidade do móvel no instante 6 s.
- o instante em que o móvel muda o sentido de seu movimento.
- se no instante inicial o móvel estava na posição 12 m, escreva a função horária da posição deste móvel.
- em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2t + 3t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Atira-se em um poço uma pedra verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Desprezar a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

- as funções horárias do movimento.
- o tempo de subida, isto é, o tempo para atingir a altura máxima.
- A altura máxima.
- O espaço do móvel e o sentido do mesmo para o instante 6 s.
- O instante da chegada ao solo.
- A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreze a resistência do ar e determine ($g = 10 \text{ m/s}^2$):

- a altura do edifício.
- o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACERELADO

Reunião em "General"

16:41

lógica_agenda_Decisão - Microsoft Word

ESTRUTURAS DE DECISÃO

Processamento	Fluxograma
Programa ex1 Declare numero como real Inicie: Exiba ("Entre com um número") Leia (numero) Se (numero > 100) Então Mensagem ("O numero inserido é maior ou igual a 100") Senão Exiba ("O numero inserido é menor que 100") Fim-SE Fim.	

Imagem 17: GEEdo - Representação do Pseudocódigo e Codificação Java - Comando "Se...Senão" para verificar se o número digitado é menor do que 100 ou se o número digitado é maior ou igual a 100.

Codificação em Java:

```
if(numero < 100) {
    System.out.println("O numero inserido é menor que 100");
} else {
    System.out.println("O numero inserido é maior ou igual a 100");
}
```

Página 18 de 22 | Palavras: 3.272 |

LUIZ PINHEIRO JUNIOR

+30 HB ITALO PEDRO LOPEZ S... LUIZ PINHEIRO JUNIOR

2021 - LISTA DE EXERCÍCIOS 1 ANO 2 BIM.pdf

4) A função horária da velocidade de um móvel é $v = 16 - 2t$ (unidades do S.I.). Determine:

- a velocidade do móvel no instante 6 s.
- o instante em que o móvel muda o sentido de seu movimento.
- se no instante o móvel estava na posição 12 m, escreva a função horária da posição deste móvel.
- em que posição este móvel estará no instante 3 s.

5) Um móvel obedece à função horária $s = -4 + 2t + 3t^2$, medidos no S.I. Determine sua posição em que o móvel se encontrará no instante 5 s.

6) Afira-se em um poço uma pedra verticalmente para baixo com uma velocidade inicial igual a 10 m/s. A pedra demora dois segundos para chegar ao fundo do poço, caindo com aceleração de 10 m/s². Qual a profundidade do poço?

7) Um móvel é atirado verticalmente para cima, a partir do solo, com velocidade inicial de 50 m/s. Despreze a resistência do ar. Dado: $g = 10 \text{ m/s}^2$. Determine:

- as funções horárias do movimento.
- o tempo de subida, isto é, o tempo para atingir a altura máxima.
- A altura máxima.
- O espaço do móvel e o sentido do mesmo para o instante 6 s.
- O instante da chegada ao solo.
- A velocidade escalar do móvel quando ele atingir o solo.

8) Abandona-se uma pedra do alto de um edifício e esta atinge o solo 4s depois. Despreze a resistência do ar e determine ($g = 10 \text{ m/s}^2$):

- a altura do edifício.
- o módulo da velocidade da pedra quando atinge o solo.

9) Para cada caso abaixo, desenhe os vetores velocidade e aceleração correspondentes a um

a) PROGRESSIVO E ACCELERADO

1406 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Até agora não foi realizado nenhum tratamento de erro quando o usuário inserir algo errado no sistema como, por exemplo, inserir um caractere no lugar de um número. Quando isso acontece sem o tratamento de erro, do ponto de vista do usuário, o programa simplesmente fecha, do ponto de vista do programador, uma mensagem de erro é gerada no IDE (ambiente de desenvolvimento integrado) para alertá-lo. A figura a seguir ilustra um erro de entrada de dados:

Imagem 20: Erro de entrada de dados. Exception in thread "main"

Parece uma coisa desnecessária realizarmos tratamento de erros à primeira vista. Mas reflita: quantas vezes não ficamos com raiva quando, ao utilizarmos algum software ou app ele simplesmente fecha sem dar nenhuma explicação? Temos que pensar sempre no usuário do programa.

Mas, o problema realmente é do usuário que fica sem saber o que aconteceu. Com certeza, algum

The image shows a Microsoft Teams interface on the left and a Microsoft Word document on the right, both titled "Reunião em 'General'".

Microsoft Teams (Left):

- General Channel:** LUIZ PINHEIRO JUNIOR posted at 12:28: "Olá a todos(as), boa tarde, Desenvolvimento de Sistema do ETIM, neste componente de TPA estaremos tratando conceitos de Lógica de Programação em Java NetBeans. A partir das 13:00 estará disponível o formulário da lista de presença, segue o formulário: https://forms.office.com/Pages/ResponsePage.aspx?id=b6f47UG2rUO6dgBuCnGKj9wYBtQCoqkSEtHuVURD2V1g1TrQSVUY2RUStQ2o5CV8VNksyWSQIQCNjPTeu".
- Participants:** ITALO PEDRO LOPES SILVA, CICERO EDUARDO CAMPOS LEITE BISPO, BEATRIZ DAMAS, VANESSA MAYTA APAZA, JULIA CRISTINA GARCIA, RAPHAEL OLIVEIRA SILVA BATISTA, and SEBASTIAN ANGEL GAONA SILVA.
- Actions:** A "Reply" button is visible at the bottom left of the message area.

Microsoft Word Document (Right):

- Title:** logica_agenda_Decisao - Microsoft Word
- Content:** A code snippet in Java showing a try-catch block for handling NumberFormatException. The code is as follows:

```
try {
    comando(s);
} catch (NumberFormatException e) {
    Comando(s);
}
```
- Text Below Code:** "Para evitar ou minimizar aborrecimentos, no Java existem várias técnicas e rotinas de tratamento de erros. Vamos apresentar aqui o comando try-catch. Ele consiste em capturar erros na conversão de tipos na entrada de dados para os nossos programas. O comando trata mais tipos de erro, mas não é o escopo desta explicação. A sintaxe é:"
- Code Example:** A Java code example demonstrating the use of try-catch for integer input.

```
import javax.swing.JOptionPane;
public class tryCatch {
    public static void main(String[] args) {
        // declaração de variáveis
        int numero=0;
        String aux;
        // entrada de dados
        try {
            aux = JOptionPane.showInputDialog("Entre com um número inteiro");
            numero = Integer.parseInt(aux);
            JOptionPane.showMessageDialog(null, "O número inserido foi " + numero);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Entrada inválida");
        }
    }
}
```
- Page Information:** Página 19 de 22 | Palavras: 3.272 |
- Bottom Navigation:** Includes icons for +30, HB, and user profiles.

Reunião em 'TPA-1A-I-ETIM DE...' 14:21 40 participantes

logica_agenda_Décisão - Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição

```
JOptionPane.showMessageDialog(null, "Entre somente com um número Inteiro",  
"E R R O", JOptionPane.ERROR_MESSAGE);  
  
O comando JOptionPane.showMessageDialog(null, "Entre com um número Inteiro" exibe a caixa de diálogo que já  
conhecemos. Note que o número de argumentos do comando aumentou:  
,"E R R O", JOptionPane.ERROR_MESSAGE);  
  
Isso que inclui a formação da caixa de diálogo. A parte em vermelho ("E R R O"), insere o título da janela e a parte  
em verde (JOptionPane.ERROR_MESSAGE) altera o ícone para uma cruz vermelha para indicar erro.
```

Uma observação: os argumentos são sempre separados por vírgula

Exemplo:

```
import javax.swing.JOptionPane;  
  
public class mensagens_de_alerta {  
    public static void main(String[] args) {  
        JOptionPane.showMessageDialog(null, "Mensagem de Alerta",  
        "Alerta", JOptionPane.WARNING_MESSAGE);  
    }  
}
```

Página: 21 de 22 | Palavras: 3.272 | 90%

LUIZ J

JULIA G LOUISY T Hertz B LUIZ J

VIDEO CALL MUTE UNMUTE SPEAKER DIALER

Wi-Fi 4G ☰ 11:44 40 participantes 13:56

Reunião em 'TPA-1A-I-ETIM DE...' 11:44 40 participantes

Exemplo:

Se quisermos exibir uma mensagem de alerta para o usuário com um ponto de exclamação o código será:

```
import javax.swing.JOptionPane;
public class mensagem_de_alerta {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Mensagem de Alerta",
        "Alerta", JOptionPane.WARNING_MESSAGE);
    }
}
```

Perceba que o título da janela foi definido como Alerta.

Alerta

Mensagem de Alerta

OK

Imagem 10: Caixa de mensagem de alerta e um botão "OK" centralizado na caixa.

Palavras: 3.272

ESTRUTURAS DE DECISÃO I

LUIZ J

JULIA G LOUISY T HERTZ B LUIZ J

VIDEO CALL MUTE SPEAKER MORE CALL

13:53

← Reunião em 'TPA-1A-I-ETIM DE...' 08:38 40 participantes

logica_agenda_Decisão - Microsoft Word

Inserir Layout da Página Referências Correspondências Revisão Exibição

Quando um erro de inserção de dados ocorre e, por exemplo, um caractere ou número real é inserido a seguinte mensagem de erro aparece:

ESTRUTURAS DE DECISÃO I

Entrada

Entre com um número inteiro
a

OK Cancelar

Imagen 08: Caixa de entrada de dados com a mensagem "Entre com um número inteiro", logo abaixo uma caixa de entrada de dados com o a letra a em minúsculo. À esquerda tem o botão "OK" e à direita o botão "Cancelar".

ERRO

Entre somente com um número Inteiro

OK

Imagen 09: Caixa de mensagem de erro com a mensagem "Entre somente com um número Inteiro" e o botão "OK" centralizado na caixa.

Note que o try-catch capturou o erro de conversão, afinal o caractere "a" não é um número inteiro e exibiu a mensagem de erro correspondente.

Um detalhe interessante: não sei se repararam, mas a caixa de diálogo da **Erro! Fonte de referência não encontrada** é bem diferente da **Erro! Fonte de referência não encontrada**. Isso porque durante a escrita da caixa de diálogo de saída, esta foi formatada. Vamos analisar o comando da linha 16 e 17:

```
JOptionPane.showMessageDialog(null, "Entre somente com um número Inteiro",  
"ERRO", JOptionPane.ERROR_MESSAGE);
```

LUIZ J

LOUISY T Hertz B LUIZ J

The screenshot shows a video conference interface with a Microsoft Word document displayed in the background. The Word document has a blue header bar and contains a small alert dialog box with the text "Mensagem de Alerta" and an "OK" button. A red callout bubble points from the text "VOCÊ NO COMANDO" to this dialog box. The interface includes participant thumbnails for Julia G, Louisy T, Hertz B, and LUIZ J, along with standard video, audio, and control buttons.

← Reunião em 'TPA-1A-I-ETIM DE...' 13:57

12:45 40 participantes

logica_agenda_Decisão Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição

Perceba que o título da janela foi definido como Alerta.

Alerta Mensagem de Alerta OK

Imagem 10: Caixa de mensagem de alerta e um botão "OK" centralizado na caixa.

VOCÊ NO COMANDO

Acabamos de ver como podemos modificar a exibição de uma caixa de diálogo de saída de dados. Explore um pouco na IDE Eclipse algumas outras personalizações desta caixa de diálogo. Utilize a função de auto completar comandos da IDE. Para usar esta função basta digitar o início de um comando e pressionar as teclas Control (CTRL) + barra de espaço.

Página: 22 de 22 | Palavras: 3.272 | 90%

JULIA G LOUISY T HERTZ B LUIZ J

VIDEO MIC SILENCIO VOLUME ... CALL

13:49

← Reunião em 'TPA-1A-I-ETIM DE...' 04:03 40 participantes

logica_agenda_Décisão - Microsoft Word

programa de computador ou app de celular já fechou inesperadamente sem nenhum aviso de erro e perdemos o que já estava sendo feito. Isso dá uma raiva imensa.

Para evitar ou minimizar aborrecimentos, no Java existem várias técnicas e rotinas de tratamento de erros. Vamos apresentar aqui o comando try-catch. Ele consiste em capturar erros na conversão de tipos na entrada de dados para os nossos programas. O comando trata mais tipos de erro, mas não é o escopo desta explicação. A sintaxe é:

```
try{  
    comando(s);  
} catch (NumberFormatException e){  
    Comando(s);  
}
```

Vamos explicar com um exemplo:

```
import javax.swing.JOptionPane;  
  
public class tryCatch {  
  
    public static void main(String[] args) {  
        // declaração de variáveis  
        int numero=0;  
        String aux;  
  
        //entrada de dado  
        try {  
            aux = JOptionPane.showInputDialog("Entre com um número inteiro");  
            numero = Integer.parseInt(aux);  
            JOptionPane.showMessageDialog(null, "O número inserido foi " +  
numero);  
        } catch (Exception e) {  
            JOptionPane.showMessageDialog(null, "Por favor entre com um número inteiro");  
        }  
    }  
}
```

Página: 19 de 22 | Palavras: 3.272 | 90%

ESTRUTURAS DE DECISÃO /

LUIZ J

JULIA G LOUISY T Hertz B LUIZ J

...

...

Reunião em 'TPA-1A-I-ETIM DE...' 13:54
09:29 40 participantes

Inserir Layout da Página Referências Correspondências Revisão Exibição

logica_agenda_Decisão - Microsoft Word

ESTRUTURAS DE DECISÃO I

Entrada

Entre com um número inteiro

a

OK Cancelar

Imagem 08: Caixa de entrada de dados com a mensagem "Entre com um número inteiro", logo abaixo uma caixa de entrada de dados com o a letra a em minúsculo. À esquerda tem o botão "OK" e à direita o botão "Cancelar".

Note que o try-catch capturou o erro de conversão, afinal o caractere "a" não é um número inteiro e exibiu a mensagem de erro correspondente.

Um detalhe interessante: não sei se repararam, mas a caixa de diálogo da Erro! Fonte de referência não encontrada. é bem diferente da Erro! Fonte de referência não encontrada. Isso porque durante a escrita da caixa de diálogo de saída, esta foi formatada. Vamos analisar o comando da linha 16 e 17:

```
JOptionPane.showMessagedialog(null, "Entre com um número Inteiro",  
"ERRO", JOptionPane.ERROR_MESSAGE);
```

O comando JOptionPane.showMessageDialog(null, "Entre com um número Inteiro" exibe a caixa de diálogo que já conhecemos. Note que o número de argumentos do comando aumentou:
,"ERRO", JOptionPane.ERROR_MESSAGE);

É isso que inclui a formação da caixa de diálogo. A parte em vermelho ("ERRO"), insere o título da janela e a parte em verde (JOptionPane.ERROR_MESSAGE) altera o ícone para uma cruz vermelha para indicar erro.

LUIZ J

JULIA G LOUISY T Hertz B LUIZ J

VIDEO CALL MUTE SPEAKER MORE CALL

13:55

← Reunião em 'TPA-1A-I-ETIM DE...' 10:23 40 participantes

logica_agenda_Decisao - Microsoft Word

Inserir Layout da Página Referências Correspondências Revisão Exibição

Quando um erro de inserção de dados ocorre e, por exemplo, um caractere ou número real é inserido a seguinte mensagem de erro aparece:

Entrada

Entre com um número inteiro
a

OK Cancelar

ERRO

Entre somente com um número Inteiro

OK

Imagem 08: Caixa de entrada de dados com a mensagem "Entre com um número inteiro", logo abaixo uma caixa de diálogo de erro com a letra a em minúsculo. À esquerda tem o botão "OK" e à direita o botão "Cancelar".

Imagem 09: Caixa de mensagem de erro com a mensagem "Entre somente com um número Inteiro" e um botão "OK" centralizado na caixa.

Note que o try-catch capturou o erro de conversão, afinal o caractere "a" não é um número inteiro e exibiu a mensagem de erro correspondente.

Um detalhe interessante: não sei se repararam, mas a caixa de diálogo da Erro! Fonte de referência não encontrada. é bem diferente da Erro! Fonte de referência não encontrada. Isso porque durante a escrita da caixa de diálogo de saída, esta foi formatada. Vamos analisar o comando da linha 16 e 17:

```
JOptionPane.showMessageDialog(null, "Entre somente com um número Inteiro",  
"ERRO", JOptionPane.ERROR_MESSAGE);
```

LUIZ J

JULIA G LOUISY T HERTZ B LUIZ J

VIDEOMÍDIA MÍDIA SILENCIO SOM ... CHAMADA

◀ ○ □

13:49

← Reunião em 'TPA-1A-I-ETIM DE...' 04:38 40 participantes

logica_agenda_Decisao Microsoft Word

Vamos explicar com um exemplo:

```
import javax.swing.JOptionPane;
public class tryCatch {
    public static void main(String[] args) {
        // declaração de variáveis
        int numero=0;
        String aux;
        //entrada de dados
        try {
            aux = JOptionPane.showInputDialog("Entre com um número inteiro");
            numero = Integer.parseInt(aux);
            JOptionPane.showMessageDialog(null, "O número inserido foi " + numero);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Entre somente com um número inteiro",
                "ERRO", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

O try-catch funciona da seguinte forma:

Na linha 11, o comando `try` tenta executar os comandos das linhas 12 a 14 que estão dentro das chaves, que é fechada na linha 15. Caso não consiga, ele executa a cláusula `catch (NumberFormatException e)` e os comandos dentro das chaves das linhas 15 a 18. Isto exibirá uma mensagem de erro.

Caso o usuário realmente digite um número como o solicitado pelo programa, a seguinte mensagem será exibida: "O número inserido foi <numero>" como nas figuras a seguir:

Página: 20 de 22 | Palavras: 3.272 | 90%

LOUISY T Hertz B LUIZ J

Wi-Fi 4G ☰ 10:00 40 participantes 13:55

Reunião em 'TPA-1A-I-ETIM DE...' 10:00 40 participantes

Vamos explicar com um exemplo:

```
import javax.swing.JOptionPane;
public class tryCatch {
    public static void main(String[] args) {
        // declaração de variáveis
        int numero=0;
        String aux;
        //entrada de dados
        try {
            aux = JOptionPane.showInputDialog("Entre com um número inteiro");
            numero = Integer.parseInt(aux);
            JOptionPane.showMessageDialog(null, "O número inserido foi " + numero);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Entre somente com um número Inteiro",
                    "ERRO", JOptionPane.ERROR_MESSAGE);
        }
    }
}
```

O try-catch funciona da seguinte forma:

Na linha 11, o comando `try` tenta executar os comandos das linhas 12 a 14 que estão dentro das chaves, que é fechada na linha 15. Caso não consiga, ele executa a cláusula `catch (NumberFormatException e)` e os comandos dentro das chaves das linhas 15 a 18. Isto exibirá uma mensagem de erro.

Caso o usuário realmente digite um número como o solicitado pelo programa, a seguinte mensagem será exibida: "O número inserido foi <numero>" como nas figuras a seguir:

The screenshot shows a video conference interface. At the top, there's a toolbar with icons for file, edit, view, insert, layout, references, correspondences, review, and display. Below the toolbar, a status bar shows the date and time (10:00 40 participantes 13:55). The main area displays a Microsoft Word document titled 'logica_agenda_Decisao'. The document contains Java code for a try-catch block. Below the document, a message reads: 'O try-catch funciona da seguinte forma:' followed by an explanation. Further down, two windows are shown: 'Entrada' (Input) with the text 'Entre com um número inteiro.' and 'Mensagem' (Message) with the text 'O número inserido foi <numero>'. The participant list at the bottom includes 'JULIA G', 'LOUISY T', 'HERTZ B', and 'LUIZ J'. A red button labeled 'Ligar' (Call) is visible at the bottom right. The interface has a dark theme with white text and light-colored buttons.

Wi-Fi 4G ☰ 10:26 40 participantes 13:55

Reunião em 'TPA-1A-I-ETIM DE...' 10:26 40 participantes

logica_agenda_Decisao - Microsoft Word

Inserir Layout da Página Referências Correspondências Revisão Exibição

conhecemos. Note que o número de argumentos do comando aumentou:
, "E R R O", JOptionPane.ERROR_MESSAGE);

É isso que inclui a formação da caixa de diálogo. A parte em vermelho ("E R R O"), insere o título da janela e a parte em verde (JOptionPane.ERROR_MESSAGE) altera o ícone para uma cruz vermelha para indicar erro.

 Uma observação: os argumentos são sempre separados por vírgula

Exemplo:

Se quisermos exibir uma mensagem de alerta para o usuário com um ponto de exclamação o código será:

```
import javax.swing.JOptionPane;
public class mensagem_de_alerta {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Mensagem de Alerta",
            "Alerta", JOptionPane.WARNING_MESSAGE);
    }
}
```

ESTRUTURAS DE DECISÃO

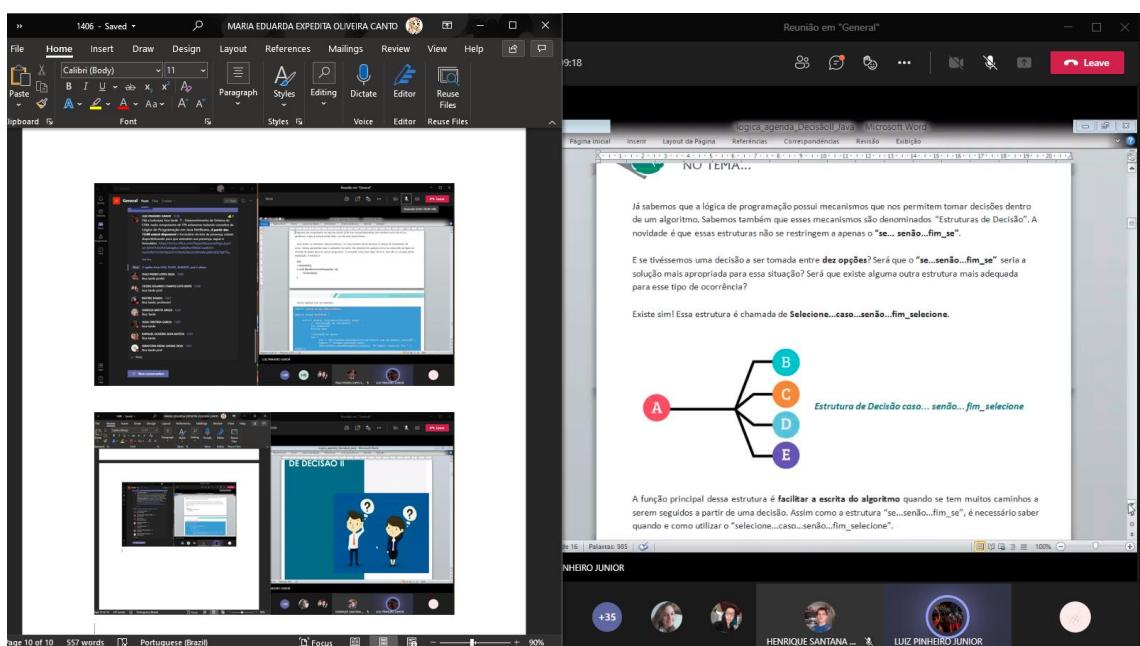
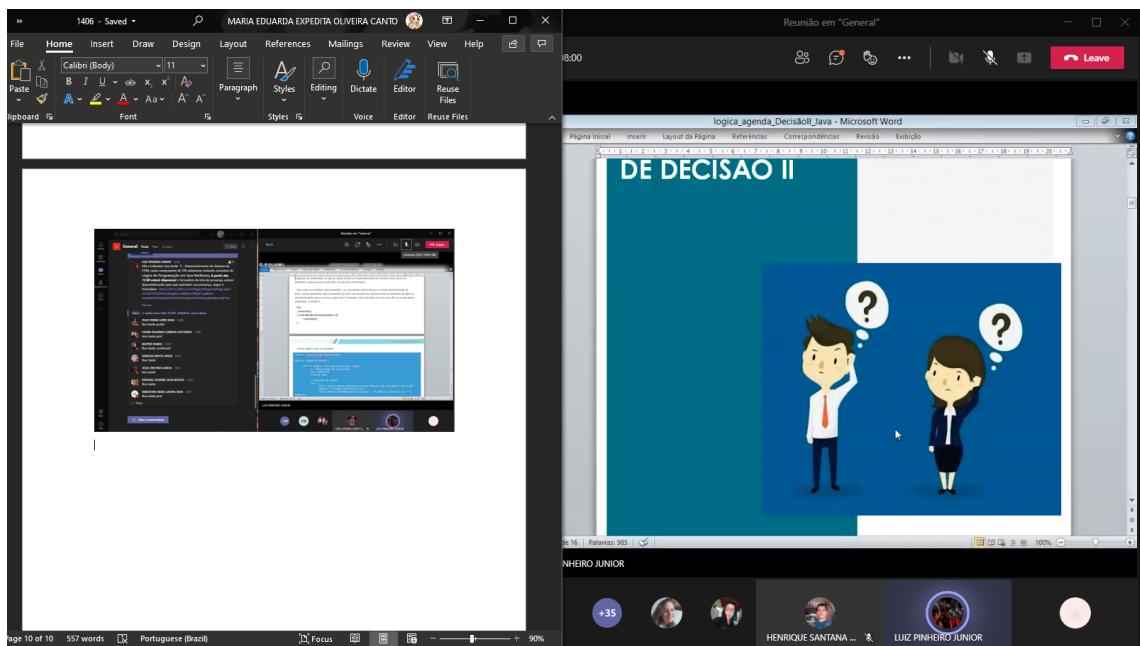
Perceba que o título da janela foi definido como Alerta.

LUIZ J

JULIA G LOUISY T HERTZ B LUIZ J

90%

Video camera icon Microphone icon Speaker icon Ellipsis icon Call icon



YouTube | WhatsApp

web.whatapp.com

Reunião em "General"

01:11:10

logica_agenda_DecisaoII.java - Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição

E se tivéssemos uma decisão a ser tomada entre **dez opções**? Será que o "se...senão...fim_se" seria a solução mais apropriada para essa situação? Será que existe alguma outra estrutura mais adequada para esse tipo de ocorrência?

Existe sim! Essa estrutura é chamada de **Selezione...caso...senão...fim_selezione**.

Estrutura de Decisão caso... senão... fim_selezione

A função principal dessa estrutura é **facilitar a escrita do algoritmo** quando se tem muitos caminhos a serem seguidos a partir de uma decisão. Assim como a estrutura "se...senão...fim_se", é necessário saber quando e como utilizar o "selezione...caso...senão...fim_selezione".

A estrutura "selezione...caso...senão...fim_selezione" do pseudocódigo corresponde à estrutura **switch-case** do Java. Elas permitem que a seleção correta seja feita a partir da comparação entre o valor do conteúdo da variável e uma lista definida durante a programação. Ao encontrar a correspondência correta, os respectivos comandos são executados e as demais opções ignoradas.

Página 2 de 16 | Palavras: 365

LUIZ PINHEIRO JUNIOR +35 HENRIQUE SANTANA ... LUIZ PINHEIRO JUNIOR

1406 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Reunião em "General"

1:58

logica_agenda_DecisaoII.java - Microsoft Word

Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição Design Layout

A função principal dessa estrutura é **facilitar a escrita do algoritmo** quando se tem muitos caminhos a serem seguidos a partir de uma decisão. Assim como a estrutura "se...senão...fim_se", é necessário saber quando e como utilizar o "selezione...caso...senão...fim_selezione".

A estrutura "selezione...caso...senão...fim_selezione" do pseudocódigo corresponde à estrutura **switch-case** do Java. Elas permitem que a seleção correta seja feita a partir da comparação entre o valor do conteúdo da variável e uma lista definida durante a programação. Ao encontrar a correspondência correta, os respectivos comandos são executados e as demais opções ignoradas.

ESTRUTURAS DE DECISÃO II

PSEUDOCÓDIGO	FLUXOGRAAMA	JAVA
Selezione Caso (Variável)	Caso condição 01	switch (variável) { case condição 01:

de 16 | Palavras: 365

LUIZ PINHEIRO JUNIOR +35 HENRIQUE SANTANA ... LUIZ PINHEIRO JUNIOR

1406 - Saved - MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição Design Layout

ESTRUTURAS DE DECISÃO II

PSEUDOCÓDIGO	FLUXOGRAMA	JAVA
<pre>Selecione Caso (variável) Caso condição 01 (comando(s)) Caso condição 02 (comando(s)) Caso condição 03 (comando(s)) ... Senão (comando(s)) Fim-Selezione</pre>		<pre>switch (variável){ case condição 01: (comando(s)); case condição 02: (comando(s)); case condição 03: (comando(s)); ... default: (comando(s)); }</pre>

Se recapitarmos o último exemplo de Estrutura de Decisão Aninhada do tópico anterior, no qual o programa exiba o dia da semana de acordo com o número digitado pelo usuário, notamos que o código do programa fica relativamente **confuso** devido à grande quantidade de comandos de seleção (se). A tendência dessa confusão é aumentar conforme o número de comandos de decisão aninhados for crescendo. Isto, é claro, levando em conta que estamos realizando a comparação sempre com a mesma variável.

Para essas situações podemos utilizar a estrutura "selecione caso...senão...fim_selezione" do pseudocódigo ou a estrutura switch-case do Java.

Page 12 of 12 | 557 words | Portuguese (Brazil) | Focus | Print | + 90%

HENRIQUE JUNIOR +35 HENRIQUE SANTANA ... LUIZ PINHEIRO JUNIOR

YouTube WhatsApp

+55 11 98373-7831 hoje às 14:08

Reunião em "General"

01:15:10

Página 4 de 16 | Palavras: 365

ESTRUTURAS DE DECISÃO II

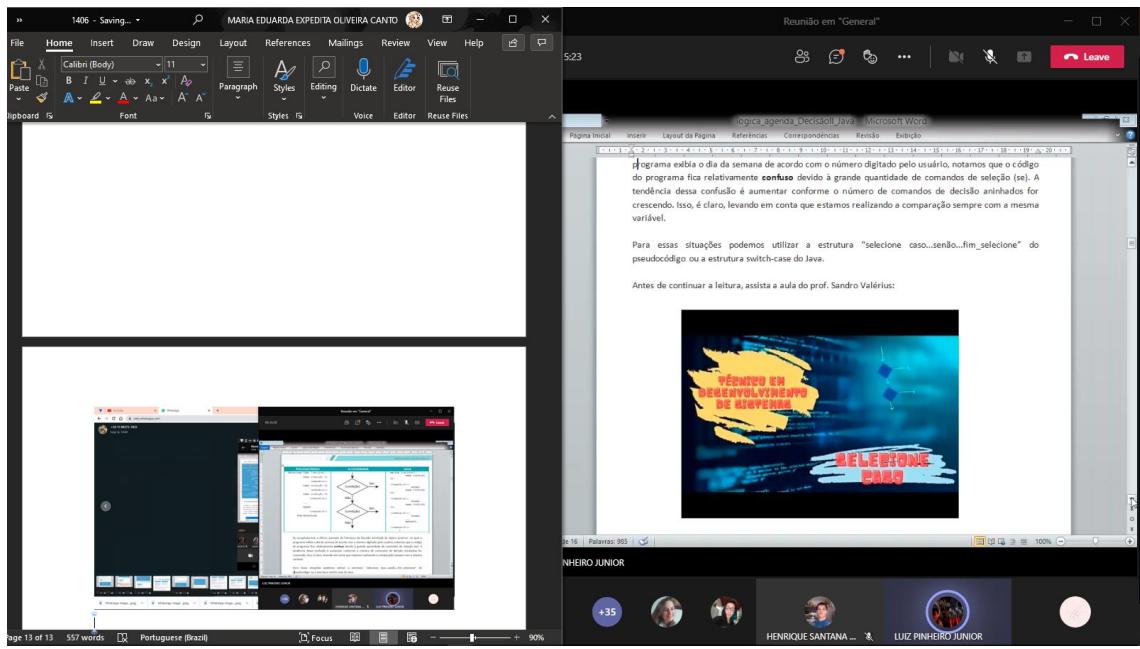
PSEUDOCÓDIGO	FLUXOGRAMA	JAVA
<pre>Selecione Caso (variável) Caso condição 01 (comando(s)) Caso condição 02 (comando(s)) Caso condição 03 (comando(s)) ... Senão (comando(s)) Fim-Selezione</pre>		<pre>switch (variável){ case condição 01: (comando(s)); case condição 02: (comando(s)); case condição 03: (comando(s)); ... default: (comando(s)); }</pre>

Se recapitarmos o último exemplo de Estrutura de Decisão Aninhada do tópico anterior, no qual o programa exiba o dia da semana de acordo com o número digitado pelo usuário, notamos que o código do programa fica relativamente **confuso** devido à grande quantidade de comandos de seleção (se). A tendência dessa confusão é aumentar conforme o número de comandos de decisão aninhados for crescendo. Isto, é claro, levando em conta que estamos realizando a comparação sempre com a mesma variável.

Para essas situações podemos utilizar a estrutura "selecione caso...senão...fim_selezione" do pseudocódigo ou a estrutura switch-case do Java.

WhatsApp Image...jpeg WhatsApp Image...jpeg WhatsApp Image...jpeg

LUIZ PINHEIRO JUNIOR +35 HENRIQUE SANTANA ... LUIZ PINHEIRO JUNIOR



1406 - Saved - MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição

FLUXOGRAMA

```

    graph TD
        Inicio([Início]) --> dia{dia como inteiro}
        dia --> Sim1{1}
        Sim1 --> V1["Vôd escuchó Domingo"]
        V1 --> Sim2{2}
        Sim2 --> V2["Vôd escuchó Segundo"]
        V2 --> Sim3{3}
        Sim3 --> V3["Vôd escuchó Tercer"]
        V3 --> Sim4{4}
        Sim4 --> V4["Vôd escuchó Cuarto"]
        V4 --> Sim5{5}
        Sim5 --> V5["Vôd escuchó Quinto"]
        V5 --> Sim6{6}
        Sim6 --> V6["Vôd escuchó Sexta"]
        V6 --> Sim7{7}
        Sim7 --> V7["Vôd escuchó Sábado"]
        V7 --> Sim8{8}
        Sim8 --> V8["Número Inválido"]
        V8 --> Fim([Fim])
    
```

Tanto as intenções em pseudocódigo quanto em Java são semelhantes como temos a seguir:

PSUEDOCÓDIGO

```

    Start()
    Imprime_mensagem()
    Imprime_dia_semana()
    Inicio()
    Dado_1()
    Dado_2()
    Dado_3()
    Dado_4()
    Dado_5()
    Dado_6()
    Dado_7()
    Dado_8()
    Entrada("Digite um número de 1 a 8")
    Se(dia == 1) {
        Imprime("Vôd escuchó Domingo")
    }
    Se(dia == 2) {
        Imprime("Vôd escuchó Segundo")
    }
    Se(dia == 3) {
        Imprime("Vôd escuchó Tercer")
    }
    Se(dia == 4) {
        Imprime("Vôd escuchó Cuarto")
    }
    Se(dia == 5) {
        Imprime("Vôd escuchó Quinto")
    }
    Se(dia == 6) {
        Imprime("Vôd escuchó Sexta")
    }
    Se(dia == 7) {
        Imprime("Vôd escuchó Sábado")
    }
    Se(dia < 1 || dia > 7) {
        Imprime("Número Inválido")
    }
    Fim()

```

FLUXOGRAMA

```

    graph TD
        Inicio([Início]) --> dia{dia como inteiro}
        dia --> Sim1{1}
        Sim1 --> V1["Vôd escuchó Domingo"]
        V1 --> Sim2{2}
        Sim2 --> V2["Vôd escuchó Segundo"]
        V2 --> Sim3{3}
        Sim3 --> V3["Vôd escuchó Tercer"]
        V3 --> Sim4{4}
        Sim4 --> V4["Vôd escuchó Cuarto"]
        V4 --> Sim5{5}
        Sim5 --> V5["Vôd escuchó Quinto"]
        V5 --> Sim6{6}
        Sim6 --> V6["Vôd escuchó Sexta"]
        V6 --> Sim7{7}
        Sim7 --> V7["Vôd escuchó Sábado"]
        V7 --> Sim8{8}
        Sim8 --> V8["Número Inválido"]
        V8 --> Fim([Fim])
    
```

ESTRUTURAS DE DECISÃO

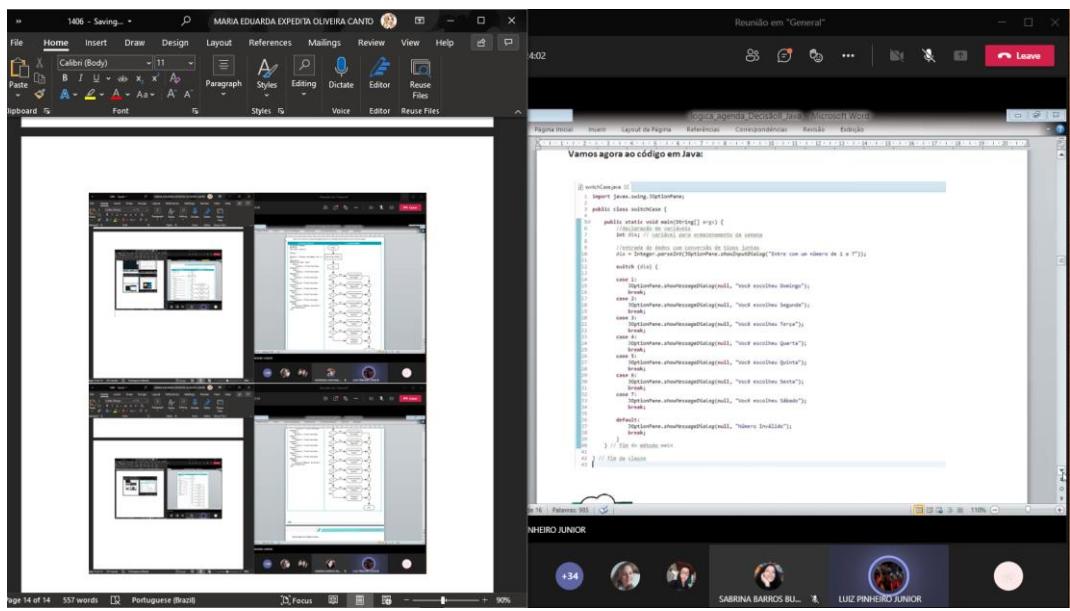
Vamos agora ao código em Java:

NHEIRO JUNIOR

+34 HENRIQUE SANTANA LUIZ PINHEIRO JUNIOR

NHEIRO JUNIOR

+34 SABRINA BARROS BU... LUIZ PINHEIRO JUNIOR



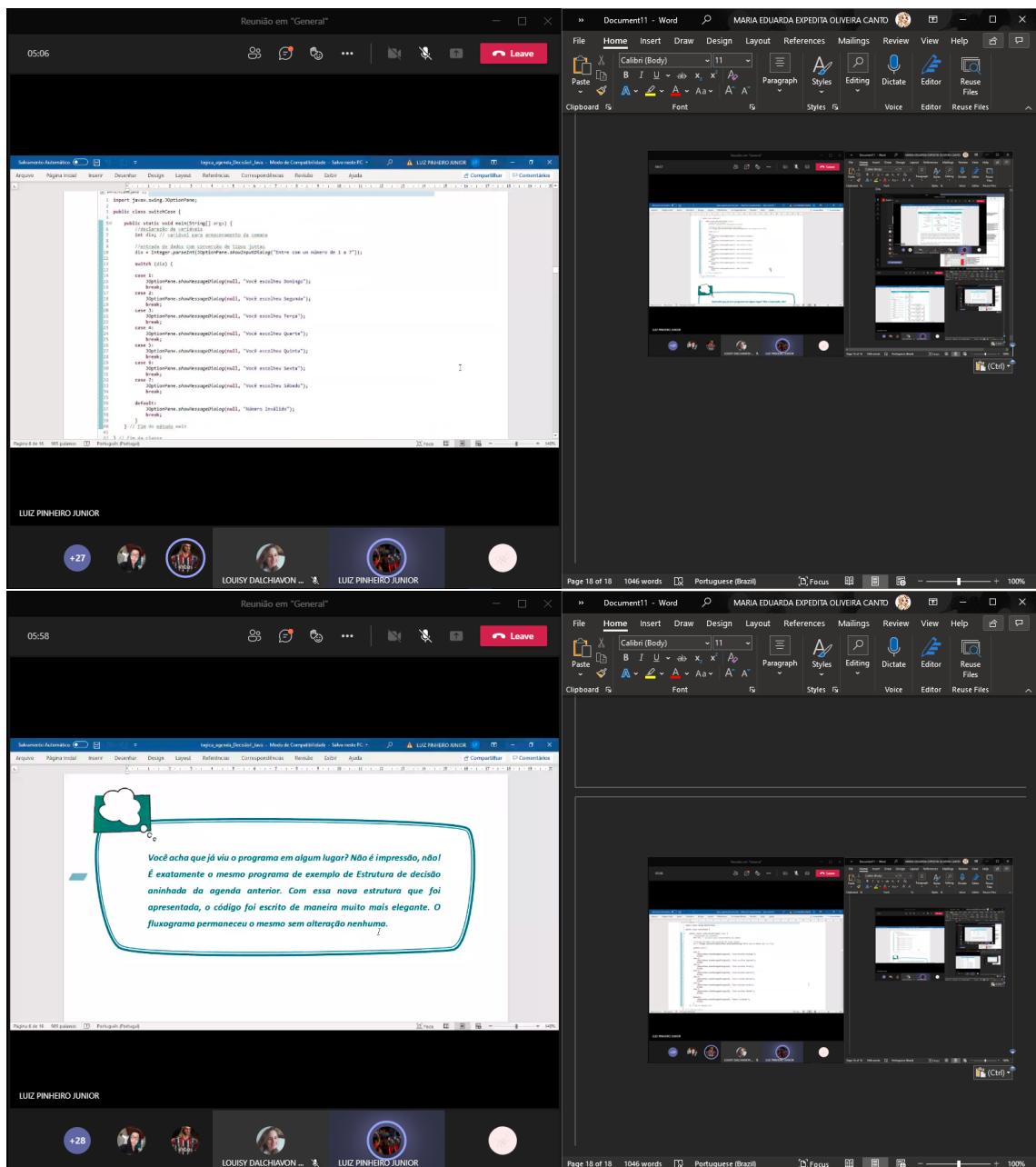
2106

The pseudocode is as follows:

```

import java.util.Scanner;
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int dia = sc.nextInt();
    //Entrada de dados que informa da dia da semana
    //Entrada de dados que informa de dia de semana
    //Entrada de dados que informa de dia de semana
    //Entrada de dados que informa de dia de semana
    switch (dia) {
        case 1:
            System.out.println("Você esconde Domingo");
            break;
        case 2:
            System.out.println("Você esconde Segunda");
            break;
        case 3:
            System.out.println("Você esconde Terça");
            break;
        case 4:
            System.out.println("Você esconde Quarta");
            break;
        case 5:
            System.out.println("Você esconde Quinta");
            break;
        case 6:
            System.out.println("Você esconde Sexta");
            break;
        default:
            System.out.println("Número Inválido");
    }
}
// Fim do algoritmo main.

```



Reunião em "General" 06:23

Document11 - Word MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Clipboard

Font Styles Voice Editor Reuse Files

Home Insert Draw Design Layout References Mailings Review View Help

File

Exemplo:

Se o usuário entrar com o valor 3, teremos dia = 3 e ao executarmos o comando case 3: (dia=3), como o resultado da comparação será verdadeiro, ele executará o comando JOptionPane.

```
showMessageDialog(null, "Você escolheu Terça");
```

e o comando break; e por fim finalizará o programa.

O comando break tem a finalidade de parar a execução do comando switch, uma vez que já foi executada a ação necessária (comparação verdadeira) e não há a necessidade de se continuar com o comando switch.

Caso nenhuma comparação resulte em uma resposta verdadeira, o comando default: é executado gerando uma mensagem de "número inválido" para o usuário.

VOCÊ NO

LOUISY DALCHIACON ... LUIZ PINHEIRO JUNIOR

Reunião em "General" 07:10

Document11 - Word MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Clipboard

Font Styles Voice Editor Reuse Files

Home Insert Draw Design Layout References Mailings Review View Help

File

Os exercícios de 1 a 4 devem ser desenvolvidos elaborando o pseudocódigo, o fluxograma e a linguagem Java:

- Vanessa precisa desenvolver um programa em que ao se digitar o código de um produto cadastrado da padaria Lápis Colored este retorno o nome do produto. Se o código do produto não estiver cadastrado, o programa deve exibir a mensagem produto não cadastrado. A tabela a seguir descreve os códigos e os produtos cadastrados:
Como Vanessa resolveria essa questão?
- Elberto deseja fazer um menu de seleção de um programa utilizando o comando de seleção múltipla. O menu consiste nas opções de:
 - Cadastrar usuário
 - Alterar dados
 - Excluir UsuárioComo Elberto faria esse programa? Elabore uma mensagem diferente para ser exibida para o usuário ao selecionar cada um dos itens dos menus.

LUIZ PINHEIRO JUNIOR

Links LOUISY DALCHIACON ... LUIZ PINHEIRO JUNIOR

Reunião em "General" 09:43

Document11 - Word MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Serviços Automáticos | logica_agente_0x001_Java - Motor de Compatibilidade - Salvo neste PC | LUIZ PINHEIRO JUNIOR

Diagrama de Fluxo:

```

    graph TD
        Start((Início)) --> Input[Leia código do produto]
        Input --> Cond1{Caso condição 100}
        Cond1 --> Print1[Imprime "Linha com o código do produto"]
        Cond1 --> End1((Fim))
        Cond1 --> Cond2{Caso condição 150}
        Cond2 --> Print2[Imprime "Corrente branca"]
        Cond2 --> End2((Fim))
        Cond2 --> Cond3{Caso condição 200}
        Cond3 --> Print3[Imprime "Corrente azul"]
        Cond3 --> End3((Fim))
        Cond3 --> Cond4{Caso condição 250}
        Cond4 --> Print4[Imprime "Corrente amarela"]
        Cond4 --> End4((Fim))
        Cond4 --> Cond5{Caso condição 300}
        Cond5 --> Print5[Imprime "Corrente vermelha"]
        Cond5 --> End5((Fim))
        Cond5 --> Cond6{Caso condição 470}
        Cond6 --> Print6[Imprime "Valor constante"]
        Cond6 --> End6((Fim))
        Cond6 --> Cond7{Caso condição 500}
        Cond7 --> Print7[Imprime "Límite de tensão branco 100%"]
        Cond7 --> End7((Fim))
        Cond7 --> Cond8{Caso condição 550}
        Cond8 --> Print8[Imprime "Límite de tensão branca 50%"]
        Cond8 --> End8((Fim))
        Cond8 --> Cond9{Caso condição 999}
        Cond9 --> Print9[Imprime "Produto não cadastrado"]
        Cond9 --> End9((Fim))
        Print9 --> Fim[Finaliza]
    
```

Page 18 of 18 1046 words

Portuguese (Brazil) Focus

LUIZ PINHEIRO JUNIOR +32 LUIZ PINHEIRO JUNIOR RODRIGO COTRIN GU... 11:27

Reunião em "General"

Document11 - Word MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Serviços Automáticos | logica_agente_0x001_Java - Motor de Compatibilidade - Salvo neste PC | LUIZ PINHEIRO JUNIOR

Diagrama de Fluxo:

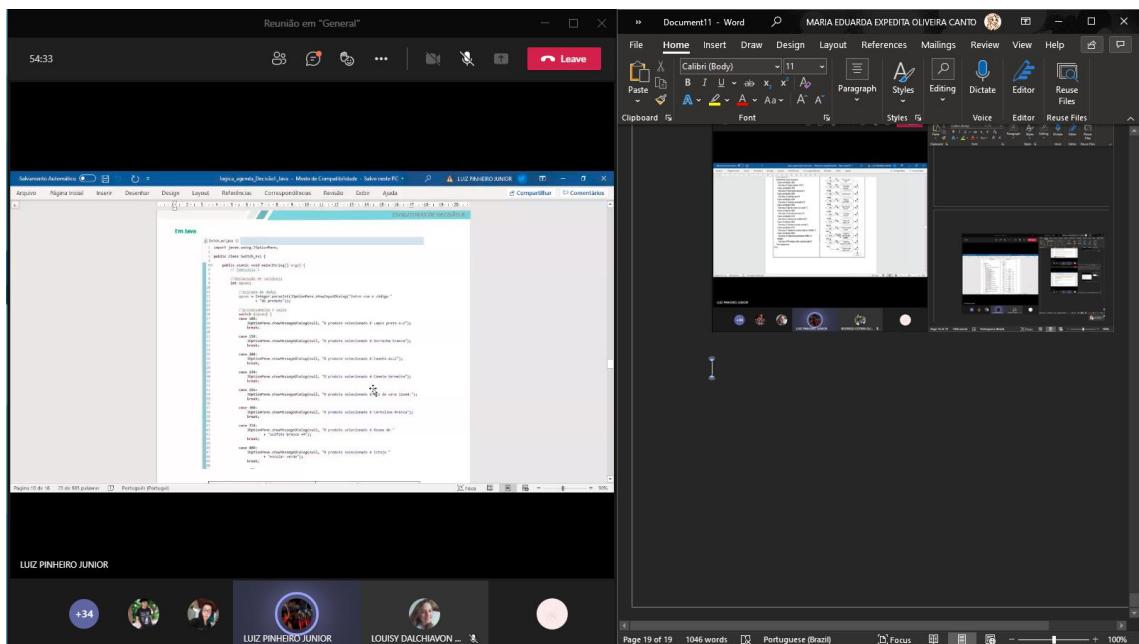
```

    graph TD
        Start((Início)) --> Input[Leia código do produto]
        Input --> Cond1{Caso condição 100}
        Cond1 --> Print1[Imprime "Linha com o código do produto"]
        Cond1 --> End1((Fim))
        Cond1 --> Cond2{Caso condição 150}
        Cond2 --> Print2[Imprime "Corrente branca"]
        Cond2 --> End2((Fim))
        Cond2 --> Cond3{Caso condição 200}
        Cond3 --> Print3[Imprime "Corrente azul"]
        Cond3 --> End3((Fim))
        Cond3 --> Cond4{Caso condição 250}
        Cond4 --> Print4[Imprime "Corrente amarela"]
        Cond4 --> End4((Fim))
        Cond4 --> Cond5{Caso condição 300}
        Cond5 --> Print5[Imprime "Corrente vermelha"]
        Cond5 --> End5((Fim))
        Cond5 --> Cond6{Caso condição 470}
        Cond6 --> Print6[Imprime "Valor constante"]
        Cond6 --> End6((Fim))
        Cond6 --> Cond7{Caso condição 500}
        Cond7 --> Print7[Imprime "Límite de tensão branco 100%"]
        Cond7 --> End7((Fim))
        Cond7 --> Cond8{Caso condição 550}
        Cond8 --> Print8[Imprime "Límite de tensão branca 50%"]
        Cond8 --> End8((Fim))
        Cond8 --> Cond9{Caso condição 999}
        Cond9 --> Print9[Imprime "Produto não cadastrado"]
        Cond9 --> End9((Fim))
        Print9 --> Fim[Finaliza]
    
```

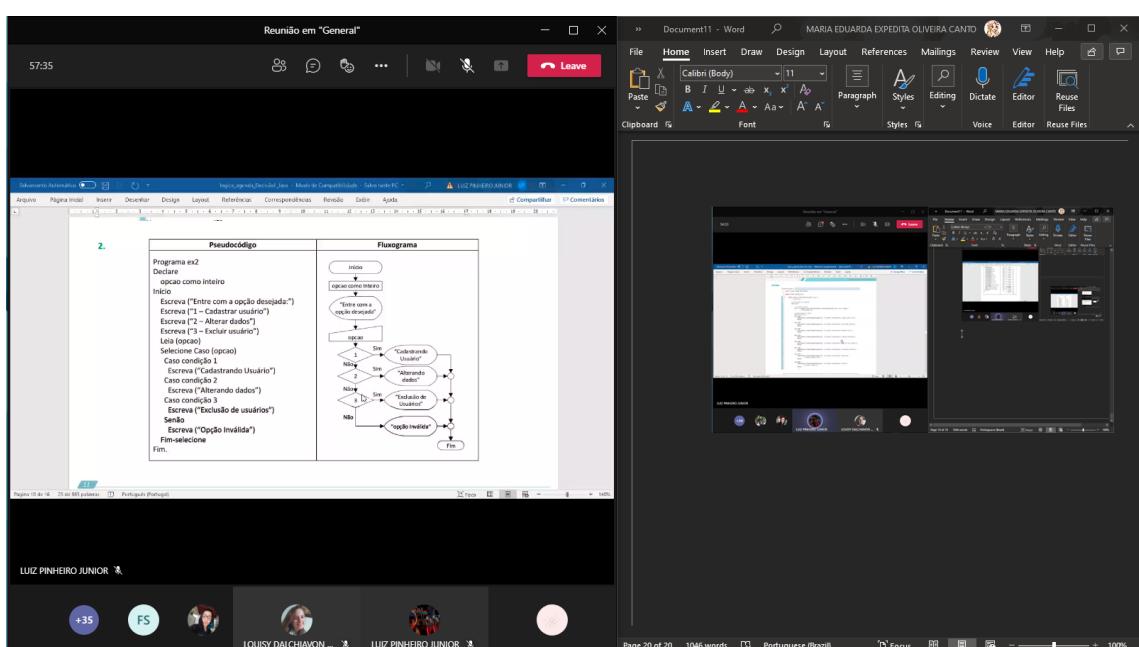
Page 19 of 19 1046 words

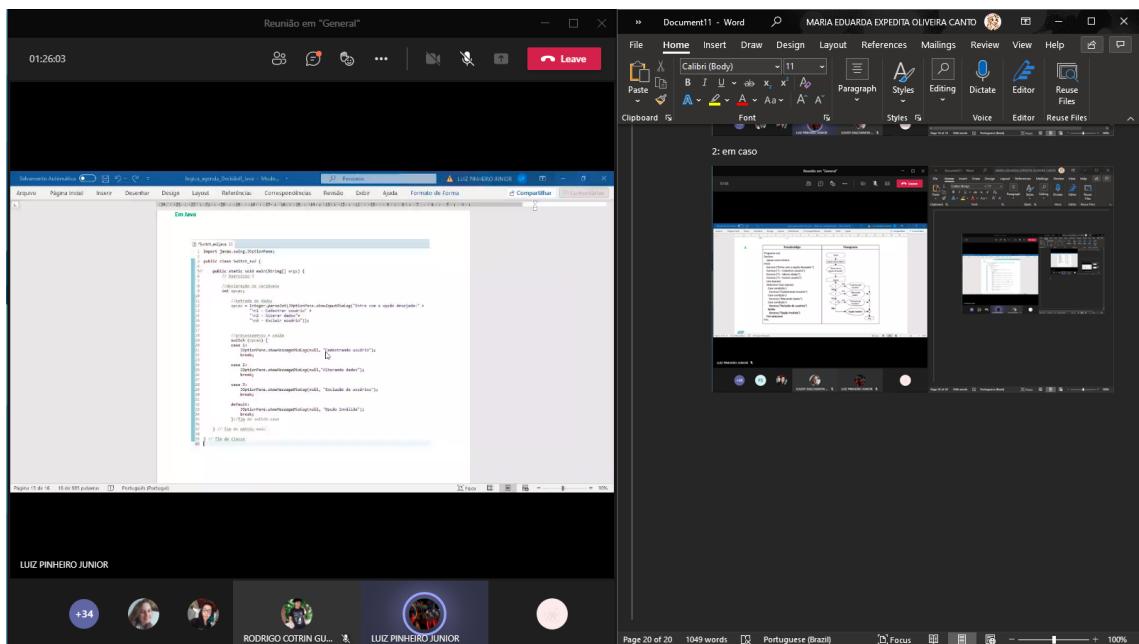
Portuguese (Brazil) Focus

LUIZ PINHEIRO JUNIOR +34 LUIZ PINHEIRO JUNIOR RODRIGO COTRIN GU...



2: em caso

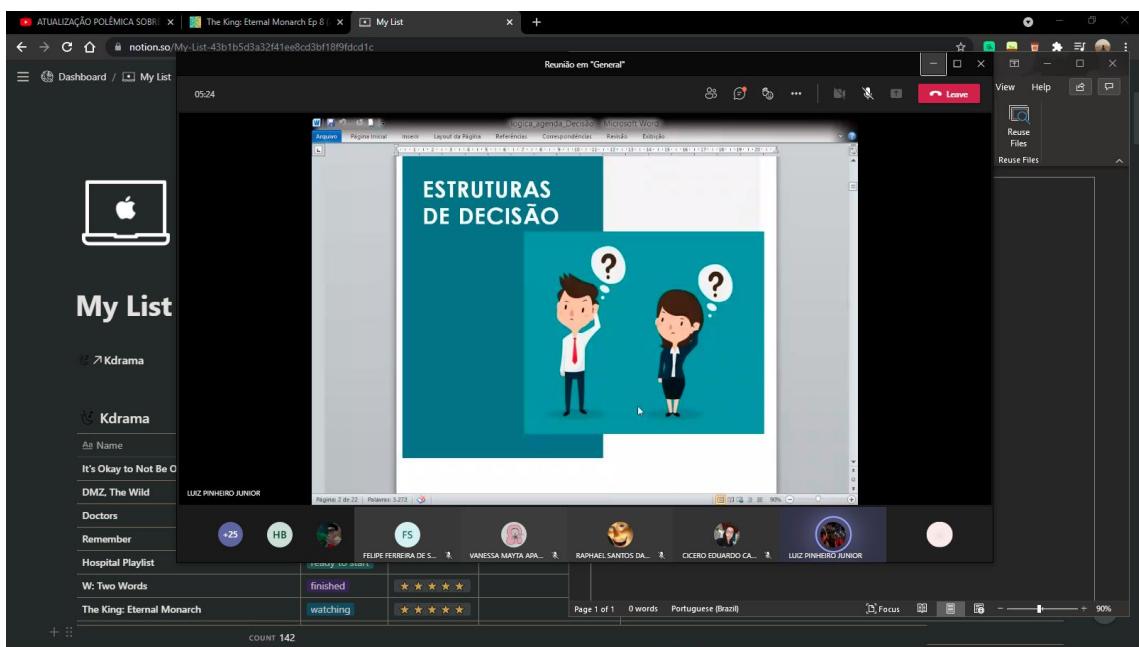




2806

0507

2607



ATUALIZAÇÃO POLÉMICA SOBRE | The King Eternal Monarch Ep 8 | My List

Reunião em "General"

0632

LOGO_agenda_Decisão - Microsoft Word

MERGULHANDO NO TEMA...

Estruturas de Decisão

As estruturas de decisão, comuns em programas que realizam tarefas simples como a visualização da saída de dados e pequenos cálculos matemáticos. Contudo, os algoritmos criados até aqui não possuem poder de decisão. Ou seja, eles sempre executam as mesmas tarefas independentemente dos resultados obtidos. Inicialmente, nem todos os testes possuem decisões que muitas vezes são difíceis e que podem alterar o rumo de nossas vidas. Com os programas ocorre a mesma coisa. Em programação, chamamos estas decisões de Estrutura de decisão. No Java, chamamos de comando IF.

Imagem 01: Frosip - Tomada de Decisão: homem indeciso sobre qual caminho seguir.

Page: 3 de 22 | Palavras: 3.272 |

LUZ PINHEIRO JUNIOR

RAPHAEL SANTOS DA... CICERO EDUARDO CAL... LUIZ PINHEIRO JUNIOR

Page 1 of 1 0 words Portuguese (Brazil) Focus

AutoSave Document References Mailings Review View Help

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Font Styles Voice Editor Reuse Files

Page 1 of 1 0 words Portuguese (Brazil) Focus

AutoSave Document References Mailings Review View Help

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Font Styles Voice Editor Reuse Files

ATUALIZAÇÃO POLÉMICA SOBRE | The King Eternal Monarch Ep 8 | My List

Reunião em "General"

08:00

LOGO_agenda_Decisão - Microsoft Word

MERGULHANDO NO TEMA...

Estruturas de decisão IF (Se...Então)

As estruturas de decisão ou testes condicionais nos permitem executar um conjunto diferente de comandos dependendo do resultado de um teste utilizando operadores relacionais. Este resultado pode ser verdadeiro ou falso conforme podemos visualizar na tabela a seguir:

A = condição, continuaria B ou desviaria para C.

Diagrama de fluxo: A (vermelha) é uma condição que leva a B (azul) ou C (verde).

Page: 3 de 22 | Palavras: 3.272 |

LUZ PINHEIRO JUNIOR

Links

Page 1 of 1 0 words Portuguese (Brazil) Focus

AutoSave Document References Mailings Review View Help

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Font Styles Voice Editor Reuse Files

Page 1 of 1 0 words Portuguese (Brazil) Focus

AutoSave Document References Mailings Review View Help

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Font Styles Voice Editor Reuse Files

A= condição, continuaria B ou desviaria para C.

Reunião em "General"

12:00

Documentos | Microsoft Word

Página Inicial | Insert | Layout da Página | Referências | Correspondências | Revisão | Edição

ESTRUTURAS DE DECISÃO

PSEUDOCÓDIGO	FLUXOGRAAMA	JAVA
SE [condição] Então [comando(s)] Fim-Se	<pre> graph TD Start(()) --> Cond{[condição]} Cond -- NÃO --> Join(()) Cond -- SIM --> Commands[Comando(s)] Commands --> Join Join --> End(()) </pre>	<pre> if [condição]{ [comando(s)] } </pre>

Exemplificando...

Page 4 of 22 | Palavras: 3,272 | 90%

LUIZ PINHEIRO JUNIOR

+3 | RAPHAEL SANTOS DA... | CICERO EDUARDO CA... | LUIZ PINHEIRO JUNIOR

Page 2 of 3 | 8 words | Portuguese (Brazil) | Focus | 90%

Reunião em "General"

16:34

Documentos | Microsoft Word

Página Inicial | Insert | Layout da Página | Referências | Correspondências | Revisão | Edição

ESTRUTURAS DE DECISÃO

PSEUDOCÓDIGO	FLUXOGRAAMA	JAVA
Se [condição] Então [comando(s) condição verdadeira] Sendo [Comando(s) condição falsa] Fim-Se	<pre> graph TD Start(()) --> Cond{[condição]} Cond -- NÃO --> CommandsFalse[Comando(s) Condição Falsa] CommandsFalse --> Join(()) Cond -- SIM --> CommandsTrue[Comando(s) Condição Verdadeira] CommandsTrue --> Join Join --> End(()) </pre>	<pre> if [condição] { [comando(s) condição verdadeira]; } else { [comando(s) condição falsa]; } </pre>

Imagem 04: GEExD – Representação de Pseudocódigo, Fluxograma e Codificação Java – Síntaxe do Comando "Se...Senão".

Observando a tabela acima, você pode notar que, caso o teste lógico condicional fale (ou seja, caso a condição não seja atendida), temos um comando ou grupo de comandos a serem executados; os comandos indicados após o Senão.

Page 6 of 22 | Palavras: 3,272 | 90%

LUIZ PINHEIRO JUNIOR

+3 | RAPHAEL SANTOS DA... | CICERO EDUARDO CA... | LUIZ PINHEIRO JUNIOR

Page 2 of 3 | 8 words | Portuguese (Brazil) | Focus | 90%

Reunião em "General" 17:28

Diagrama 10: GEdu - Representação de Pseudocódigo, Fluxograma e Codificação Java - Comando "Se" para verificar se a idade é maior que 18 anos.

PSEUDOCÓDIGO:

```
SE (idade >= 18) Então
    Escreva ("Maior de idade")
    Senão
        Escreva ("Menor de idade")
    Fim Se
```

FLUXOGRAMA:

```

graph TD
    Start(( )) --> Cond{ }
    Cond -- SIM --> Maior[Maior de idade]
    Maior --> Join(( ))
    Cond -- NÃO --> Menor[Menor de idade]
    Menor --> Join
    Join --> End(( ))

```

JAVA:

```

if (idade >= 18){
    JOptionPane.showMessageDialog(null, "Maior de idade");
} else {
    JOptionPane.showMessageDialog(null, "Menor de idade");
}

```

Percebemos que o pseudocódigo acima exemplo anterior, porém caso o teste condicional falso [resultado falso] não executamos um comando que exibe a mensagem "Menor de idade" para o usuário. Isso é feito por meio da utilização da cláusula Senão no Pseudocódigo, e é feito no Java.

Programa completo codificado em Java:

```

import javax.swing.JOptionPane;
public class IFCompleto {
}
```

Page 7 of 22 | Palavras: 3,272 | 90%

LUIZ PINHEIRO JUNIOR RAPHAEL SANTOS DA... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Reunião em "General" 11:16

Diagrama 11: GEdu - Representação de Pseudocódigo e Codificação Java - Comando "Se...Senão" para verificar se a idade é maior que 18 anos.

PSEUDOCÓDIGO:

```
SE (idade >= 18) Então
    Escreva ("Idade = 18")
    Senão
        Escreva ("Maior de 18")
    Fim Se
    Senão
        Escreva ("Menor de idade")
    Fim Se
```

FLUXOGRAMA:

```

graph TD
    Start(( )) --> Cond{ }
    Cond -- SIM --> Idade[Idade = 18]
    Idade --> Join1(( ))
    Cond -- NÃO --> Maior[Maior de 18]
    Maior --> Join1
    Join1 --> Senao{ }
    Senao --> Menor[Menor de idade]
    Menor --> Join2(( ))
    Join2 --> End(( ))

```

JAVA:

```

if (idade >= 18){
    System.out.println("Idade = 18");
} else {
    System.out.println("Maior de 18");
}
else {
    System.out.println("Menor de idade");
}

```

Noté que ao executar a primeira formatação de decisão se (idade >= 18) em caso verdadeiro saber-se-á imediatamente que a idade é maior ou igual a 18. Para saber se a idade é igual a 18, é necessária a execução de outra estrutura de decisão se (idade<18). Em caso afirmativo sabemos que é igual e em caso negativo sabemos que é maior de 18 anos (maior de 18). Isso é que chamamos de estrutura de decisão aninhada.

Vej o código em Java:

```

System.out.println("Idade = 18");
System.out.println("Maior de 18");
System.out.println("Menor de idade");

```

Page 9 of 22 | Palavras: 3,272 | 90%

LUIZ PINHEIRO JUNIOR RAPHAEL SANTOS DA... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Reunião em "General" 18:43

Documentos | Microsoft Word

OPERAÇÃO DE DECISÃO - Microsoft Word

Veja o código em Java:

```
import java.awt.OptionPane;
public class IFAninhado {
    public static void main(String[] args) {
        //Declaração de variáveis
        int idade = 10; //idade auxiliar
        //Entrada de dados
        String aux = JOptionPane.showInputDialog("Entre com a sua idade");
        //Conversão de tipos
        idade = Integer.parseInt(aux);
        //Decisão
        if (idade > 18) { // o primeiro if
            //Comandos se condição for verdadeira
            if (idade == 18)
                JOptionPane.showMessageDialog(null, "Igual a 18");
            else
                JOptionPane.showMessageDialog(null, "Maior de 18");
        } else { // o segundo if
            //Comandos se condição for falsa
            JOptionPane.showMessageDialog(null, "Menor de 18");
        }
    }
}
```

CONTINUAR

Conhecendo um novo exemplo:

Página 10 de 22 | Palavras: 3.272 |

LUIZ PINHEIRO JUNIOR RAPHAEL SANTOS DA... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Maria Eduarda Expedita Oliveira Canto

File Home Insert Draw Design Layout References Mailings Review View Help

AutoSave Document Search Reuse Files

Clipboard Font Styles Paragraph Editing Dictate Editor

Page 4 of 4 8 words Portuguese (Brazil)

Reunião em "General" 18:49

Documentos | Microsoft Word

OPERAÇÃO DE DECISÃO - Microsoft Word

HÉITOR AUGUSTO DE CARVALHO SIlva Assinei, valeu por lembrar

Agora, veja esse segundo exemplo: suponha que você precise fazer um programa em que o usuário insira um número de 1 a 7 e o programa apresente qual é o dia da semana correspondente. Você sabe que domingo é o início da semana, correspondendo ao número 1 e assim sucessivamente.

Como você poderia resolver esse problema?

1. Primeiramente, você deve pensar nas variáveis necessárias:
Como o usuário deverá inserir um número de 1 a 7, é preciso que exista uma variável que vamos chamar de entrada.
Qual seria o tipo da variável entrada?

Se você pensou "inteiro", acertou, pois a variável irá armazenar somente números inteiros.
Será que mais alguma variável é necessária?
A resposta é não, porque vamos fazer a saída diretamente imprimindo na tela.

2. Em seguida, pense no fluxograma, pois é mais fácil de entender e visualizar:
Você pode também optar por escrever o pseudocódigo. Escolha entre o fluxograma ou o pseudocódigo, aquele que você preferir para simbolizar a seqüência lógica do seu programa.

Página 11 de 22 | Palavras: 3.272 |

LUIZ PINHEIRO JUNIOR RAPHAEL SANTOS DA... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

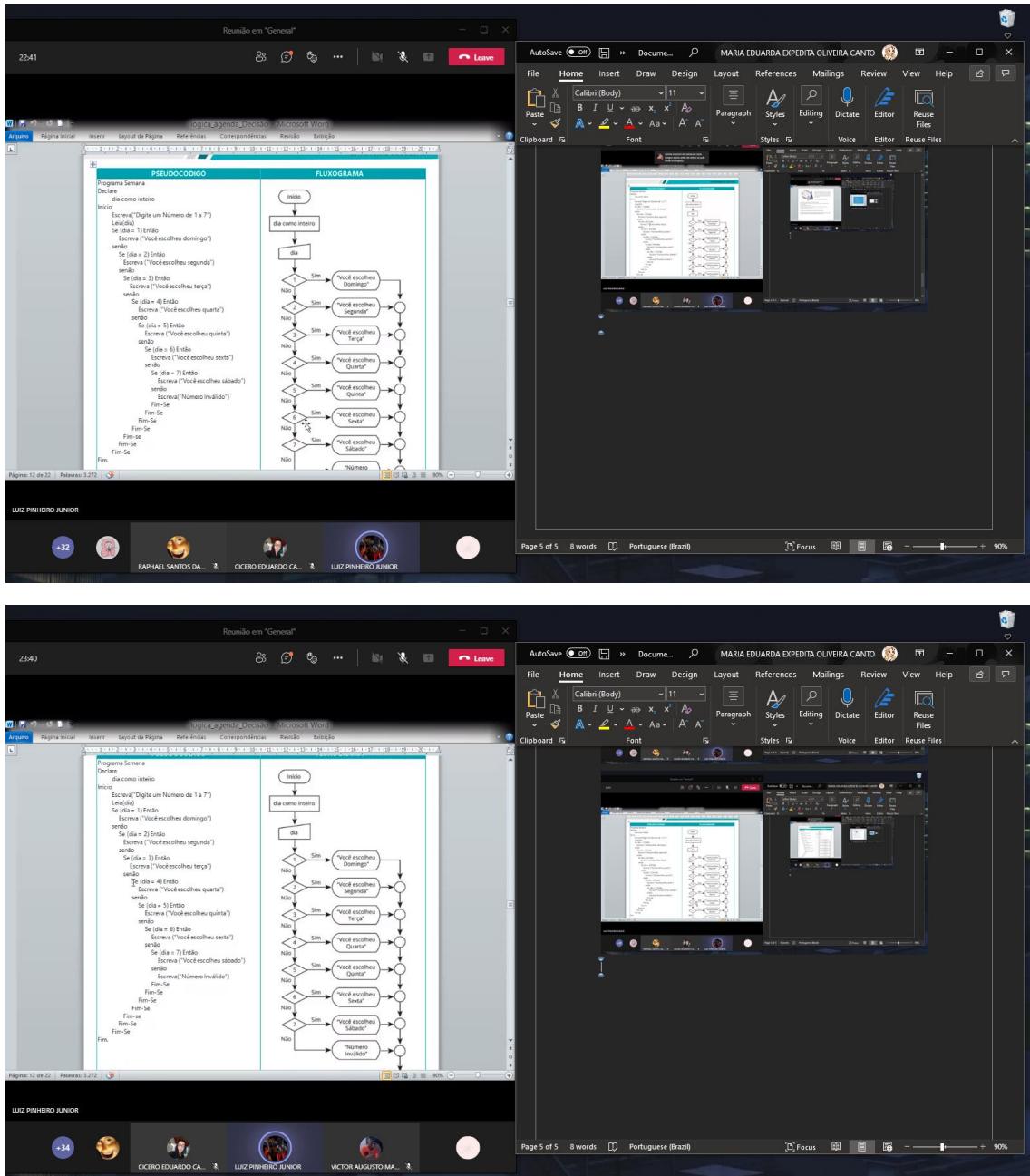
Maria Eduarda Expedita Oliveira Canto

File Home Insert Draw Design Layout References Mailings Review View Help

AutoSave Document Search Reuse Files

Clipboard Font Styles Paragraph Editing Dictate Editor

Page 4 of 4 8 words Portuguese (Brazil)



Os círculos são conectores de linha, no UML AS LINHAS NÃO SE CRUZAM/NÃO PODE.

Reunião em "General"

27:05

logica_agenda_Decisão - Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição Design Layout Ferramentas de Tabela

Ferramentas de Tabela

Leia(dia)

```

Se (dia <= 1) Ento
    Escreva ("Você escolheu domingo")
    senão
        Se (dia > 2) Ento
            Escreva ("Você escolheu segunda")
            senão
                Se (dia > 3) Ento
                    Escreva ("Você escolheu terça")
                    senão
                        Se (dia > 4) Ento
                            Escreva ("Você escolheu quarta")
                            senão
                                Se (dia > 5) Ento
                                    Escreva ("Você escolheu quinta")
                                    senão
                                        Se (dia > 6) Ento
                                            Escreva ("Você escolheu sexta")
                                            senão
                                                Se (dia > 7) Ento
                                                    Escreva ("Você escolheu sábado")
                                                    senão
                                                        Escreva ("Número inválido")
                                                        Fim-Se
                                                        Fim-Se
                                                        Fim-Se
                                                        Fim-Se
                                                        Fim-Se
                                                        Fim-Se

```

dia como intiero

Os círculos são conectores de linhas, no UML AS LINHAS NÃO SE CRUZAM/NÃO PODE

Page 6 of 6 | 22 words | Portuguese (Brazil) | Focus | 90%

LUIZ PINHEIRO JUNIOR +35 CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR VICTOR AUGUSTO MA...

Reunião em "General"

30:29

logica_agenda_Decisão - Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição Design Layout Ferramentas de Tabela

Ferramentas de Tabela

```

import javax.swing.JOptionPane;
public class iFinalizadoSemana {
    public static void main(String[] args) {
        //declaração de variáveis
        int dia; // variável para armazenamento da semana

        //entrada de dados com conversão de tipos juntas
        dia = Integer.parseInt(JOptionPane.showInputDialog("Entre com um
número de 1 a 7"));

        //processamento

        if (dia == 1) { //if 1
            JOptionPane.showMessageDialog(null, "Você escolheu Domingo");
        } else {
            if (dia == 2) { //if 2
                JOptionPane.showMessageDialog(null, "Você escolheu
Segunda");
            } else {
                if (dia == 3) { //if 3
                    JOptionPane.showMessageDialog(null, "Você escolheu
Terça");
                } else {
                    if (dia == 4) { //if 4
                        JOptionPane.showMessageDialog(null, "Você
escolheu Quarta");
                    } else {
                        if (dia == 5) { //if 5
                            JOptionPane.showMessageDialog(null, "Você
escolheu Quinta");
                        } else {
                            if (dia == 6) { //if 6
                                JOptionPane.showMessageDialog(null,
"Você escolheu Sexta");
                            } else {
                                if (dia == 7) { //if 7
                                    JOptionPane.showMessageDialog(null,
"Você escolheu Sábado");
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

Página: 13 de 22 | Palavras: 3.272 | +35 | CICERO EDUARDO CA... | LUIZ PINHEIRO JUNIOR | RODRIGO COTRIN GU...

LUIZ PINHEIRO JUNIOR +35 CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR RODRIGO COTRIN GU...

Pseudocódigo: onde o

Fluxograma: coloca o

Page 7 of 7 | 37 words | Portuguese

Reunião em "General"

25:04

Leave

AutoSave Off

File Home

Clipboard

logica_agenda_Decisão Microsoft Word

Ferramentas de Tabela

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição Design Layout

```
if (dia == 2) { //if 2
    JOptionPane.showMessageDialog(null, "Você escolheu
Segunda");
} else {
    if (dia == 3) { //if 3
        JOptionPane.showMessageDialog(null, "Você escolheu
Terça");
    } else {
        if (dia == 4) { //if 4
            JOptionPane.showMessageDialog(null, "Você
escolheu Quarta");
        } else {
            if (dia == 5) { //if 5
                JOptionPane.showMessageDialog(null, "Você
escolheu Quinta");
            } else {
                if (dia == 6) { //if 6
                    JOptionPane.showMessageDialog(null,
"Você escolheu Sexta");
                } else {
                    if (dia == 7) { //if 7
                        JOptionPane.showMessageDialog(null, "Você
escolheu Sábado");
                    } else {
                        JOptionPane.showMessageDialog(null, "Número Inválido");
                    } // fim do if 7
                } // fim do if 6
            } // fim do if 5
        } // fim do if 4
    } // fim do if 3
} // fim do if 2
} // fim do if 1
} // fim da classe
```

Páginas: 13 de 22 Palavras: 3.272

LUIZ PINHEIRO JUNIOR

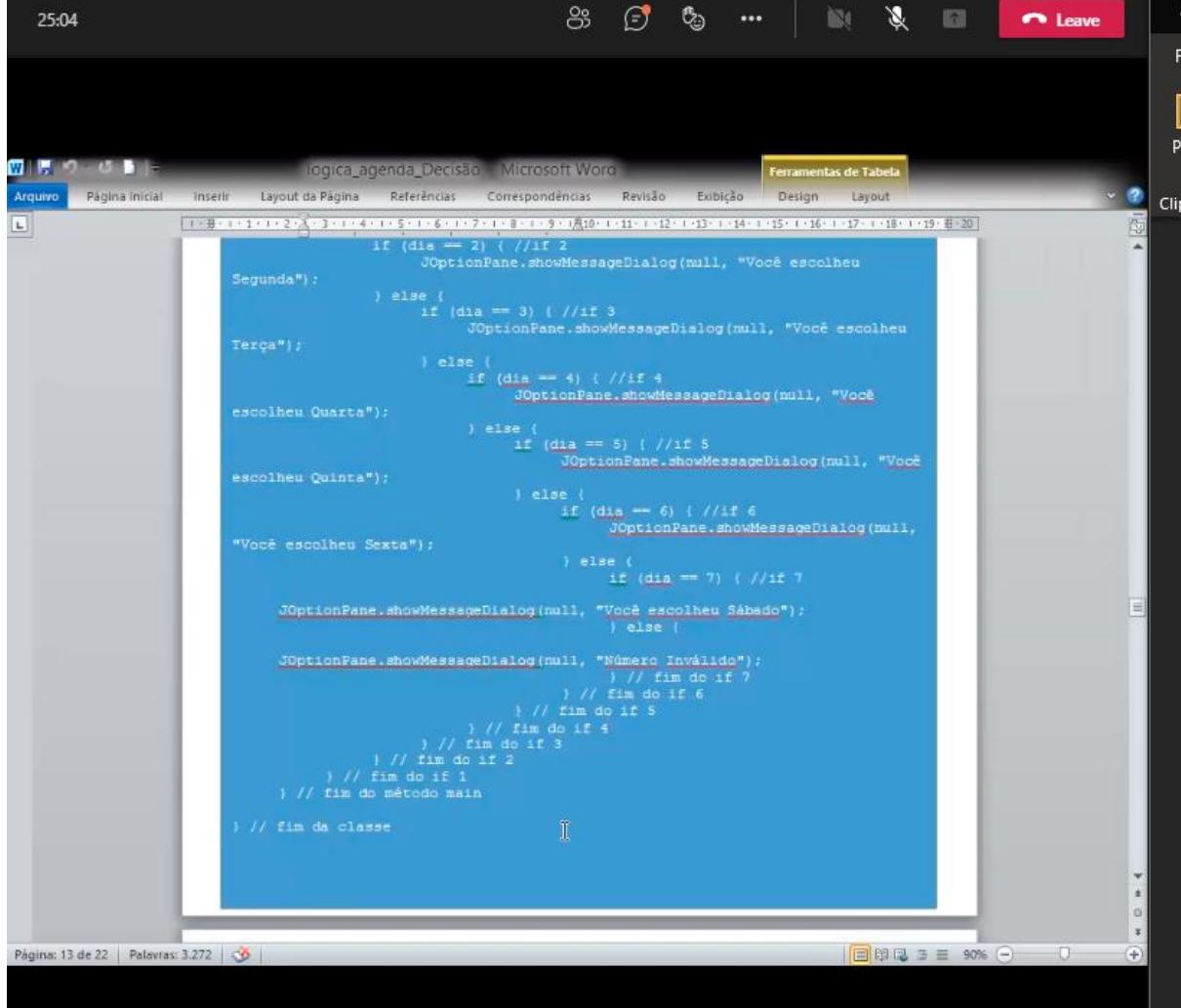
+34

CICERO EDUARDO CA...

LUIZ PINHEIRO JUNIOR

VICTOR AUGUSTO MA...

Page 6 of 6 8 words



Pseudocódigo: onde cria o algoritmo no português estruturado

Fluxograma: coloca os símbolos representativos do algoritmo

Vamos assistir aos vídeos do Prof. Sandro Valéris para complementar nossos estudos:

Lógica – Desvio Condicionais:

```

graph TD
    Start(( )) --> Media{media > 7}
    Media -- Sim --> Approved[Aluno Aprovado]
    Media -- Não --> Failed[Aluno Reprovado]
    Approved --> Fim((Fim))
    Failed --> Fim

```

Lógica e Programação em Java – Desvio Condicional SE / IF:

Página 13 de 22 | Palavras: 3.272 | Page 6 of 6 | 8 words.

LUZ PINHEIRO JUNIOR

RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

AutoSave Document Search MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

ESTRUTURAS DE ALGORITMO

Pseudocódigo

```

Programa ex1
Declarar numero como real
Início
  Escreva ("Entre com um número")
  Lea (numero)
  Se (numero>=100 ) Então
    Escreva ("O número inserido é maior ou igual a 100")
  Senão
    Escreva ("O número inserido é menor que 100")
  Fim-Se
Fim.

```

Fluxograma

```

graph TD
    Inicio([Início]) --> Cond{numero >= 100}
    Cond -- Sim --> Maior["Escreva ('O número inserido é maior ou igual a 100')"]
    Maior --> Fim1[Fim]
    Cond -- Não --> Menor["Escreva ('O número inserido é menor que 100')"]
    Menor --> Fim2[Fim]

```

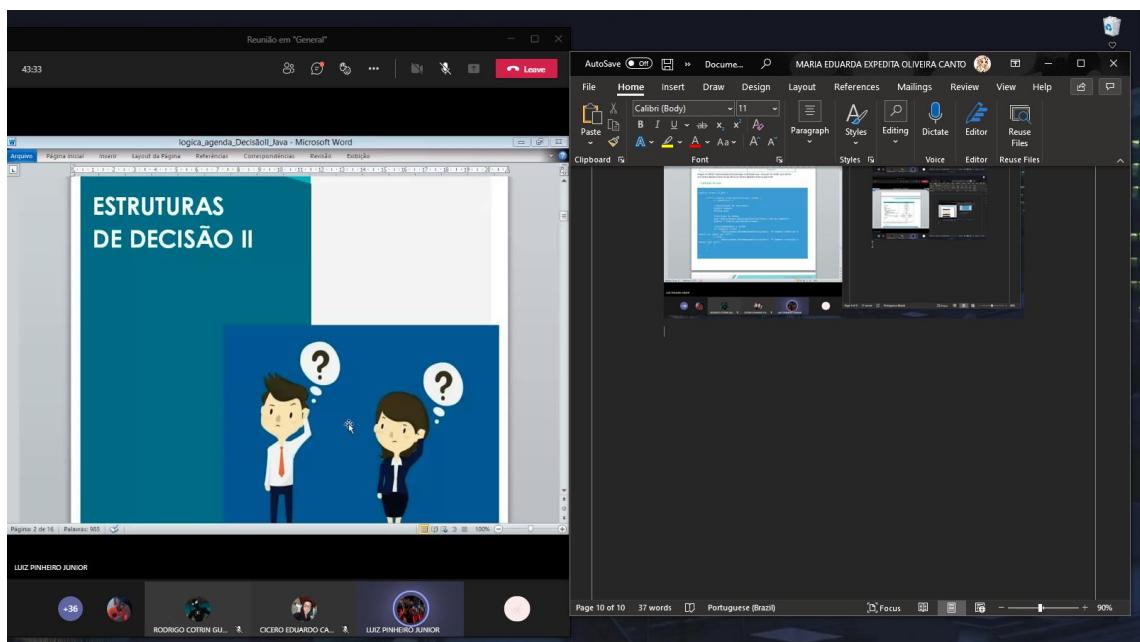
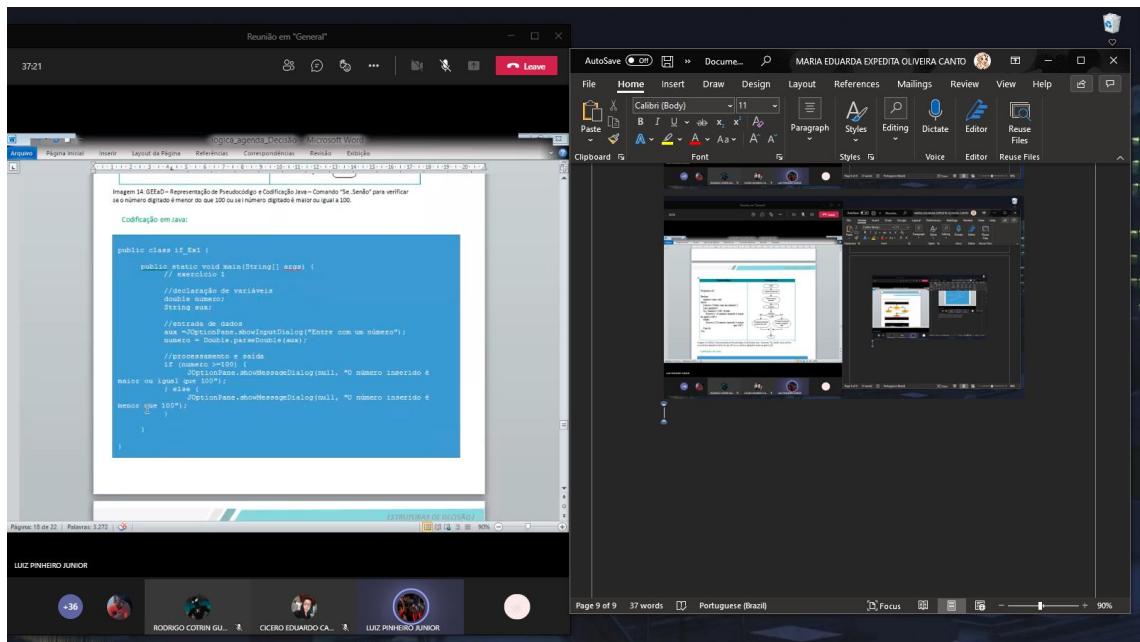
Imagem 14: GTEx0 – Representação de Pseudocódigo e Codificação Java – Comando “Se...Senão” para verificar se o número digitado é menor do que 100 ou se o número digitado é maior ou igual a 100.

Codificação em java:

Página 17 de 22 | Palavras: 3.272 | Page 9 of 9 | 37 words | Portuguese (Brazil)

LUZ PINHEIRO JUNIOR

RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR



Já sabemos que a lógica de programação possui mecanismos que nos permitem tomar decisões dentro de um algoritmo. Sabemos também que esses mecanismos são denominados "Estruturas de Decisão". A novidade é que essas estruturas não se restringem a apenas o "se...senão...fim_se".

E se tivéssemos uma decisão a ser tomada entre **dez opções**? Será que o "se...senão...fim_se" seria a solução mais adequada para essa situação? Será que existe alguma outra estrutura mais adequada para esse tipo de ocorrência?

Existe sim! Essa estrutura é chamada de **Seleção...caso...senão...fim_seleção**.

Estrutura de Decisão caso...senão...fim_seleção

A função principal dessa estrutura é **facilitar a escrita do algoritmo** quando se tem muitos caminhos a serem seguidos a partir de uma decisão. Assim como a estrutura "se...senão...fim_se", é necessário saber quando e como utilizar o "seleção...caso...senão...fim_seleção".

Página 3 de 16 | Palavras: 963

LUIZ PINHEIRO JUNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Page 10 of 10 37 words Portuguese (Brazil) Focus Reuse Files

Já sabemos que a lógica de programação possui mecanismos que nos permitem tomar decisões dentro de um algoritmo. Sabemos também que esses mecanismos são denominados "Estruturas de Decisão". A novidade é que essas estruturas não se restringem a apenas o "se...senão...fim_se".

E se tivéssemos uma decisão a ser tomada entre **dez opções**? Será que o "se...senão...fim_se" seria a solução mais adequada para essa situação? Será que existe alguma outra estrutura mais adequada para esse tipo de ocorrência?

Existe sim! Essa estrutura é chamada de **Seleção...caso...senão...fim_seleção**.

Estrutura de Decisão caso...senão...fim_seleção

A função principal dessa estrutura é **facilitar a escrita do algoritmo** quando se tem muitos caminhos a serem seguidos a partir de uma decisão. Assim como a estrutura "se...senão...fim_se", é necessário saber quando e como utilizar o "seleção...caso...senão...fim_seleção".

Página 3 de 16 | Palavras: 963

LUIZ PINHEIRO JUNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Page 11 of 11 37 words Portuguese (Brazil) Focus Reuse Files

Reunião em "General" 44:10

ESTRUTURAS DE DECISÃO II

PSEUDOCÓDIGO

```
Selecione Caso (variável)
  Caso condição 01
  (comando(s))
  Caso condição 02
  (comando(s))
  Caso condição 03
  (comando(s))
  ...
  Senão
  (comando(s))
Fim-Seleção
```

FLUXOGRAMA

JAVA

```
switch (variável) {
    case condição 01:
        (comando(s));
    break;
    case condição 02:
        (comando(s));
    break;
    case condição 03:
        (comando(s));
    break;
    ...
    default:
        (comando(s));
}
```

Se recapitarmos o último exemplo de Estrutura de Decisão Aninhada do tópico anterior, no qual o programa exiba o dia da semana de acordo com o número digitado pelo usuário, notamos que o código do programa fica relativamente confuso devido à grande quantidade de comandos de seleção (se). A tendência dessa confusão é aumentar conforme o número de comandos de decisão aninhados for crescendo. Isso, é claro, levando em conta que estamos realizando a comparação sempre com a mesma variável.

Para essas situações podemos utilizar a estrutura "seleciona caso...senão...fim_seleção" do pseudocódigo ou a estrutura switch-case do Java.

Antes de continuar a leitura, assista a aula do prof. Sandro Valérios:

LUIZ PINHEIRO JÚNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JÚNIOR

Page 3 of 16 | Palavras: 905 | Leave

File Home Insert Draw Design References Mailings Review View Help

AutoSave Document MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Font

Page 11 of 11 | 37 words | Portuguese (Brazil) Focus

Reunião em "General" 44:43

ESTRUTURAS DE DECISÃO II

PSEUDOCÓDIGO

```
Selecione Caso (variável)
  Caso condição 01
  (comando(s))
  Caso condição 02
  (comando(s))
  Caso condição 03
  (comando(s))
  ...
  Senão
  (comando(s))
Fim-Seleção
```

FLUXOGRAMA

JAVA

```
01:
  (comando(s));
  break;
02:
  (comando(s));
  break;
03:
  (comando(s));
  ...
  break;
  default:
    (comando(s));
}
```

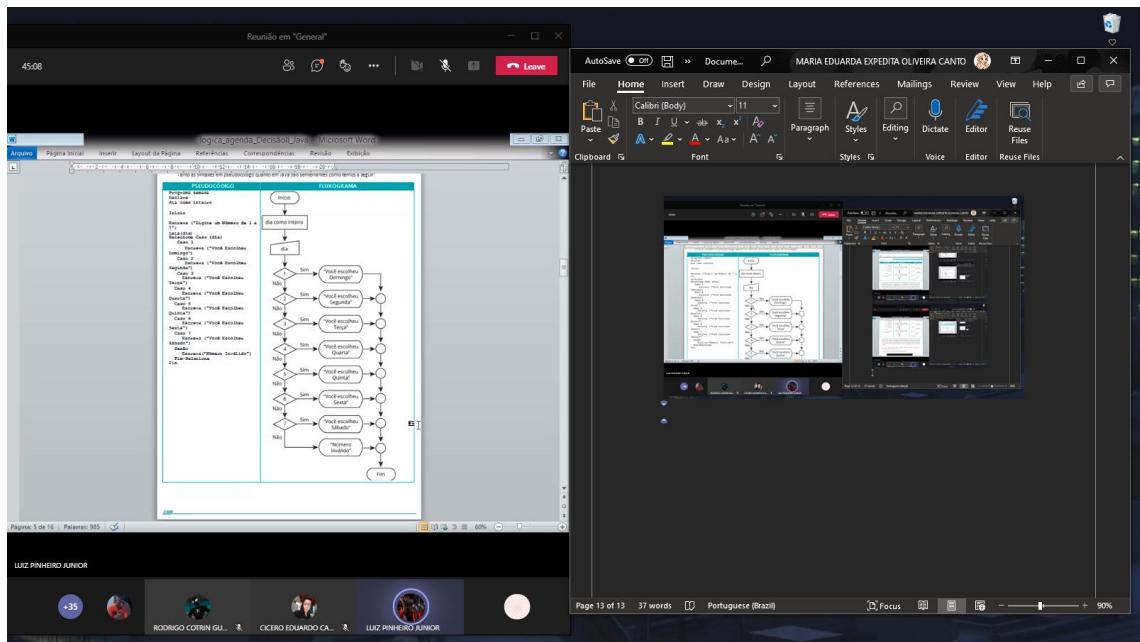
Se recapitarmos o último exemplo de Estrutura de Decisão Aninhada do tópico anterior, no qual o programa exiba o dia da semana de acordo com o número digitado pelo usuário, notamos que o código do programa fica relativamente confuso devido à grande quantidade de comandos de seleção (se). A tendência dessa confusão é aumentar conforme o número de comandos de decisão aninhados for crescendo. Isso, é claro, levando em conta que estamos realizando a comparação sempre com a mesma variável.

Para essas situações podemos utilizar a estrutura "seleciona caso...senão...fim_seleção" do pseudocódigo ou a estrutura switch-case do Java.

Antes de continuar a leitura, assista a aula do prof. Sandro Valérios:

LUIZ PINHEIRO JÚNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JÚNIOR

Page 12 of 12 | 37 words | Portuguese (Brazil) Focus



O SE e o CASO não têm diferença nenhuma, os dois são a mesma coisa

The image shows a Microsoft Word document with Java code for a class named "menuCaso". The code includes a main method that reads a command-line argument and prints a menu based on the input. The menu options correspond to the decisions in the flowchart: "1º dia", "2º dia", "3º dia", "4º dia", "5º dia", "6º dia", "7º dia", "8º dia", "9º dia", "10º dia", "11º dia", "12º dia", "13º dia", "14º dia", "15º dia", "16º dia", "17º dia", "18º dia", "19º dia", "20º dia", "21º dia", "22º dia", "23º dia", "24º dia", "25º dia", "26º dia", "27º dia", "28º dia", "29º dia", "30º dia", "31º dia", "32º dia", "33º dia", "34º dia", "35º dia", "36º dia", "37º dia", "38º dia", "39º dia", "40º dia", "41º dia", and "42º dia".

Below the Word document is a presentation slide with the title "O SE e o CASO não têm diferença nenhuma, os dois são a mesma coisa". The slide contains two screenshots of Microsoft Word, one showing the Java code and another showing the flowchart from the previous slide.

Reunião em "General"

52:43

logica_agenda_DecisãoII_Uvila - Microsoft Word

AutoSave Documento... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Página Inicial Insert Layout da Página Referências Correspondências Revisão Edição

Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

O SE e o CASO não têm diferença nenhuma, os dois são a mesma coisa

Page 5 of 16 | Palavras: 965

LUZ PINHEIRO JUNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Page 13 of 13 | 52 words | Portuguese (Brazil) Focus

1 // se é dia de niver
else
if (dia == 12 || dia == 1 || dia == 2 || dia == 3 || dia == 4 || dia == 5 || dia == 6 || dia == 7 || dia == 8 || dia == 9 || dia == 10 || dia == 11 || dia == 12 || dia == 13 || dia == 14 || dia == 15 || dia == 16 || dia == 17 || dia == 18 || dia == 19 || dia == 20 || dia == 21 || dia == 22 || dia == 23 || dia == 24 || dia == 25 || dia == 26 || dia == 27 || dia == 28 || dia == 29 || dia == 30 || dia == 31)
{
 System.out.println("Parabéns!");
}
else
{
 System.out.println("Nada de comemorar");
}

E exatamente o mesmo programa de exemplo de Estrutura de decisão animada da agenda anterior. Com essa nova estrutura que foi apresentada, o código foi escrito de maneira muito mais elegante. O fluxograma permaneceu o mesmo sem alteração nenhuma.

Reunião em "General"

52:57

logica_agenda_DecisãoII_Uvila - Microsoft Word

AutoSave Documento... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Página Inicial Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Page 14 of 14 | 52 words | Portuguese (Brazil) Focus

LUZ PINHEIRO JUNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

VOCÊ NO COMANDO Exercícios sobre estrutura de seleção:

LISTRUTURAS DE DECISÃO

Analogamente ao pseudocódigo, em Java, após fazermos a entrada do dado pelo usuário, o comando switch (dia) irá comparar o valor armazenado na variável dia com um valor de 1 a 7 em cada comando case:

Exemplo:

Se o usuário entrar com o valor 3, retomos dia = 3 e ao executarmos o comando case 3: (dia=3), como o resultado da comparação será verdadeiro, ele executará o comando JOptionPane.

showMessageDialog(null, "Você escolheu Terça");

e o comando break; e por fim finalizará o programa.

O comando break tem a finalidade de parar a execução do comando switch, uma vez que já foi executada a ação necessária (comparação verdadeira) e não há a necessidade de se continuar com o comando switch.

Caso nenhuma comparação resulte em uma resposta verdadeira, o comando default: é executado gerando uma mensagem de "número inválido" para o usuário.

Reunião em "General"

53:08

logica_agenda_DecisãoII_Uvila Microsoft Word

Arquivo Página Inicial Insert Layout da Página Referências Correspondências Revisão Edição

AutoSave Documento MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

2. Elberto deseja fazer um menu de seleção de um programa utilizando o comando de seleção múltipla. O menu consiste nas opções de:

1. Cadastrar usuário
2. Alterar dados
3. Excluir Usuário

Como Elberto faria esse programa? Elabore uma mensagem diferente para ser exibida para o usuário ao selecionar cada um dos itens dos menus.

3. Além do menu apresentado na questão anterior, Elberto precisa fazer alguns submenus com a seguinte estrutura:

1. Cadastrar usuário
2. Alterar dados
3. Alterar nome
2. Alterar Endereço
3. Alterar telefone
4. Alterar RG
3. Excluir usuário

Altere o programa do exercício anterior para incluir este submenu e exiba uma mensagem correspondente de acordo com o menu escolhido pelo usuário.

4. Crie um programa para uma calculadora com cinco operações básicas (soma, subtração, multiplicação, divisão) e exiba a saída da operação desejada. A estrutura deve ser a seguinte:

```

    menu()
        opcao = int(input("Digite a opção desejada:"))
        if opcao == 1:
            soma()
        elif opcao == 2:
            subtracao()
        elif opcao == 3:
            multiplicacao()
        elif opcao == 4:
            divisao()
        else:
            print("Opção inválida")
    
```

Página 5 de 16 | Palavras: 983

LUZ PINHEIRO JUNIOR

RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Page 14 of 14 52 words Portuguese (Brazil)

Focus Reuse Files

90%

Reunião em "General"

53:22

logica_agenda_DecisãoII_Uvila Microsoft Word

Arquivo Página Inicial Insert Layout da Página Referências Correspondências Revisão Edição

AutoSave Documento MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

1.

Pseudocódigo	Fluxograma
<pre> Programa ex1 Declarar opcao como inteiro inicio leia("Entre com o código do produto:") Seleciona Caso (opcao) Caso condicao 100 Escreva ("Lápis preto nº2") Caso condicao 150 Escreva ("Lápis de cor branca") Caso condicao 200 Escreva ("Caneta azul") Caso condicao 300 Escreva ("Caneta vermelha") Caso condicao 400 Escreva ("giz de cera 12 unid.") Caso condicao 300 Escreva ("Lápis de cor branca") Caso condicao 310 Escreva ("Tresma de sulfite A4") Caso condicao 320 Escreva ("Tresma escolar verde") Caso condicao 370 Escreva ("Caderno universitário 100fls") Caso condicao 500 Escreva ("Caderno brechura 50fls") Senão Escreva ("Produto não cadastrado") Finsel Fim-sel Fim. </pre>	

Página 5 de 16 | Palavras: 983

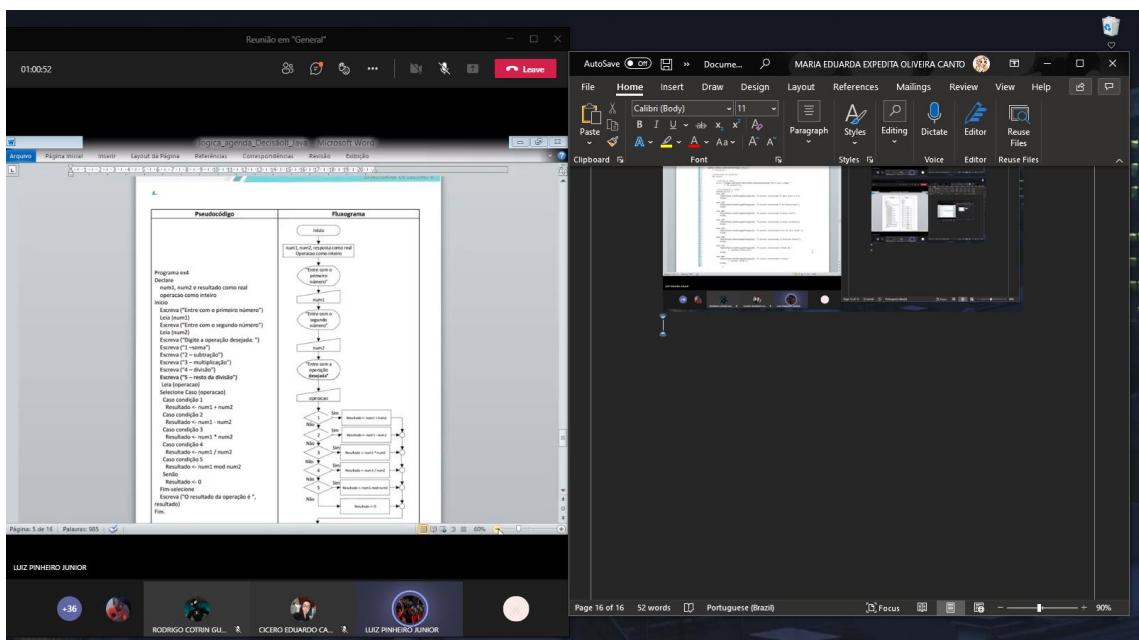
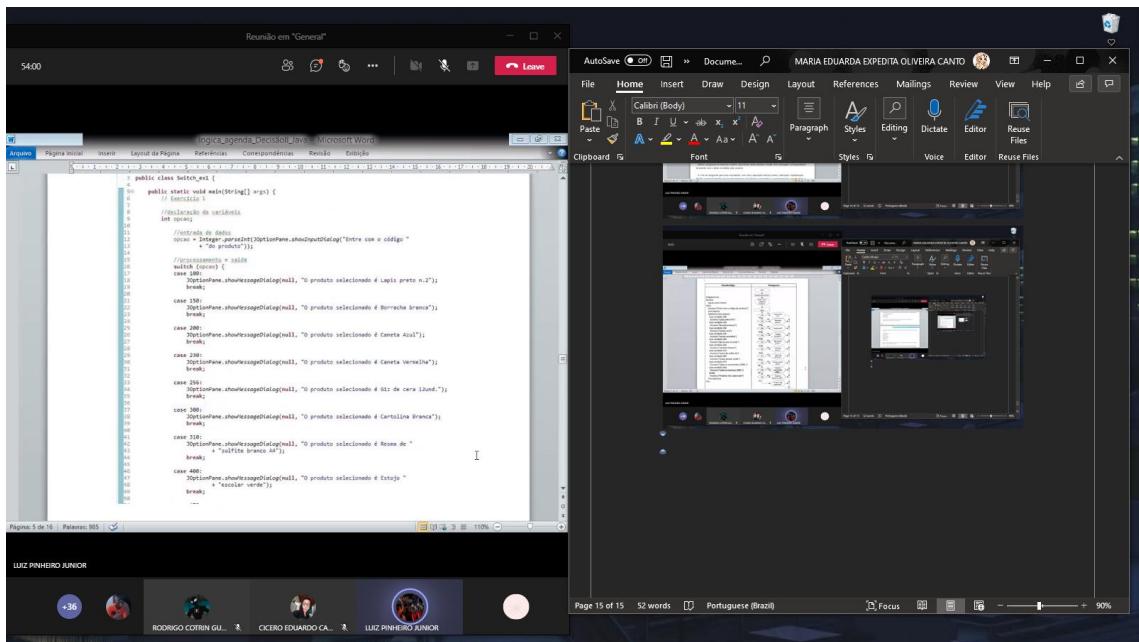
LUZ PINHEIRO JUNIOR

RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Page 15 of 15 52 words Portuguese (Brazil)

Focus Reuse Files

90%



Reunião em "General" 01:00:56

Diagrama de fluxo para a lógica da agenda.

```

    graph TD
        Inicio([Início]) --> LendoEntrada[Leia número]
        LendoEntrada --> Operacao[Operação]
        Operacao --> Resposta[Resposta]
        Resposta --> Fim([Fim])
        
        subgraph Operacao [Operação]
            A1[Entrar com o primeiro número] --> Soma[Somar]
            Soma --> A2[Entrar com o segundo número]
            A2 --> Multiplicar[Multiplicar]
            Multiplicar --> A3[Entrar com o resultado]
            A3 --> Dividir[Dividir]
            Dividir --> A4[Entrar com o divisor]
            A4 --> Resto[Resto]
            Resto --> A5[Entrar com o resultado]
            A5 --> Sair[Sair]
        end
    
```

Página 5 de 16 | Palavras: 963

LUIZ PINHEIRO JUNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR

Reunião em "General" 01:03:10

Código Java para a lógica da agenda.

```

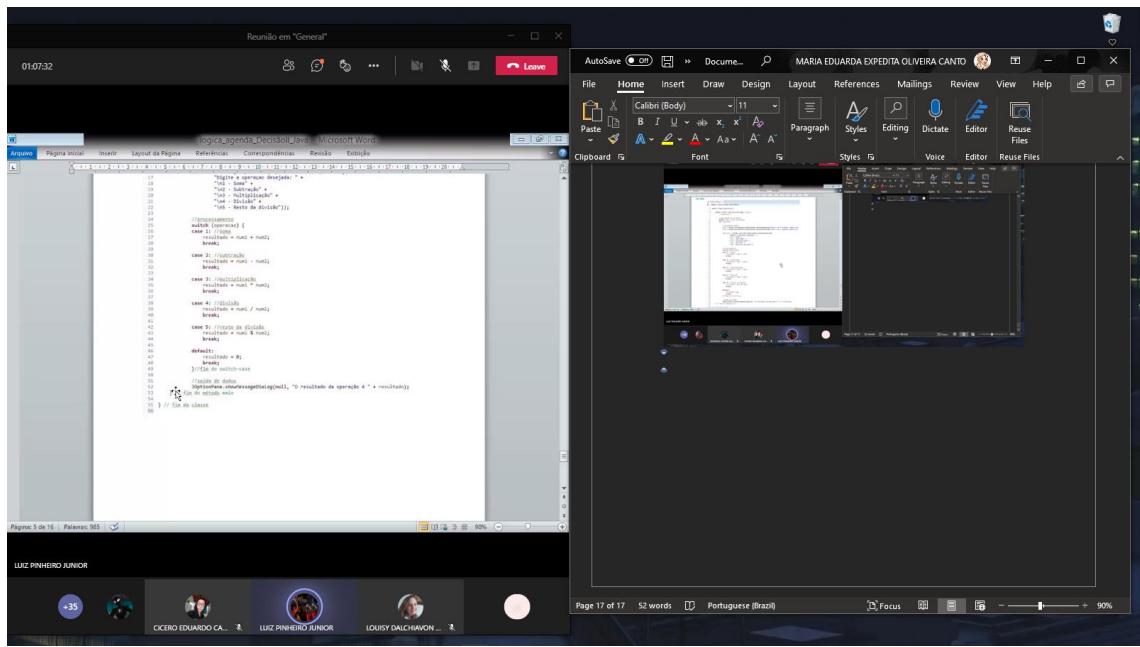
    package agenda;
    import java.util.Scanner;

    public class Agenda {
        public static void main(String[] args) {
            // Entrada de dados
            Scanner entrada = new Scanner(System.in);
            double resultado = 0;
            String operacao = "digite para saber o resultado da operação";

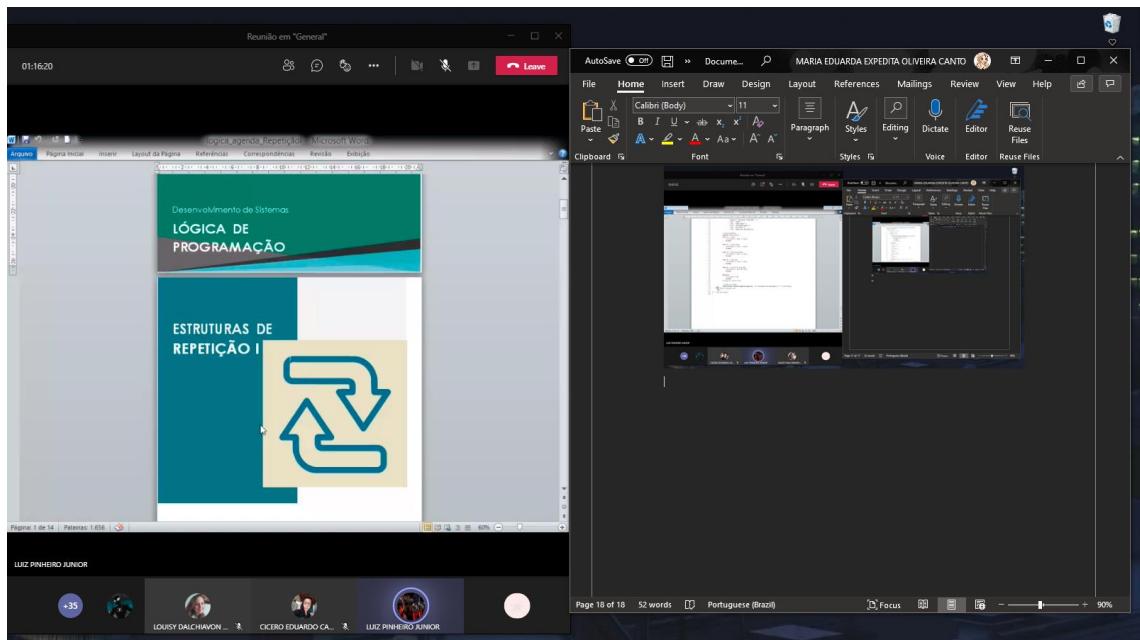
            operacao = Integer.parseInt(entrada.nextLine());
            if (operacao == 1) {
                resultado = resultado + resultado;
                System.out.println("O resultado da operação é: " + resultado);
            } else if (operacao == 2) {
                resultado = resultado * resultado;
                System.out.println("O resultado da operação é: " + resultado);
            } else if (operacao == 3) {
                resultado = resultado / resultado;
                System.out.println("O resultado da operação é: " + resultado);
            } else if (operacao == 4) {
                resultado = resultado % resultado;
                System.out.println("O resultado da operação é: " + resultado);
            } else {
                System.out.println("Operação inválida");
            }
        }
    }
    
```

Página 5 de 16 | Palavras: 963

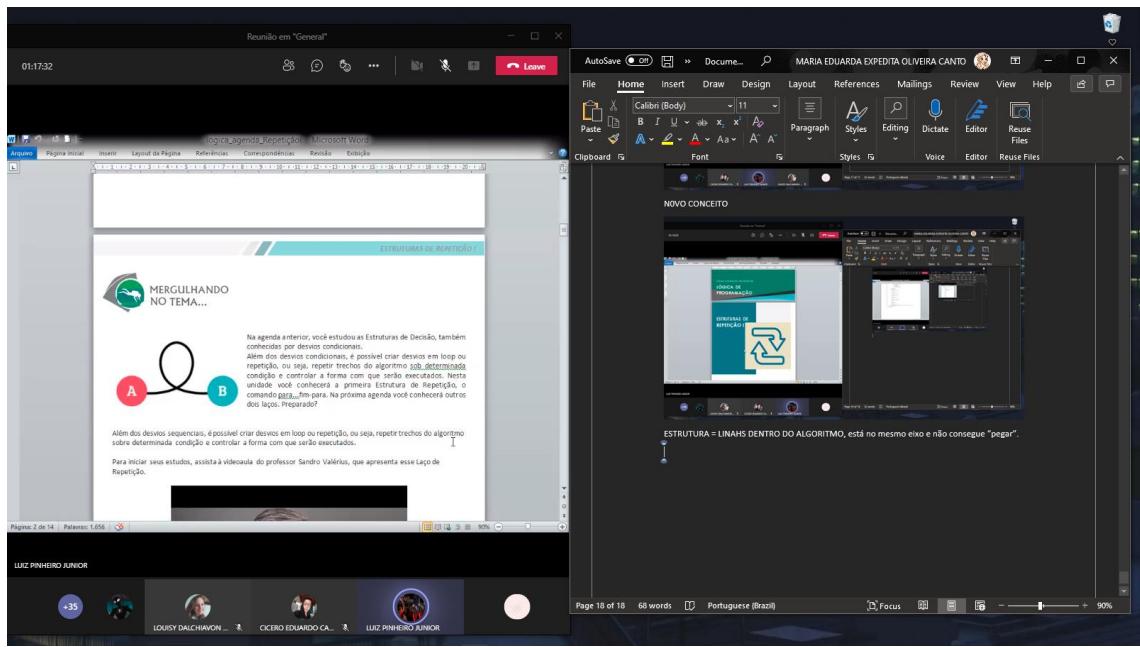
LUIZ PINHEIRO JUNIOR RODRIGO COTRIN GU... CICERO EDUARDO CA... LUIZ PINHEIRO JUNIOR



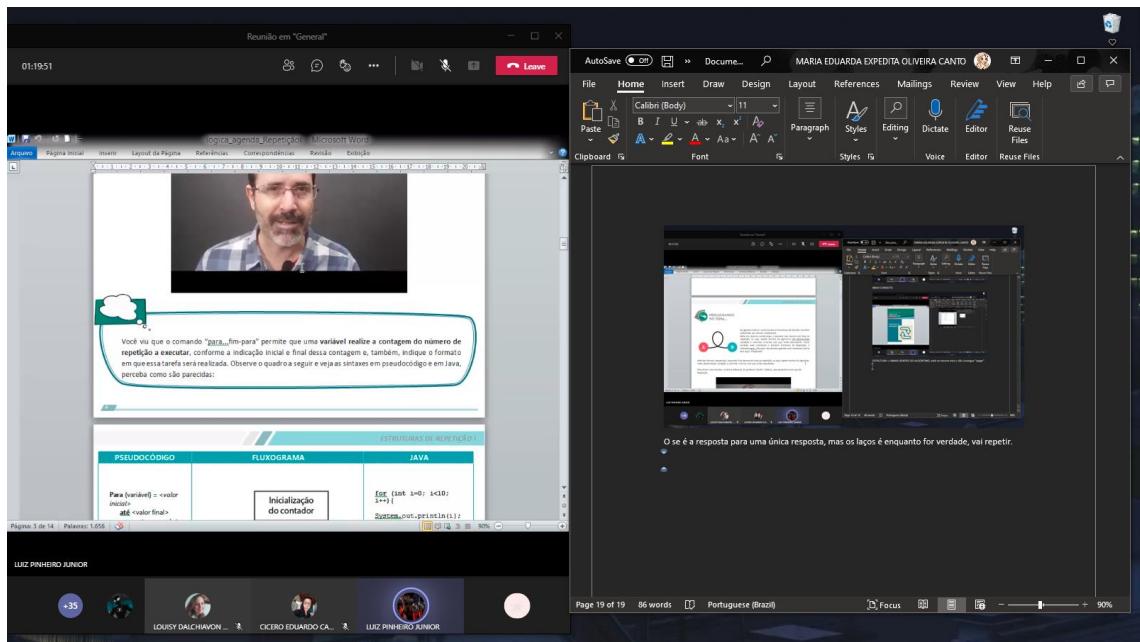
NOVO CONCEITO



ESTRUTURA = LINHAS DENTRO DO ALGORITMO, está no mesmo eixo e não consegue “pegar”.



O se é a resposta para uma única resposta, mas os laços é enquanto for verdade, vai repetir.



Reunião em "General" 01:19:57

ESTRUTURAS DE REPETIÇÃO I

PSEUDOCÓDIGO	FLUXOGRAAMA	JAVA
<pre> Para [variável] = > valor inicial Até [valor final] passo [argumento] faça [comandos] fim para </pre>		<pre> for (int i=0; i<10; System.out.println(i); } </pre>

Page 3 of 14 - Palavras: 1.056

LUIZ PINHEIRO JUNIOR

LOUISY DALCHIAVON ... CICERO EDUARDO CA ... LUIZ PINHEIRO JUNIOR

Page 19 of 19 86 words Portuguese (Brazil)

Focus Reuse Files

Incremento = acrescentar 1/+1 , decremento = tira 1/-1

Contador: quantas vezes tem que realizar aquele loop e para que ele não fique infinito. Você quer 100 repetições (para/até): Incluir o 100 = 1 ao 100, não inclui o 100 = 0 a 99. Passo = incremento/acrescentar. Loopings = quantas vezes passa pela variável. Neste exemplo é de 0 a 9.

Reunião em "General" 01:25:12

ESTRUTURAS DE REPETIÇÃO I

PSEUDOCÓDIGO	FLUXOGRAAMA	JAVA
<pre> Para [variável] = > valor inicial Até [valor final] passo [argumento] faça [comandos] fim para </pre>		<pre> for (int i=0; i<10; System.out.println(i); } </pre>

Observe que no Fluxograma aparece a palavra **contador**, e tanto no Pseudocódigo quanto no Java não aparecem. Calma! Ela aparece, porém de outra forma. Veja que no Pseudocódigo tem (**[variável]**) e no Java aparece **i**. Ambas são os contadores que aparecem no Fluxograma.

Confira um exemplo:

A execução é muito fácil. Para facilitar a compreensão, veja um exemplo de um programa que mostra ao usuário os números de 0 a 9 na tela:

```

0 5
1 6
2 7
3 8
4 9
  
```

Page 4 of 14 - Palavras: 1.056

LUIZ PINHEIRO JUNIOR

LOUISY DALCHIAVON ... CICERO EDUARDO CA ... LUIZ PINHEIRO JUNIOR

Page 20 of 20 141 words Portuguese (Brazil)

Focus Reuse Files

0208

TPA

ESTRUTURAS DE DECISÃO II

Como se fosse um misto, junta o que forma no seu local e de trazido de outros locais.

TPA

Page 5 of 5 | 369 words | Portuguese (Brazil)

MERGULHANDO NO TEMA...

Já sabemos que a lógica de programação possui mecanismos nos que permitem tomar decisões dentro de um algoritmo. Sabemos também que esses mecanismos são denominados "Estruturas de Decisão". A novidade é que essas estruturas não se restringem a apenas o "se... senão... fim_se".

E se tivéssemos uma decisão a ser tomada entre **dez opções**? Será que o "se...senão...fim_se" seria a solução mais apropriada para essa situação? Será que existe alguma outra estrutura mais adequada para esse tipo de ocorrência?

Existe sim! Essa estrutura é chamada de **Selezione...caso...senão...fim_seleção**.

A diagram illustrating a selection structure:

```
graph LR; A((A)) --> B((B)); A --> C((C)); A --> D((D)); A --> E((E));
```

Estrutura de Decisão caso... senão... fim_seleção

TPA

Page 5 of 5 | 369 words | Portuguese (Brazil)

Reunião em "Geral" 13:33

lógica_agenda_DecisãoII.java Microsoft Word

A função principal dessa estrutura é facilitar a escrita do algoritmo quando se tem muitos caminhos a serem seguidos a partir de uma decisão. Assim como a estrutura "se...senão...fim_se", é necessário saber quando e como utilizar o "seleciona...caso...senão...fim_seleciona".

A estrutura "seleciona...caso...senão...fim_seleciona" do pseudocódigo corresponde à estrutura **switch-case** do Java. Elas permitem que a seleção correta seja feita a partir da comparação entre o valor do conteúdo de variável e uma lista definida durante a programação. Ao encontrar a correspondência correta, os respectivos comandos são executados e as demais opções ignoradas.

ESTRUTURAS DE DECISÃO II

PSEUDOCÓDIGO	FLUXOGRAMA	JAVA
<pre>Seleciona Caso (variável) Caso condição 01 (comando(s)) Caso condição 02 (comando(s)) Caso condição 03 (comando(s)) ... Senão (comando(s)) Fin-Seleciona</pre>		<pre>switch (variável){ case condição 01: (comando(s)); break; case condição 02: (comando(s)); break; case condição 03: (comando(s)); break; ... default: (comando(s)); }</pre>

Página: 3 de 16 | Palavras: 905 |

LUIZ PINHEIRO JUNIOR

HB GABRIEL TAYLOR SOU... RODRIGO MARTINEZ LUIZ PINHEIRO JUNIOR

Page 5 of 5 369 words Portuguese (Brazil) Focus

Reunião em "Geral" 13:40

lógica_agenda_DecisãoII.java Microsoft Word

Se recapitularmos o último exemplo de Estrutura de Decisão Aninhada do tópico anterior, no qual o programa exiba o dia de semana de acordo com o número digitado pelo usuário, notamos que o código do programa fica relativamente confuso devido à grande quantidade de comandos de seleção (se). A

ESTRUTURAS DE DECISÃO II

PSEUDOCÓDIGO	FLUXOGRAMA	JAVA
<pre>Seleciona Caso (variável) Caso condição 01 (comando(s)) Caso condição 02 (comando(s)) Caso condição 03 (comando(s)) ... Senão (comando(s)) Fin-Seleciona</pre>		<pre>switch (variável){ case condição 01: (comando(s)); break; case condição 02: (comando(s)); break; case condição 03: (comando(s)); break; ... default: (comando(s)); }</pre>

Página: 3 de 16 | Palavras: 905 |

LUIZ PINHEIRO JUNIOR

HB GABRIEL TAYLOR SOU... RODRIGO MARTINEZ LUIZ PINHEIRO JUNIOR

Page 6 of 6 369 words Portuguese (Brazil) Focus

Reunião em "Geral" 15:24

logica_agenda_DecisãoIIJava Microsoft Word

Tendo as informações produzidas quanto ao dia da semana, como temos a seguir:

```

    graph TD
        Start((Início)) --> Pergunta[Pergunta: Sabendo que o dia de hoje é dia Sábado, qual é o dia da semana?]
        Pergunta --> Resposta1[Resposta: "Tôgita um dia de 1 a 6"]
        Resposta1 --> Sim1{Sim}
        Sim1 --> Dia1[Dia 1]
        Dia1 --> Sim2{Sim}
        Sim2 --> Dia2[Dia 2]
        Dia2 --> Sim3{Sim}
        Sim3 --> Dia3[Dia 3]
        Dia3 --> Sim4{Sim}
        Sim4 --> Dia4[Dia 4]
        Dia4 --> Sim5{Sim}
        Sim5 --> Dia5[Dia 5]
        Dia5 --> Sim6{Sim}
        Sim6 --> Dia6[Dia 6]
        Dia6 --> Sim7{Sim}
        Sim7 --> Dia7[Dia 7]
        Dia7 --> Nao1{Não}
        Nao1 --> Erro1[Não respondeu corretamente]
        Erro1 --> Fim1((Fim))
        Nao1 --> Nao2{Não}
        Nao2 --> Erro2[Não respondeu corretamente]
        Erro2 --> Fim1
    
```

Página: 5 de 16 | Palavras: 955 |

LUIZ PINHEIRO JUNIOR

GABRIEL TAYLOR SOU... RODRIGO MARTINES... LUIZ PINHEIRO JUNIOR

MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Page 7 of 7 369 words Focus 90%

Reunião em "Geral" 16:33

logica_agenda_DecisãoIIJava Microsoft Word

`public class mainCenario {
 public static void main(String[] args) {
 //Entrada de dados
 Scanner teclado = new Scanner(System.in);
 System.out.println("Por favor, entre com o número de dias: ");
 int n = teclado.nextInt();
 //Processo de dados com conversão de tipos inteiros
 String s = Integer.toString(n);
 //Saída de dados
 System.out.println("Entre com o dia da semana: ");
 String dia = teclado.nextLine();
 switch (dia) {
 case "1":
 System.out.println("Sábado");
 break;
 case "2":
 System.out.println("Domingo");
 break;
 case "3":
 System.out.println("Segunda");
 break;
 case "4":
 System.out.println("Terça");
 break;
 case "5":
 System.out.println("Quarta");
 break;
 case "6":
 System.out.println("Quinta");
 break;
 case "7":
 System.out.println("Sexta");
 break;
 default:
 System.out.println("Número Inválido");
 }
 }
}`

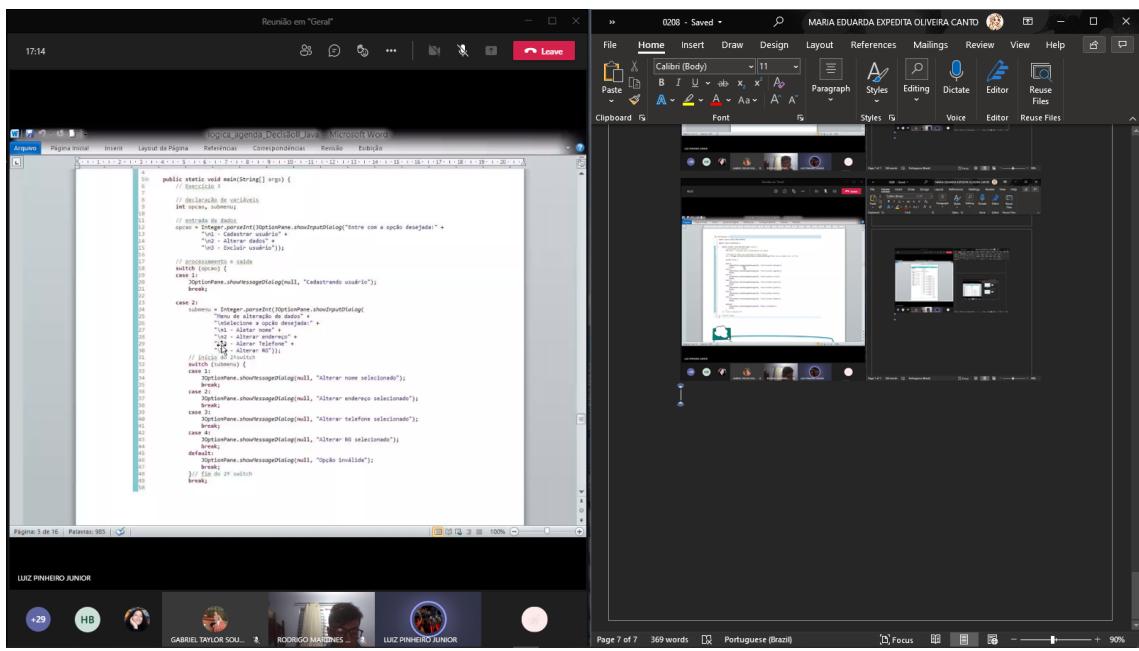
Página: 5 de 16 | Palavras: 955 |

LUIZ PINHEIRO JUNIOR

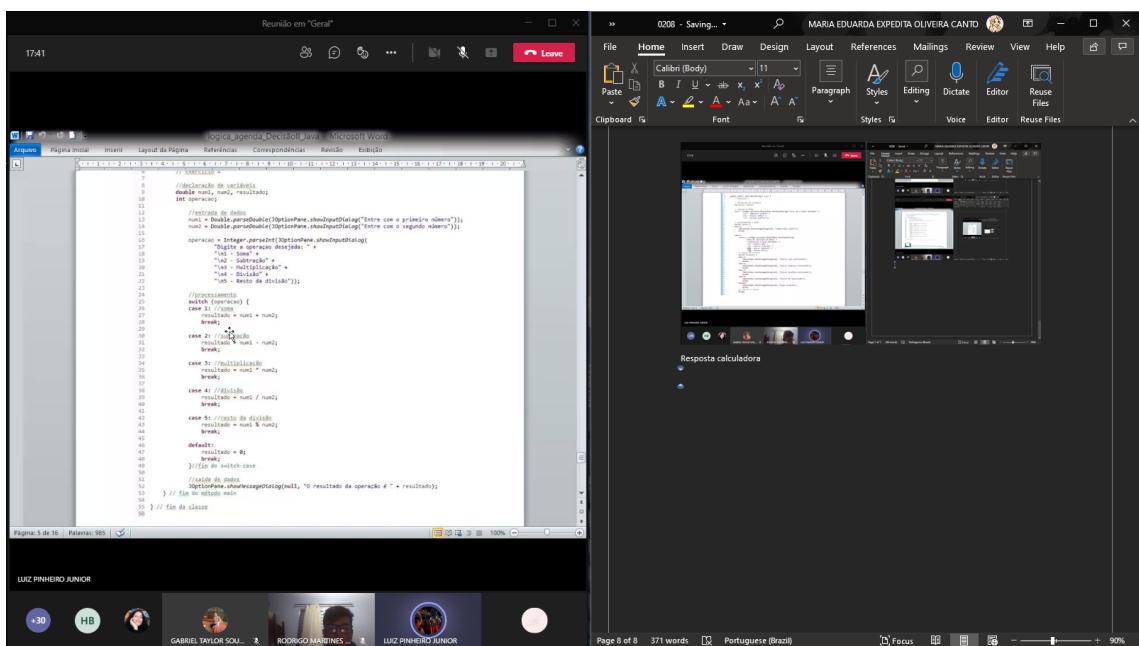
GABRIEL TAYLOR SOU... RODRIGO MARTINES... LUIZ PINHEIRO JUNIOR

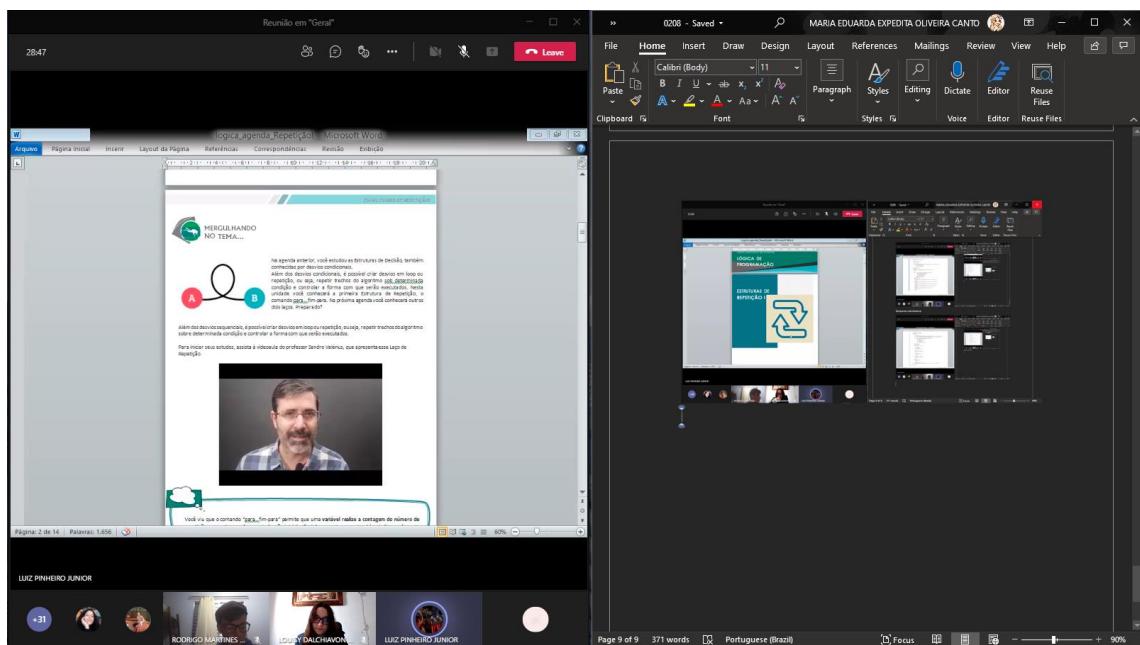
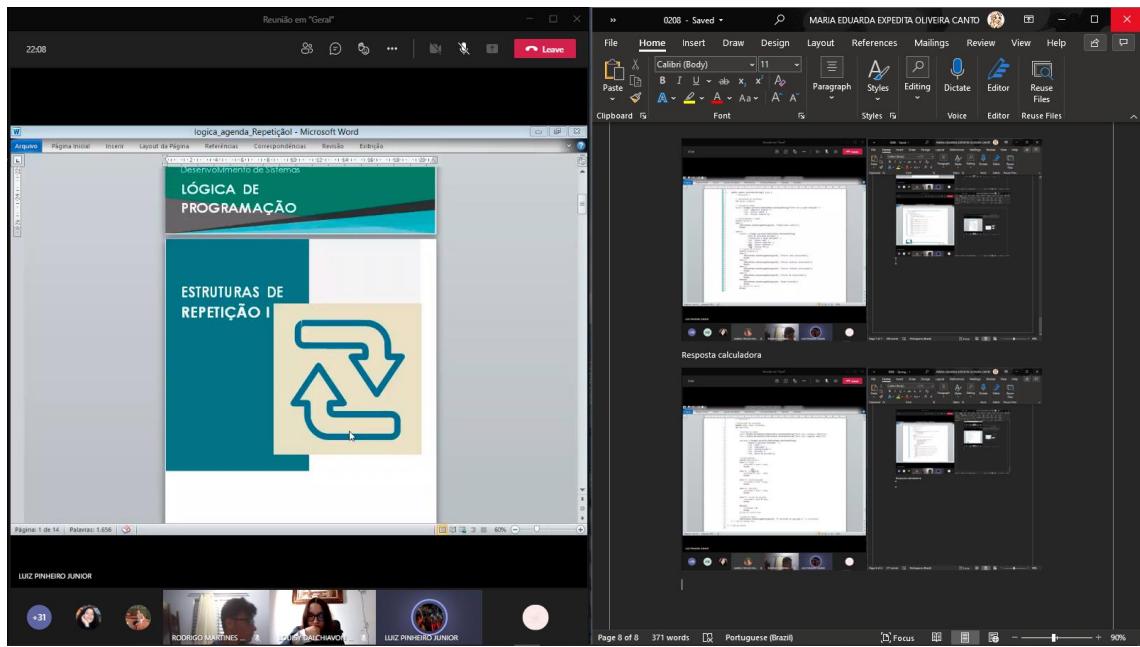
MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

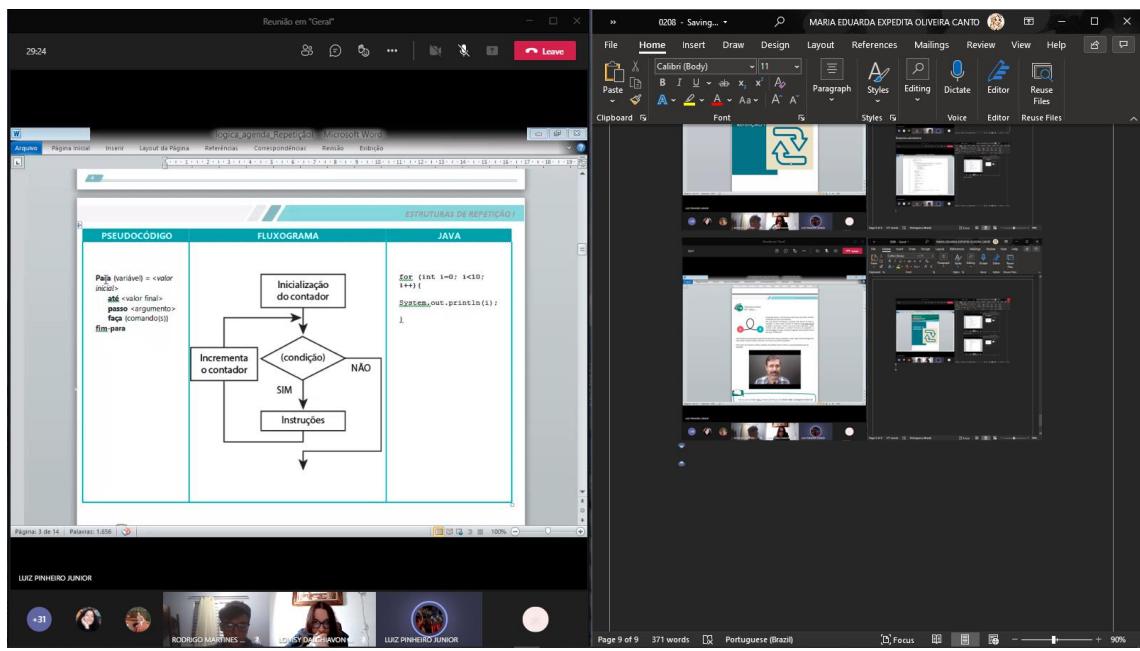
Page 7 of 7 369 words Focus 90%



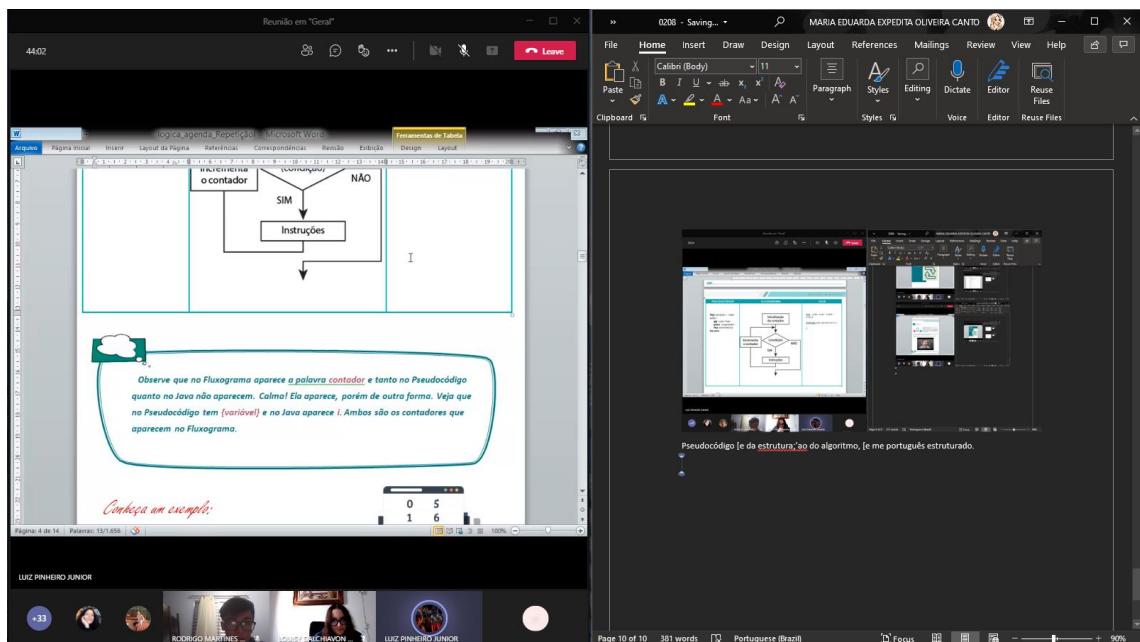
Resposta calculadora







Pseudocódigo [e da estrutura;'ao do algoritmo, [e me português estruturado.



Reunião em "Geral" 44:27

logica_agenda_Repetição.docx Microsoft Word

Ferramentas de Tabela

Arquivo Página inicial Início Layout da Página Referências Correspondências Revisão Exibição Design Layout

Observe que no Fluxograma aparece a palavra **contador** e tanto no Pseudocódigo quanto no Java não aparecem. Calma! Ela aparece, porém de outra forma. Veja que no Pseudocódigo tem (**variável**) e no Java aparece **i**. Ambos são os contadores que aparecem no Fluxograma.

Conheça um exemplo:

A execução é muito fácil. Para facilitar a compreensão, veja um exemplo de um programa que mostra ao usuário os números de 0 a 9 na tela:

Página: 4 de 14 | Palavras: 51.656 | Português (Brasil)

LUIZ PINHEIRO JUNIOR

RODRIGO MARTINS, LOUISY CHAVON, LUIZ PINHEIRO JUNIOR

Page 10 of 10 381 words Portuguese (Brazil) Focus

File Home Insert Draw Design Layout References Mailings Review View Help

0208 - Saved MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Font Styles Voice Editor Reuse Files

Clipboard

Font

Styles

Voice

Editor

Reuse Files

Editing Dictate

Editor

Reuse Files

Reunião em "Geral" 44:51

logica_agenda_Repetição.docx Microsoft Word

Ferramentas de Tabela

Arquivo Página inicial Início Layout da Página Referências Correspondências Revisão Exibição Design Layout

ESTRUTURAS DE REPETIÇÃO

PSEUDOCÓDIGO	FLUXOGRAAMA
<pre>Programa mostrar números de 0 até 9 Declarar contador como inteiro Início contador < 0; Para contador de 0 até 9 passo 1 faça Escreva (contador) Fim para Fim.</pre>	<pre> graph TD inicio([início]) --> contador0[contador = 0] contador0 --> contadorP[contador = contador + 1] contadorP --> cond{contador < 10} cond -- SIM --> escreva[Escreva (contador)] escreva --> contadorP cond -- NÃO --> fim </pre>

Página: 5 de 14 | Palavras: 51.656 | Português (Brasil)

LUIZ PINHEIRO JUNIOR

RODRIGO MARTINS, LOUISY CHAVON, LUIZ PINHEIRO JUNIOR

Page 11 of 11 381 words Portuguese (Brazil) Focus

File Home Insert Draw Design Layout References Mailings Review View Help

0208 - Saved MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Font Styles Voice Editor Reuse Files

Clipboard

Font

Styles

Voice

Editor

Reuse Files

Editing Dictate

Editor

Reuse Files

Reunião em "Geral" 45:30

lógica_agenda_Repetição in Microsoft Word

Programa mostrar números de 0 até 9

```

    Declarar
        contador como inteiro
    Início
        contador < 0;
        Para contador de 0 até 9
            passo 1 faça
                Escreva (contador)
            Fim para
        Fim.
    
```

Página: 5 de 14 | Palavras: 571,656 | 100% | Page 11 of 11 | 381 words | Portuguese (Brazil) | Focus | 90%

LUIZ PINHEIRO JUNIOR

+33 RODRIGO MARTINES LOUIS DALCHAVON LUIZ PINHEIRO JUNIOR

Reunião em "Geral" 49:56

lógica_agenda_Repetição in Microsoft Word

Apenas para você entender melhor, o pseudocódigo inicia-se com a declaração da variável `contador` com tipo de dados `inteiro`, ou seja, esta variável irá receber apenas números inteiros. Após a leitura dessa variável, iniciada com o valor `0`, ocorre o incremento de mais 1 por meio da instrução `passo 1` até o valor `10`. Esse tipo de estrutura de repetição pode ser usado toda vez que houver a necessidade de repetir trechos finitos, em que se conhece os valores inicial e final.

Para z de 0 até 9 ----- 123456789

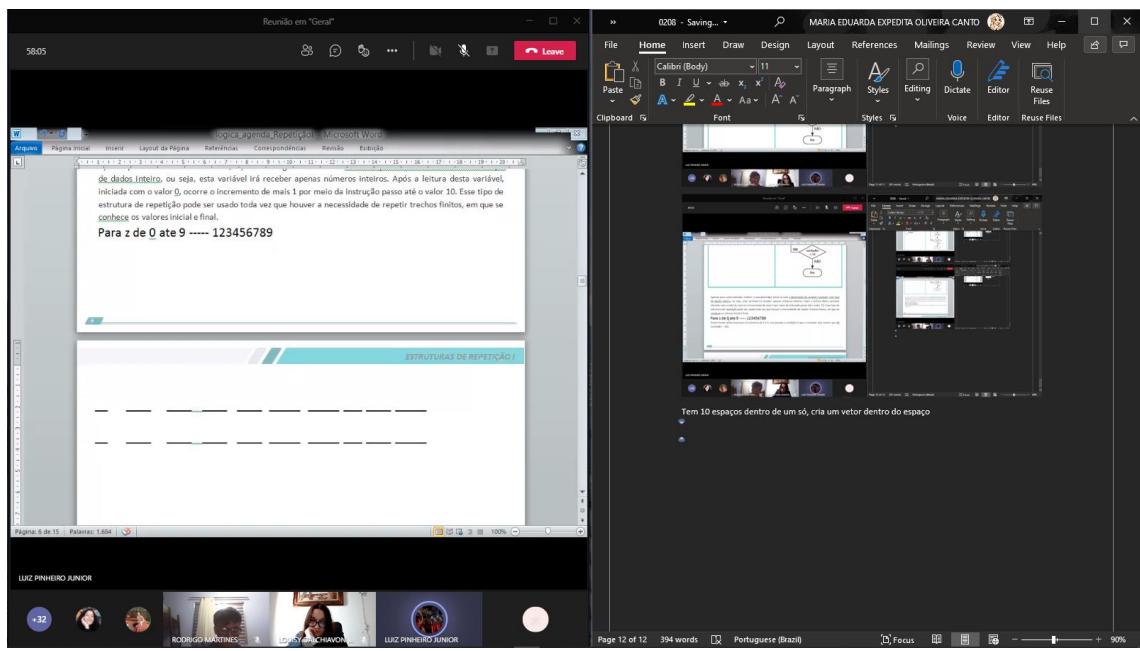
Dessa forma serão impressos os números de 0 a 9, isso porque a condição é que o contador seja menor que `10` (`contador < 10`).

Página: 5 de 14 | Palavras: 1,664 | 100% | Page 12 of 12 | 381 words | Portuguese (Brazil) | Focus | 90%

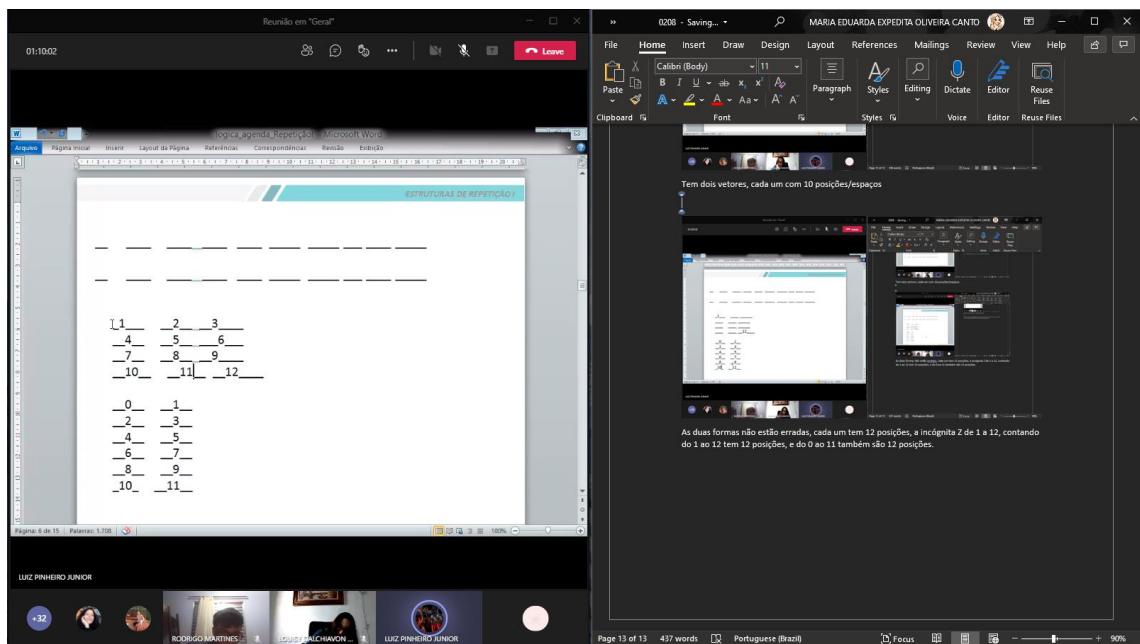
LUIZ PINHEIRO JUNIOR

+33 RODRIGO MARTINES LOUIS DALCHAVON LUIZ PINHEIRO JUNIOR

Tem 10 espaços dentro de um só, cria um vetor dentro do espaço



Tem dois vetores, cada um com 10 posições/espaços



As duas formas não estão erradas, cada um tem 12 posições, a incógnita Z de 1 a 12, contando do 1 ao 12 tem 12 posições, e do 0 ao 11 também são 12 posições.

01:16:15

Reunião em "Geral"

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibições

0208 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

V1 v1 v2 v3
V1 1 2 3
V2 4 5 6
V3 7 8 9
V4 10 11 12

0 1
2 3
4 5
6 7
8 9
10 11

1,1 1,2 1,3
2,1 2,2 2,3

Page 14 of 14 437 words Português (Brazil)

Vetor linha = VL1; vetor coluna = VC1

01:18:22

Reunião em "Geral"

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibições

0208 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

V1 v1 v2 v3
V1 1 2 3
V2 4 5 6
V3 7 8 9
V4 10 11 12

0 1
2 3
4 5
6 7
8 9
10 11

1,1 1,2 1,3
2,1 2,2 2,3

Vetor linha = VL1; vetor coluna = VC1

1,1 (linha 1, coluna 1); 1,2 (linha 1, coluna 2); 2,1 (linha 2, coluna 1), ... é sempre primeiro a linha e depois a coluna. A tabela do Excel é matriz e vetor. 0 1 é a célula, o um número.

Page 14 of 14 460 words Português (Brazil)

The screenshot shows a Microsoft Word document titled "lógica_agenda_Repetição.docx". On the left, there is a diagram illustrating a matrix concept. It features three horizontal rows labeled Vc1, vc2, and vc3 above a grid of numbers from 1 to 12. Below the grid, the text "1,1 1,2 1,3" and "2,1 2,2 2,3" is shown, with "Celula" written underneath. On the right, a video call interface is visible, showing participants and a presentation slide about vectors and matrices.

Isso seria a matriz, porque a matriz é a intersecção da linha e da coluna.

0908

TPA

The screenshot shows a Microsoft Word document titled "lógica_agenda_Repetição.docx". The main content is a flowchart titled "ALGORITMO" illustrating a counter mechanism. It includes steps like "Iniciação do contador", "Verifica o contador", "S/N", and "Instruções". A note below the flowchart states: "Observe que no fluxograma aparece a palavra contendo, e tanto no pseudocódigo quanto no伪代码 não aparecem. Certo é que aparece, porém de outra forma, veja que no pseudocódigo tem (verdade) e no Java aparece true. Ambos são os condicionais que aparecem no fluxograma." At the bottom, there is an example of a Java code snippet showing a counter from 0 to 9. On the right, a video call interface is visible.

Reunião em "General"

13:56

lógica_agenda_Repetição.docx Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição

AutoSave Documento... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Paste Calibri (Body) 11 Paragraph Styles Editing Dictate Editor Reuse Files

Font Styles Voice Editor Reuse Files

Fluxograma

```

graph TD
    Inicio([Início]) --> Contador[Contador = 0]
    Contador --> Cond{contador < 10}
    Cond -- SIM --> Contador[contador = contador + 1]
    Contador -- NÃO --> Fim([Fim])
    
```

Código de exemplo:

```

Programa mostrar números de 0 até 9
Declare
    contador como inteiro
Início
    contador <- 0
    Para contador de 0 até 9
    passo 1
        Escreva (contador)
    Fim para
Fim.
    
```

Fluxograma

Observe que no fluxograma anterior a **atribuição** é feita no **Fluxograma**, quando no código não aparecem. Caso não aparece, pode de extrair forma, visto que na linguagem C tem **declarar** e não aparece. Ambas são os considerados que aparecem no fluxograma.

Exemplo de código:

A execução é muito fácil para facilitar a compreensão, veja um exemplo de um programa que mostra ao usuário os números de 0 a 9 na tela.

Fluxograma

Programa mostrar números de 0 até 9
Declare
 contador como inteiro

INÍCIO

Fluxograma

INÍCIO

contador = 0

contador < 10

contador = contador + 1

contador

SIM

contador <= 10

NAO

FIM

Page 1 of 1 0 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

RODRIGO MARTINES LUCAS DANTAS ROSE LUIZ PINHEIRO JUNIOR

Page 1 of 1 0 words Portuguese (Brazil)

14:07

lógica_agenda_Repetição.docx Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição

AutoSave Documento... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Paste Calibri (Body) 11 Paragraph Styles Editing Dictate Editor Reuse Files

Font Styles Voice Editor Reuse Files

Fluxograma

```

graph TD
    Inicio([Início]) --> Contador[contador = 0]
    Contador --> Cond{contador < 10}
    Cond -- SIM --> Contador[contador = contador + 1]
    Contador -- NÃO --> Fim([Fim])
    
```

Código de exemplo:

```

Programa mostrar números de 0 até 9
Declare
    contador como inteiro
Início
    contador <- 0
    Para contador de 0 até 9
    passo 1
        Escreva (contador)
    Fim para
Fim.
    
```

Fluxograma

Apesar para você não ver melhor, o fluxograma media-se com a declaração de que o contador é tipo inteiro. Devido a isso, seu valor é iniciado em zero. A cada iteração, o valor é incrementado em 1, ou seja, é incrementado com o valor q, ocorre o incremento de mais 1 por meio da instrução posse ser 0 e virar 10, desse tipo de estrutura de repetição pode se usar tanto todo vez que for necessário de repetir trechos finitos, em que se destaca a menor incia e final.

Fluxograma

Programa mostrar números de 0 até 9
Declare
 contador como inteiro

INÍCIO

Fluxograma

INÍCIO

contador = 0

contador < 10

contador = contador + 1

contador

SIM

contador <= 10

NAO

FIM

Page 1 of 1 0 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

RODRIGO MARTINES LUCAS DANTAS ROSE LUIZ PINHEIRO JUNIOR

Page 1 of 1 0 words Portuguese (Brazil)

POSIÇÃO é o valor que já assumiu.

Reunião em "General"

15:44

MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

0908 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Diagrama de fluxo:

```
graph TD; A(( )) --> B(( )); B --> C(( )); C --> D(( )); D --> E(( )); E --> F(( )); F --> G(( )); G --> H(( )); H --> I(( )); I --> J(( )); J --> K(( )); K --> L(( )); L --> M(( )); M --> N(( )); N --> O(( )); O --> P(( )); P --> Q(( )); Q --> R(( )); R --> S(( )); S --> T(( )); T --> U(( )); U --> V(( )); V --> W(( )); W --> X(( )); X --> Y(( )); Y --> Z(( )); Z --> A;
```

POSIÇÃO é o valor que já assumiu.

Diagrama de fluxo:

```
graph TD; A(( )) --> B(( )); B --> C(( )); C --> D(( )); D --> E(( )); E --> F(( )); F --> G(( )); G --> H(( )); H --> I(( )); I --> J(( )); J --> K(( )); K --> L(( )); L --> M(( )); M --> N(( )); N --> O(( )); O --> P(( )); P --> Q(( )); Q --> R(( )); R --> S(( )); S --> T(( )); T --> U(( )); U --> V(( )); V --> W(( )); W --> X(( )); X --> Y(( )); Y --> Z(( )); Z --> A;
```

Veja agora o código em Java:

```
public class Loops { public static void main(String[] args) { int i = 0; // i é o resultado, que será utilizado para o laço de repetição for. for (int i = 0; i < 10; i++) { System.out.println(i); } // i é o resultado, que será utilizado para o laço de repetição for. } // Deixa o bloco. } // Deixa a classe.
```

Comentários do Professor: Informo, primeiramente, que não, desculpem a minha falta de humor, mas é só isso. Apesar de ser muito legal, não é uma tarefa muito interessante quando se trata de programação genérica, variável final e/ou inicial incrementada 1. Ou seja, o resultado é mostrar as sequências dos números 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

Página 6 de 14 | Palavras: 1.658 | Page 6 of 6 676 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

JA RODRIGO MARTINS LUCAS MIGUEL ROSE LUIZ PINHEIRO JUNIOR

Reunião em "General"

15:49

MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

0908 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Diagrama de fluxo:

```
graph TD; A(( )) --> B(( )); B --> C(( )); C --> D(( )); D --> E(( )); E --> F(( )); F --> G(( )); G --> H(( )); H --> I(( )); I --> J(( )); J --> K(( )); K --> L(( )); L --> M(( )); M --> N(( )); N --> O(( )); O --> P(( )); P --> Q(( )); Q --> R(( )); R --> S(( )); S --> T(( )); T --> U(( )); U --> V(( )); V --> W(( )); W --> X(( )); X --> Y(( )); Y --> Z(( )); Z --> A;
```

POSIÇÃO é o valor que já assumiu.

Diagrama de fluxo:

```
graph TD; A(( )) --> B(( )); B --> C(( )); C --> D(( )); D --> E(( )); E --> F(( )); F --> G(( )); G --> H(( )); H --> I(( )); I --> J(( )); J --> K(( )); K --> L(( )); L --> M(( )); M --> N(( )); N --> O(( )); O --> P(( )); P --> Q(( )); Q --> R(( )); R --> S(( )); S --> T(( )); T --> U(( )); U --> V(( )); V --> W(( )); W --> X(( )); X --> Y(( )); Y --> Z(( )); Z --> A;
```

Veja agora o código em Java:

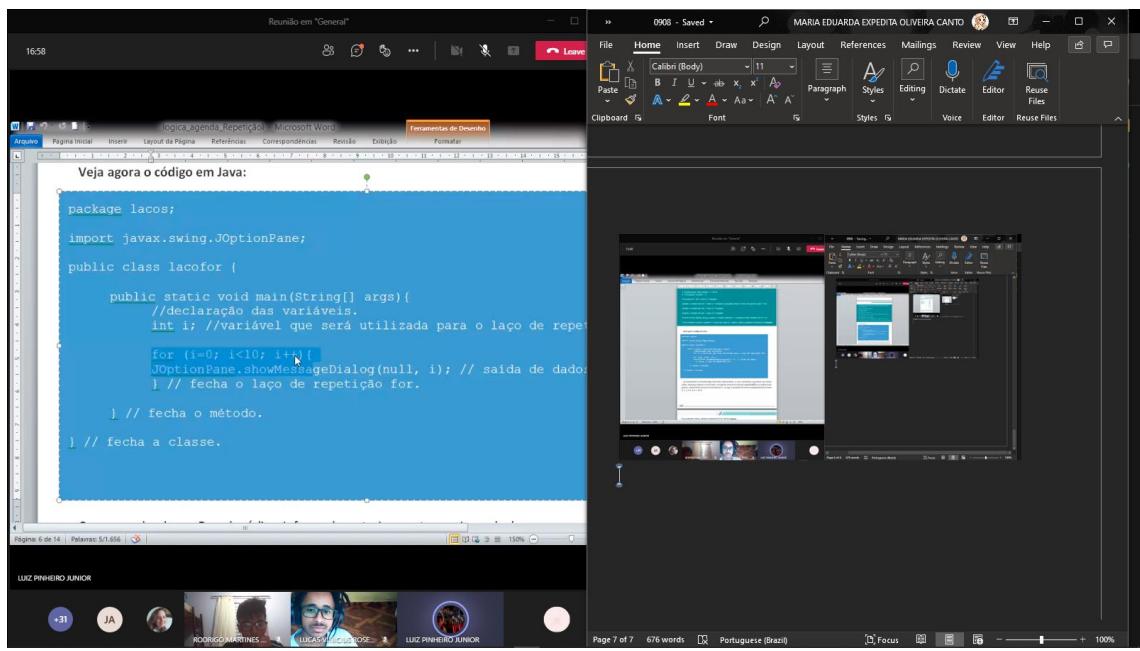
```
public class Loops { public static void main(String[] args) { int i = 0; // i é o resultado, que será utilizado para o laço de repetição for. for (int i = 0; i < 10; i++) { System.out.println(i); } // i é o resultado, que será utilizado para o laço de repetição for. } // Deixa o bloco. } // Deixa a classe.
```

Comentários do Professor: Informo, primeiramente, que não, desculpem a minha falta de humor, mas é só isso. Apesar de ser muito legal, não é uma tarefa muito interessante quando se trata de programação genérica, variável final e/ou inicial incrementada 1. Ou seja, o resultado é mostrar as sequências dos números 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

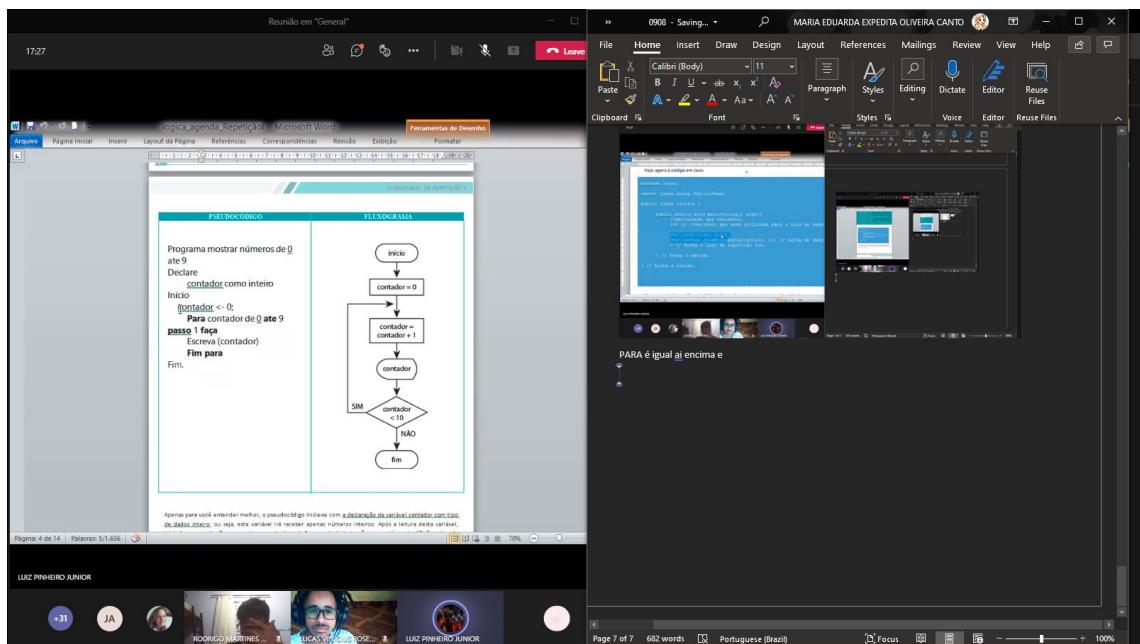
Página 6 de 14 | Palavras: 1.658 | Page 6 of 6 676 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

JA RODRIGO MARTINS LUCAS MIGUEL ROSE LUIZ PINHEIRO JUNIOR



PARA é igual ai encima e vai ser maior que 10 e crescente (i++)



Tem que escrever no pseudocódigo, não pode representar que nem no Java.

Reunião em "General" 19:32

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição Design Layout

VOCÊ NO MUNDO

Perceba como poderíamos elaborar um programa utilizando a estrutura para-fim para que o usuário mointe os números de 01 até 10 no teclado.

Reflexão como podemos resolver essa questão antes de prosseguir a leitura.

Agora, dê uma olhada em um pseudocódigo que resolve esse exercício:

Programa mostrar números
Programa mostrarmultiplos
dezenas
dados como inteiros
int i
Para i igual a 1 ate 10 passo 1 feça
escrever(i)
Fim-para

Fim.

Veja, então, o Java:

```
public class ex02 {  
    public static void main(String[] args) {  
        System.out.println("Os números de 0 a 10 são:  
        for (int i=0; i<10; i++) //laço de repetição for.  
        System.out.println(i); // saída de dados  
    }  
}
```

Este é o resultado:

```
Programa ex02 [Java Application] C:\Program Files\Java\jre1.8.0_171  
Os números de 0 a 10 são:  
0  
1  
2  
3
```

Página 7 de 14 | Palavras: 1.656

MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

0908 - Saved

Page 8 of 8 702 words Portuguese (Brazil)

Focus

100%

Reunião em "General" 20:01

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exibição Design Layout

Fim.

Veja, então, o Java:

```
public class ex02 {  
    public static void main(String[] args) {  
        System.out.println("Os números de 0 a 10 são:  
        for (int i=0; i<10; i++) //laço de repetição for.  
        System.out.println(i); // saída de dados  
    }  
}
```

Este é o resultado:

```
Programa ex02 [Java Application] C:\Program Files\Java\jre1.8.0_171  
Os números de 0 a 10 são:  
0  
1  
2  
3
```

Página 7 de 14 | Palavras: 1.656

MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

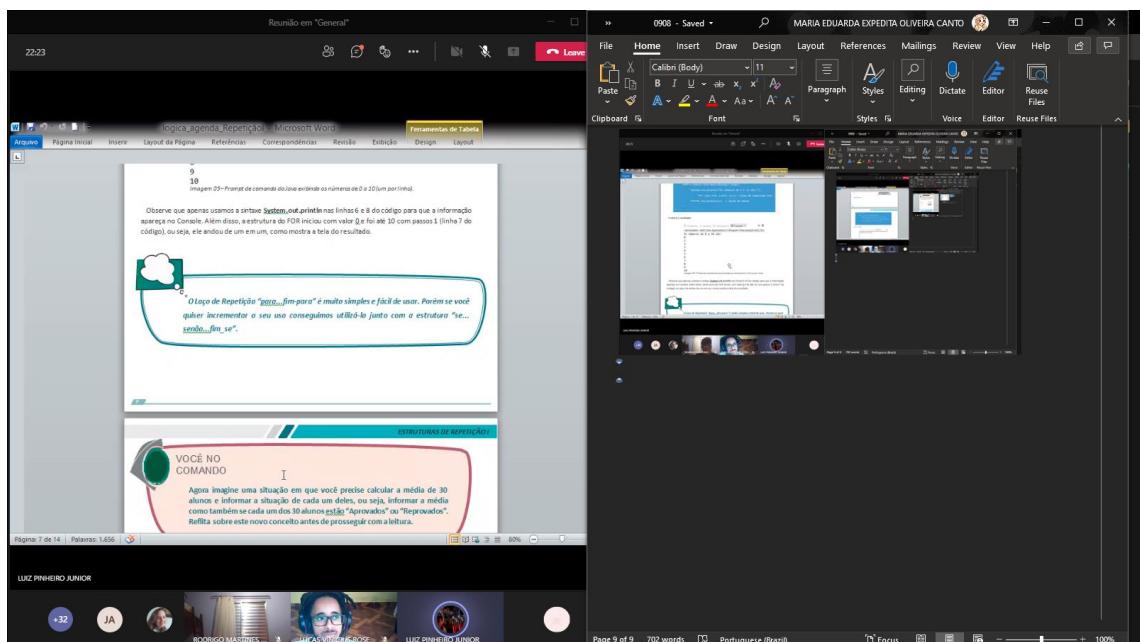
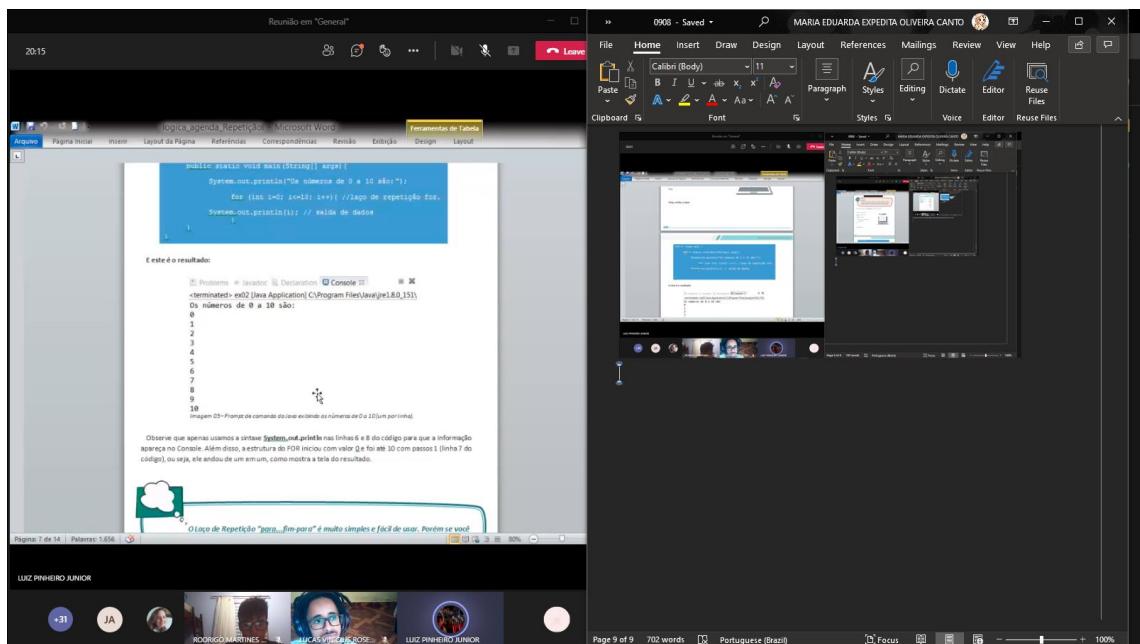
Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

0908 - Saved

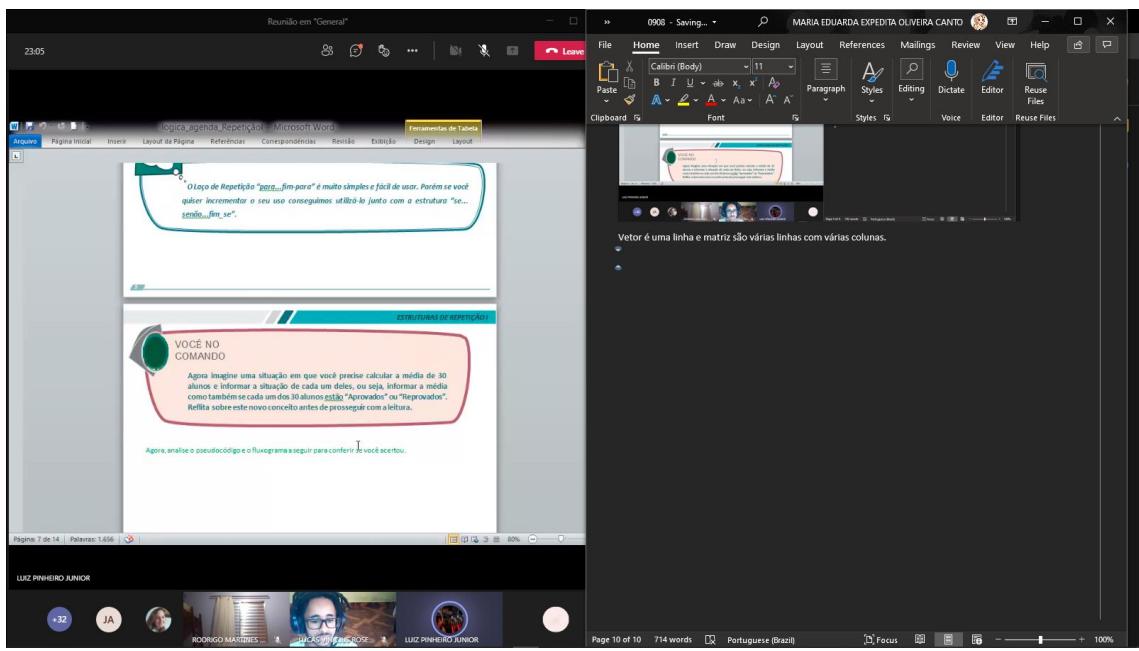
Page 8 of 8 702 words Portuguese (Brazil)

Focus

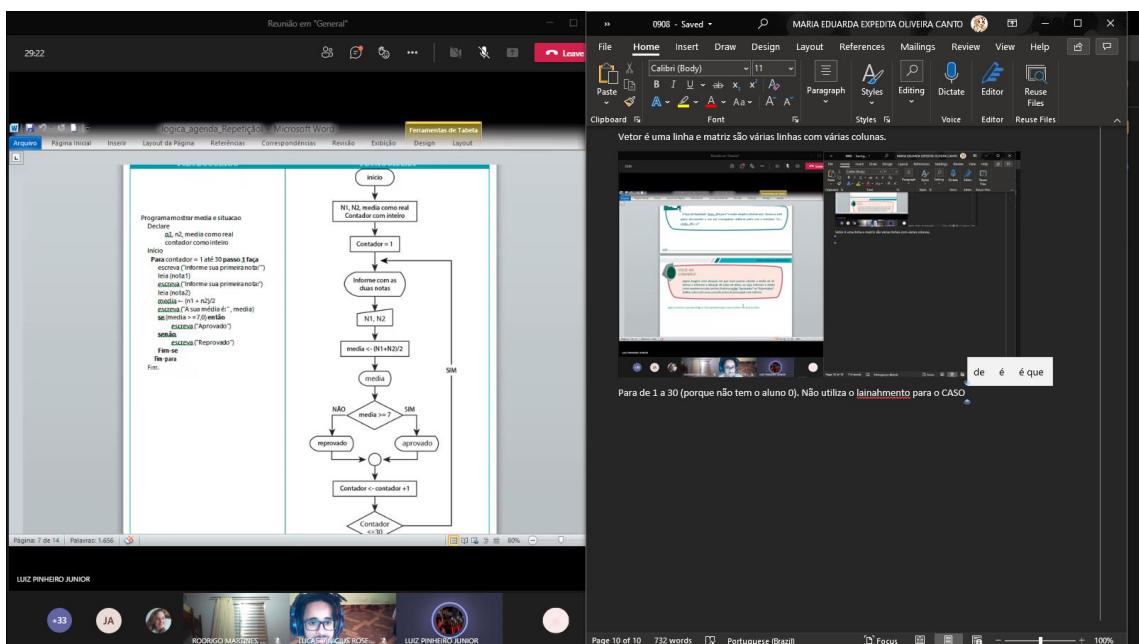
100%



Vetor é uma linha e matriz são várias linhas com várias colunas.



Para de 1 a 30 (porque não tem o aluno 0). Não utiliza o laiaamento para o CASO



O PARA estar lendo a posição que ele encontra (1 execute, 2 execute...) e no CASE por isso não usa porque vai ter que ler 3 posições e 150 perguntas para responder (caso se for no 1, vai, se caso for 2, vai até 50, casose for 3 vai até 50...) logo, usa o SE (se isso, faça isso)

Reunião em "General"

46:49

Lógica agenda Repetição - Microsoft Word

```

    graph TD
        Start([Início com as duas notas]) --> N1N2[N1, N2]
        N1N2 --> Media[media = (N1+N2)/2]
        Media --> Grade{Grade}
        Grade -- NÃO --> Reprovedo[reprovado]
        Grade -- SIM --> Approved[aprovado]
        Reprovedo --> Counter[Contador = contador + 1]
        Counter --> Grade
        Approved --> Counter
        Counter -- NÃO --> Fim([Fim])
        Counter -- SIM --> SM[SM]
    
```

Para contador = 1 até 30 passo 1 faça
escreva ("Informe sua primeira nota")
lea nota1
nota2 = nota1
media = (nota1+nota2)/2
escreva ("A sua média é ", media)
se media >= 7 entao
escreva ("Aprovado")
senão
escreva ("Reprovado")
fim para
fim.

O PARA estar lendo a posição que ele encontra (1 execute, 2 execute...) e no CASE por isso não usa porque vai ter que ler 3 posições e 150 perguntas para responder (caso se for no 1, vai se caso for 2, vai até 50, casose for 3 vai até 50...) logo, usa o SE (se isso, faça isso)

Reunião em "General"

48:05

Lógica_agenda Repetição - Microsoft Word

```

import java.util.Scanner;
public class ex01 {
    public static void main(String[] args) {
        double n1, n2, media;
        n1 = 0;
        n2 = 0;
        // utilizando Scanner como visto nas aulas anteriores.
        Scanner scanner = new Scanner(System.in);
        for (int i=0; i<30; i++) { // loop de repetição para os 30 alunos
            System.out.println("Por favor, informe a primeira nota: ");
            n1 = scanner.nextDouble();
            System.out.println("Por favor, informe a nota e este usuário: ");
            n2 = scanner.nextDouble();
            System.out.println("Digite a segunda nota: " + n2);
            media = (n1 + n2)/2; // cálculo da média.
            if (media >= 7) {
                System.out.println("Sua situação é: Aprovada " + media);
            } else {
                System.out.println("Sua situação é: Reprovada " + media);
            }
        }
    }
}

```

OBSERVE QUE UTILIZAMOS O LAÇO DE REPETIÇÃO PARA (Linha 11) E O LAÇO DE DECISÃO (Linha 18) PARA RESOLVER O PROBLEMA INFORMADO ANTERIORMENTE. Além disso, utilizamos o Scanner (linhas: 1, 10, 13 e 16) para que o usuário pudesse digitar as notas e o programa realizar todo cálculo.

O PARA estar lendo a posição que ele encontra (1 execute, 2 execute...) e no CASE por isso não usa porque vai ter que ler 3 posições e 150 perguntas para responder (caso se for no 1, vai se caso for 2, vai até 50, casose for 3 vai até 50...) logo, usa o SE (se isso, faça isso)

Reunião em "General"

49:41

lógica_agenda_Repetição.java

```
import java.util.Scanner;
public class ex01 {
    public static void main(String[] args) {
        double n1, n2, media;
        n1 = 0;
        n2 = 0;
        // utilizando Scanner com while das agendas anteriores.
        Scanner ler = new Scanner(System.in);
        for (int i=0; i<30; i+=1) //loop de repetição para os 30 alunos
        {
            System.out.println("Entre com a primeira nota: " + i);
            n1 = ler.nextDouble();
            if (n1 <= 100)
                n2 = ler.nextDouble();
            else
                System.out.println("Nota digitada é inválida");
            media = (n1 + n2)/2; // calcula a média.
            if (media >= 60)
                System.out.println("Sua situação é: Aprovada " + media);
            else
                System.out.println("Sua situação é: Reprovada " + media);
        }
    }
}
```

Observe que utilizamos o Laço de Repetição para (linha 11) e o laço de decisão (linha 15) para resolver o problema informado anteriormente. Além disso, utilizamos o Scanner (linhas 1, 10, 13 e 16) para que o usuário pudesse digitar as notas e o programa realizar todo cálculo.

Para entender melhor, assista à videoaula do Prof. Sandro Valerius:

Page 11 of 14 | Palavras: 1.656 | 09:08 - Saved - MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

LUIZ PINHEIRO JUNIOR

+34 LUIZ PINHEIRO JUNIOR LORETA DAUCHAVON CICERO EDUARDO CA...

Reunião em "General"

52:24

lógica_agenda_Repetição.java

```
Teste de Mesa
```

Para que possamos nos certificar de que o algoritmo realizado está correto, antes de passar para a linguagem de programação (no nosso caso, Java) podemos testá-lo simulando valores e verificar se o resultado é o esperado. Esta simulação não é realizada no computador utilizando nenhum software. É realizada no papel.

Calaf! O Teste de Mesa é muito simples de realizar. Basta montar uma pequena tabela e começar a simular os valores utilizando o seu pseudocódigo.

Para exemplificar, vamos voltar no algoritmo anterior (Mostrar os números de 01 até 09 com passo 01) e aplicar o Teste de Mesa.

contador	resultado
0	0
0+1	1
1+1	2
2+1	3
3+1	4
4+1	5
5+1	6
6+1	7
7+1	8
8+1	9

Imagem 07 – Tabela indicando o avanço do Contador de 0 a 9

Para exemplificar, vamos voltar no algoritmo anterior (Mostrar os números de 01 até 09 com passo 01) e aplicar o Teste de Mesa.

LUIZ PINHEIRO JUNIOR

+34 LUIZ PINHEIRO JUNIOR LORETA DAUCHAVON CICERO EDUARDO CA...

Reunião em "General" 09:30

lógica_agenda_Repetição.doc Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição

Parte que podemos nos referir de que o algoritmo está correto, antes de passar para a linguagem de programação (no nosso caso, Java) podemos testá-lo simulando valores e verificar se o resultado é o esperado. Esta simulação não é realizada no computador utilizando software. É feita no papel.

Calcular O Teste de Mesa é muito simples de realizar. Basta montar uma pequena tabela e começar a simular os valores utilizando o seu pseudocódigo.

Para exemplificar, vamos voltar no algoritmo anterior (Mostrar os números de 01 até 09 com passo 01) e aplicar o Teste de Mesa.

contador	resultado
0	0
0+1	1
1+1	2
2+1	3
3+1	4
4+1	5
5+1	6
6+1	7
7+1	8
8+1	9

Vejam que você iniciou com zero e como é exercício pede o incremento, ou seja, precisa somar 1 ao resultado anterior.

E como foi dito anteriormente, como o valor final é 10, então temos até o valor 9.

Imagem 07 - Tabela inédita e exange do Contador de 0 a 9.

Para exemplificar, vamos voltar no algoritmo anterior (Mostrar os números de 01 até 09 com passo 01) e aplicar o Teste de Mesa.

Page 12 of 13 | Palavras: 1,650 | 100%

LUIZ PINHEIRO JUNIOR

+33 LUIZ PINHEIRO JUNIOR LOU DALCHAVON CICERO EDUARDO CA...

Maria Eduarda Expedita Oliveira Canto 0908 - Saved

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Voice Editor Reuse Files

Page 13 of 13 791 words Portuguese (Brazil)

Reunião em "General" 05:34

lógica_agenda_Repetição.doc Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição Ferramentas de Tabelas Design Layout

VOCÊ NO COMANDO

Elabore o Algoritmo, o Fluxograma e um Programa em Java que some todos os números no intervalo de 0 até 100.

Agora confira se você conseguiu resolver o desafio:

1. **Passo a Passo**

Passo a Passo	Fluxograma
<pre>Programa ex: Declare num, soma como int Início soma = 0 Escreva("Entre com um número") [para i = 1 ate 100 pase 1 faça] soma = soma + num Escreva(soma) fim Escreva(soma)</pre>	<pre> Início Leitura num Soma = 0 Repetir Soma = Soma + num num = num + 1 Fim Saída Soma </pre>

Imagem 08 – Resolução do problema em Passo a Passo e Fluxograma.

Veja no Java:

```
public class ex01 {
    public static void main (String[] args) {
```

Page 13 of 13 | Palavras: 171,656 | 100%

LUIZ PINHEIRO JUNIOR

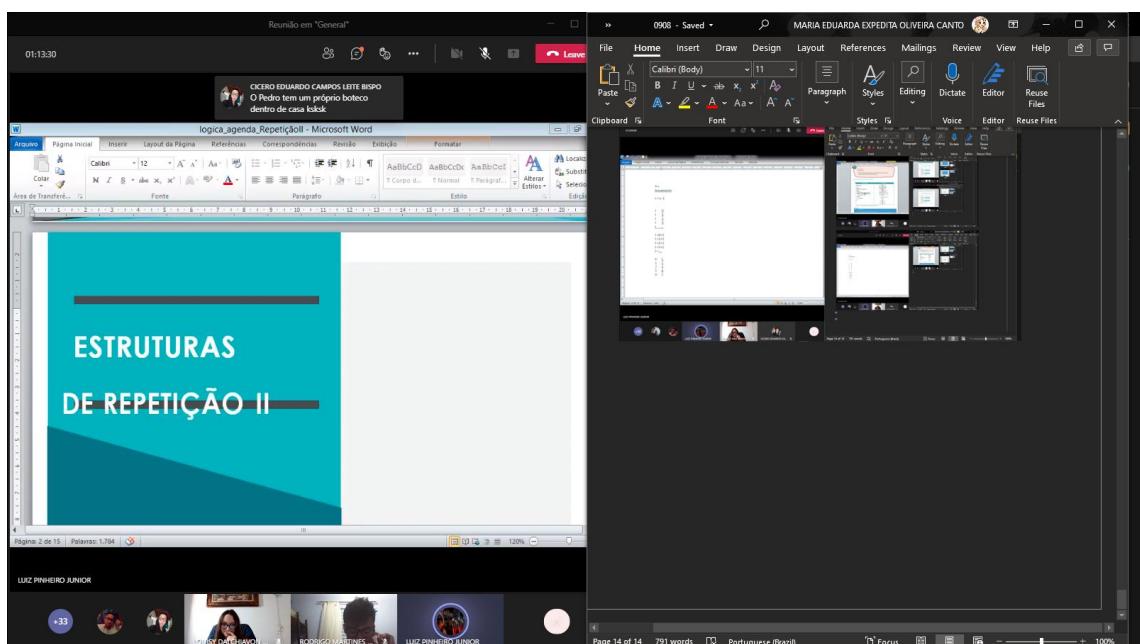
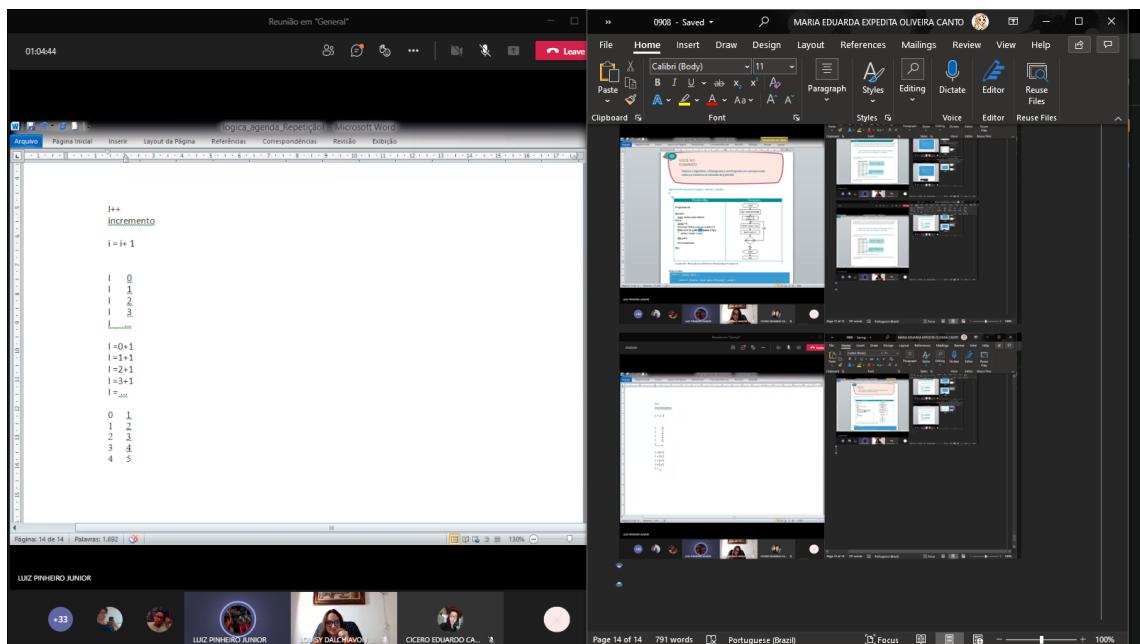
+33 LUIZ PINHEIRO JUNIOR LOU DALCHAVON CICERO EDUARDO CA...

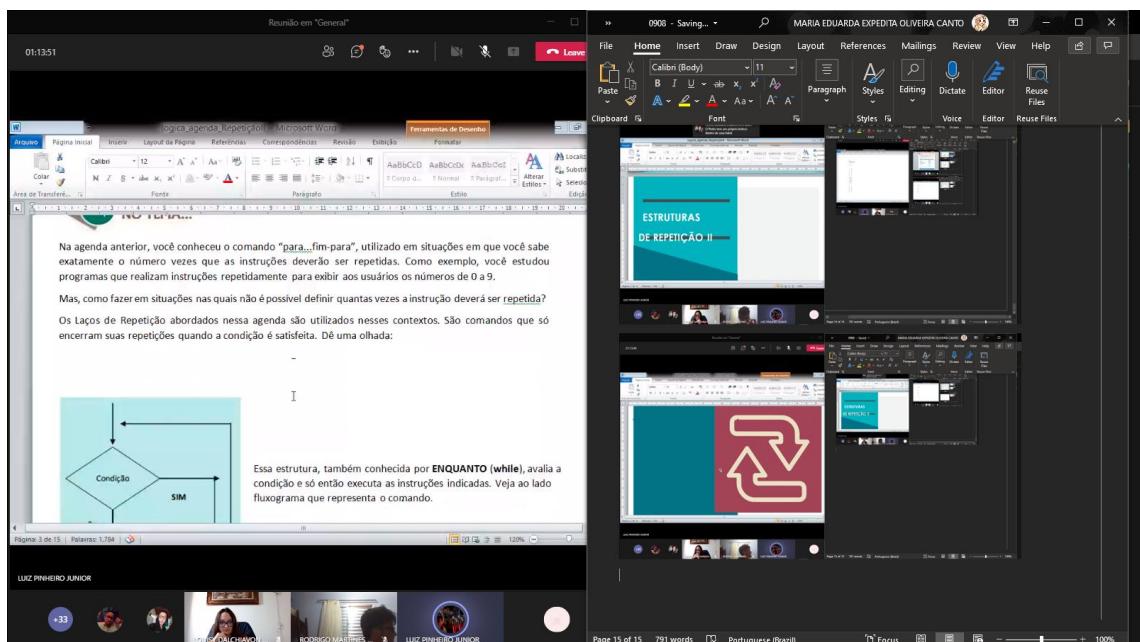
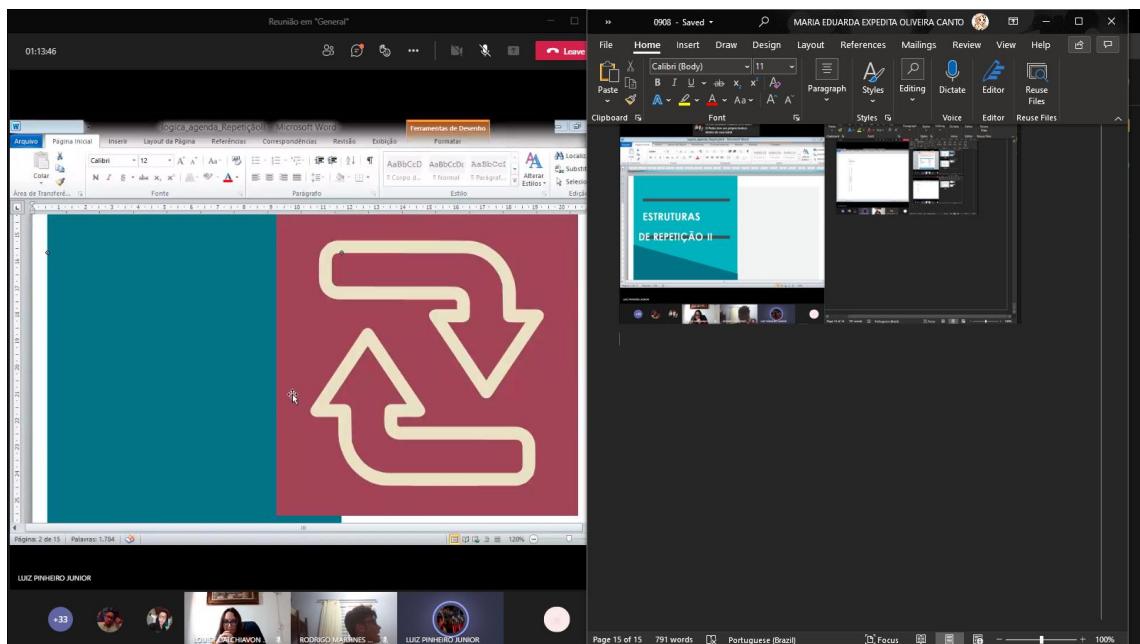
Maria Eduarda Expedita Oliveira Canto 0908 - Saved

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Voice Editor Reuse Files

Page 13 of 13 791 words Portuguese (Brazil)





Reunião em "General"

01:14:10

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição Fórmulas

Na agenda anterior, você conheceu o comando "para...fim-par", utilizado em situações em que você sabe exatamente o número de vezes que as instruções deverão ser repetidas. Como exemplo, você estudou programas que realizam instruções repetidamente para exibir aos usuários os números de 0 a 9.

Mas, como fazer em situações nas quais não é possível definir quantas vezes a instrução deverá ser repetida?

Os laços de Repetição abordados nessa agenda são utilizados nesses contextos. São comandos que só encerram suas repetições quando a condição é satisfeita. Dê uma olhada:

Imagem 01: Representação básica de um fluxograma de estrutura de Repetição Enquanto (while)

LUIZ PINHEIRO JUNIOR

0908 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

0908 - Saved... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Page 16 of 16 791 words Portuguese (Brazil) Focus

Essa condição tem que ter bem claro que as linhas de comando (bloco de execução) do ENQUANTO vai ser maior que o PARA e que o SE, a gente não sabe a quantidade que vai passar por ali. Enquanto A CONDIÇÃO FOR VERDADEIRA ELE EXECUTA, SE NÃO FOR VERDADEIRA, ELA É FALSA.

Reunião em "General"

01:20:06

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição Fórmulas

Agora o "REPITA... ATÉ QUE" conhecido também como `do...While`. Esta estrutura é muito parecida com a primeira e por conta disso elistem vários Programadores que não o usam.

Esta estrutura é utilizada quando, também, não conhecemos o número de repetições que ocorrerá e só encerra, também, quando uma condição é satisfeita. A diferença desta para a primeira é que a condição é testada por último fazendo com que ao menos uma vez o trecho seja executado.

Imagem 01: Representação básica de um fluxograma de estrutura de Repetição Enquanto (while)

Imagem 02: Representação básica de um fluxograma de estrutura de Repetição Repita...até que (do while)

LUIZ PINHEIRO JUNIOR

0908 - Saved... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

0908 - Saved... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Editing Dictate Editor Reuse Files

Page 16 of 16 843 words Portuguese (Brazil) Focus

No `do while` o bloco de comando vem antes e no `while` vem depois. O bloco = instruções.

1608

TPA

Reunião em "General" 1608 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Paste Calibri (Body) 11 Font Paragraph Styles Editing Dictate Editor Reuse Files

Clipboard References Correspondences Review Exit

Arquivo Página Inicial Insere Layout da Página Referências Correspondências Revisão Exclusão

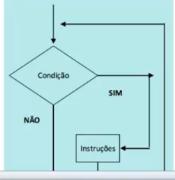
03:09

logica_agenda_Repetição.docx Microsoft Word

Na agenda anterior, você conheceu o comando "para...fim-para", utilizado em situações em que você sabe exatamente o número vezes que as instruções deverão ser repetidas. Como exemplo, você estudou programas que realizam instruções repetidamente para exibir aos usuários os números de 0 a 9.

Mas, como fazer em situações nas quais não é possível definir quantas vezes a instrução deverá ser repetida?

Os Laços de Repetição abordados nessa agenda são utilizados nesses contextos. São comandos que só encerram suas repetições quando a condição é satisfeita. Dê uma olhada:

Fluxograma de um laço while. Ele mostra um retângulo rotulado "Condição" com uma seta apontando para cima. A seta sai da direita do retângulo e vai para o lado direito, rotulada "SIM". A seta sai da esquerda do retângulo e vai para baixo, rotulada "NÃO". Da seta "SIM", uma seta aponta para cima para dentro de um retângulo rotulado "Instruções".

Essa estrutura, também conhecida por ENQUANTO (while), avalia a condição e só então executa as instruções indicadas. Veja ao lado fluxograma que representa o comando.

Page 3 of 15 | Palavras: 1,754 120% LUIZ PINHEIRO JUNIOR +22 CICERO EDUARDO CA... LOURIS DALCHAVON LUIZ PINHEIRO JUNIOR Page 2 of 2 206 words Focus 90%

Reunião em "General" 1608 - Saved MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Paste Calibri (Body) 11 Font Paragraph Styles Editing Dictate Editor Reuse Files

Clipboard References Correspondences Review Exit

Arquivo Página Inicial Insere Layout da Página Referências Correspondências Revisão Exclusão

04:45

logica_agenda_Repetição.docx Microsoft Word

Os laços de repetição abordados nessa agenda são utilizados nesses contextos. São comandos que só encerram suas repetições quando a condição é satisfeita. Dê uma olhada:

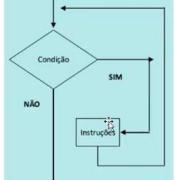
Fluxograma de um laço while. Ele mostra um retângulo rotulado "Condição" com uma seta apontando para cima. A seta sai da direita do retângulo e vai para o lado direito, rotulada "SIM". A seta sai da esquerda do retângulo e vai para baixo, rotulada "NÃO". Da seta "SIM", uma seta aponta para cima para dentro de um retângulo rotulado "Instruções".

Imagem 01:Representação básica de um fluxograma de Estrutura de Repetição Enquanto (while).

Page 3 of 15 | Palavras: 1,754 120% LUIZ PINHEIRO JUNIOR +25 CICERO EDUARDO CA... LOURIS DALCHAVON LUIZ PINHEIRO JUNIOR Page 2 of 2 206 words Focus 90%

Reunião em "General" 04:48

logica_agenda_Repetição.docx

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exclusão

Imagem 01: Representação básica de um fluxograma de Estrutura de Repetição Enquanto (while).

Imagem 02: Representação básica de um fluxograma

Agora o "REPITA... ATÉ QUE" conhecida também como **do...While**. Esta estrutura é muito parecida com a primeira e por conta disso existem vários Programadores que não a usam.

Esta estrutura é utilizada quando, também, não conhecemos o número de repetições que ocorrerá e só encerra, também, quando uma condição é satisfeita. A diferença desta para a primeira é que a condição é testada por último fazendo com que **ao menos uma vez** o trecho seja executado.

Page 3 of 3 | Palavras: 1,794 | 16:08 - Saving... | MARIA EDUARDA EXPEDITA OLIVEIRA CANTO | Focus | 90%

Reunião em "General" 06:55

logica_agenda_Repetição.docx

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Exclusão

"ENQUANTO... FIM ENQUANTO"

Este laço de repetição, como informado anteriormente, trabalha enquanto a condição é verdadeira e vai executando as instruções. Portanto, a condição sendo falsa, ele sai do laço e vai para o próximo comando no programação.

PSEUDOCÓDIGO	FLUXOGRAFO	JAVA
Enquanto <condição> faça <Comando> <Comando> FimEnquanto		<pre>while (<condição>){ <instruções> }</pre>

Agora que você conheceu a definição e a estrutura do laço, veja o exemplo a seguir.

Elabore um Algoritmo, um Fluxograma e um Programa em Java que mostre todos os números menores que 10.

Page 4 of 4 | Palavras: 206 words | 16:08 - Saving... | MARIA EDUARDA EXPEDITA OLIVEIRA CANTO | Focus | 90%

Reunião em "General" 13:19

logica_agenda_Repetição - Microsoft Word

ESTRUTURA DE REPETIÇÃO

PSEUDOCÓDIGO

```

Programa
    Declarar
        num como inteiro
        inicio
        Num = 0
        enquanto (Num <= 10) faça
            escreva (Num)
            num +> 1
        fim- enquanto
    fim

```

FLUXOGRAMA

Agora veja em Java:

```

1- public class ex01 {
2-     3-     public static void main(String[] args) {
4-         5-         int num;
5-         6-         while (num <= 10) {
6-             7-             System.out.println("Número: " + num);
7-             8-             num++;
8-         }
9-     }
9- }

```

Page 5 of 5 | Palavras: 1/1 704

MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Voice Editor Reuse Files

Page 4 of 5 206 words Focus 90%

LUIZ PINHEIRO JUNIOR +29 CICERO EDUARDO CA... LOURY DALCHAVON LUIZ PINHEIRO JUNIOR

Reunião em "General" 13:38

logica_agenda_Repetição - Microsoft Word

ESTRUTURA DE REPETIÇÃO

FLUXOGRAMA

Agora veja em Java:

```

1- public class ex01 {
2-     3-     public static void main(String[] args) {
4-         5-         int num;
5-         6-         while (num <= 10) {
6-             7-             System.out.println("Número: " + num);
7-             8-             num++;
8-         }
9-     }
9- }

```

Observar que na linha 5 temos a estrutura de repetição (enquanto (while)) te não como condição ação que o exercício calculou (números menores que 10). As linhas 5 e 7 são exatamente os comandos que são executados dentro da estrutura.

Veja o resultado apresentado pelo computador:

```

Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
Número: 6
Número: 7
Número: 8
Número: 9
Número: 10

```

Page 5 of 5 | Palavras: 1/1 704

MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles Voice Editor Reuse Files

Page 5 of 5 206 words Focus 90%

LUIZ PINHEIRO JUNIOR +29 CICERO EDUARDO CA... LOURY DALCHAVON LUIZ PINHEIRO JUNIOR

Reunião em "General" 15:28

ITALO PEDRO LOPES SILVA
Obrigado

logica_agenda_Repetição - Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição Fazendo de Desenvolvedor

1608 - Saved MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Font Paragraph Styles Voice Editor Reuse Files

Observe que na lista de impressão a estrutura de repetição Enquanto (While) temos como condição a ação que o exercicio consegue imprimir numeros maiores que 10. As linhas 6 e 7 são exatamente os comandos que são executados dentro da estrutura.

Veja o Resultado apresentado pelo computador:

```
Número 0
Número 1
Número 2
Número 3
Número 4
Número 5
Número 6
Número 7
Número 8
Número 9
```

Imagem 07: Representação do resultado do Programa em Java que mostra todos os números menores que 10 (Número 0, Número 1, Número 2, Número 3, Número 4, Número 5, Número 6, Número 7, Número 8, Número 9).

Page 5 of 5 | Palavras: 5/1704 | Page 5 of 5 | 206 words | Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

+30 CICERO EDUARDO CA... LOURISY DALCHAVON LUIZ PINHEIRO JUNIOR

Reunião em "General" 15:47

ITALO PEDRO LOPES SILVA
Obrigado

logica_agenda_Repetição - Microsoft Word

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Edição Fazendo de Desenvolvedor

1608 - Saved MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Font Paragraph Styles Voice Editor Reuse Files

Observe que na lista de impressão a estrutura de repetição Enquanto (While) temos como condição a ação que o exercicio consegue imprimir numeros maiores que 10 (Número 0, Número 1, Número 2, Número 3, Número 4, Número 5, Número 6, Número 7, Número 8, Número 9).

Imagem 07: Representação do resultado do Programa em Java que mostra todos os números menores que 12 (Número 0, Número 1, Número 2, Número 3, Número 4, Número 5, Número 6, Número 7, Número 8, Número 9).

Dica! A condição colocada acima foi num < 10 e por isso você deve ter percebido no resultado que os números que saíram foram de 0 até o 9.

O Laço de repetição Enquanto...Fim-Enquanto não possui incremento automático como o laço Para...Fim. Para.

Fique atento a isso!

Exemplo a seguir apresenta uma repetição infinita e demonstra as implicações da ausência de contador. Veja:

```
I
```

Page 5 of 5 | Palavras: 5/1704 | Page 5 of 5 | 206 words | Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

+30 CICERO EDUARDO CA... LOURISY DALCHAVON LUIZ PINHEIRO JUNIOR

16:00



logica_agenda_RepetiçãoII - Microsoft Word

FERRAMENTAS DE DESENHO

PSEUDOCÓDIGO

```
Programa ex
    Declare
        num como inteiro
    Início
        Num <- 0
        Enquanto (num < 10) faça
            escreva num
        fim-enquanto
```

FLUXOGRAMA

```
graph TD
    Inicio([Início]) --> NumInt[Num como inteiro]
    NumInt --> Cond{Num < 0}
    Cond --> Sim{Num < 10}
    Sim -- Sim --> Numoval([num])
    Numoval --> Fim([fim])
    Sim -- Não --> Cond
```

Veja como fica em Java:

Página: 7 de 15 Palavras: 5/1.784

80%

LUIZ PINHEIRO JUNIOR

+30



CICERO EDUARDO CA...



LOUISY DALCHAVON ...



LUIZ PINHEIRO JUNIOR

Reunião em "General" 16:13

lógica_agenda_Repetição.docx - Microsoft Word

Ferramentas do Desenvolvedor

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Execução Formatar

Clipboard

Fonte

Paragraph

Styles

Editor

Voice

Reuse Files

```

    graph TD
        A{Num <10} -- Sim --> B[num]
        B --> C[fim]
    
```

```

    public class ExemploOperadores {
        public static void main(String[] args) {
            int num = 0;
            while (num < 10) {
                System.out.println("Número = " + num);
                num++;
            }
        }
    }
    
```

Nota: que o programa é idêntico ao anterior com a exceção de que não temos o incremento do contador num (num = num+1). Com isso, o resultado ficará exatamente como o valor de 0 (zero) e o programa nunca terá um fim.

Página 7 de 15 | Palavras: 517,784

LUIZ PINHEIRO JUNIOR

+30 CICERO EDUARDO CA... LOURIS DALCHAVON LUIZ PINHEIRO JUNIOR

Page 6 of 6 206 words Focus

Reunião em "General" 16:17

lógica_agenda_Repetição.docx - Microsoft Word

Ferramentas do Desenvolvedor

Arquivo Página Inicial Inserir Layout da Página Referências Correspondências Revisão Execução Formatar

Clipboard

Fonte

Paragraph

Styles

Editor

Voice

Reuse Files

```

    graph TD
        A{Num <= 10} -- Sim --> B[Número]
        B --> C[fim]
    
```

```

    public class ExemploOperadores {
        public static void main(String[] args) {
            int num = 0;
            while (num <= 10) {
                System.out.println("Número = " + num);
                num++;
            }
        }
    }
    
```

A diferença deste código no Java com o anterior é apenas o operador igual (=). Ao colocar a condição <= 10 você é incluído o 10 na condição e o resultado (mostrando o 10) aparecerão os números de 0 até 10.

Dica: este assunto já foi abordado na agenda anterior!

Contra com a codificação a seguir:

```

    public class Exemplo {
        public static void main(String[] args) {
            int num = 0;
            while (num <= 10) {
                System.out.println("Número = " + num);
                num++;
            }
        }
    }
    
```

Página 8 de 15 | Palavras: 517,784

LUIZ PINHEIRO JUNIOR

+30 CICERO EDUARDO CA... LOURIS DALCHAVON LUIZ PINHEIRO JUNIOR

Page 6 of 6 206 words Focus

Reunião em "General" 16:25

logica_agenda_Repetição.docx Microsoft Word

FLUXOGRAMA

```

    graph TD
        Inicio([Início]) --> NumInt[Num como inteiro]
        NumInt --> NumZero[Num = 0]
        NumZero --> Num[Num]
        Num --> NumMais[Num <- num + 1]
        NumMais --> Cond{Num > 10}
        Cond -- NÃO --> Fim([fim])
        Cond -- SIM --> NumZero
    
```

JAVA

```

for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
    
```

Como você já sabe, este estrutura é parecida com a while, que você acabou de conhecer. A única diferença é que a condição dela é executada por último. Vamos aos detalhes para entender melhor?

Agora que você conheceu um pouco mais este estrutura, vamos voltar ao exemplo anterior.

Número 0	Número 9
Número 1	Número 8
Número 2	Número 7
Número 3	Número 6
Número 4	Número 5
Número 5	Número 4
Número 6	Número 3
Número 7	Número 2
Número 8	Número 1
Número 9	Número 0

Page 9 of 15 | Palavras: 5/1.704

LUIZ PINHEIRO JUNIOR

+30 CICERO EDUARDO CA... LOURY DALCHAVON LUIZ PINHEIRO JUNIOR

Page 7 of 7 206 words Focus Reuse files

Reunião em "General" 19:47

logica_agenda_Repetição.docx Microsoft Word

FLUXOGRAMA

```

    graph TD
        Inicio([Início]) --> NumInt[Num como inteiro]
        NumInt --> NumZero[Num = 0]
        NumZero --> Num[Num]
        Num --> NumMais[Num <- num + 1]
        NumMais --> Cond{Num > 10}
        Cond -- NÃO --> Fim([fim])
        Cond -- SIM --> NumZero
    
```

JAVA

```

for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
    
```

Observe como fica em Java:

Page 9 of 15 | Palavras: 5/1.704

LUIZ PINHEIRO JUNIOR

+30 CICERO EDUARDO CA... LOURY DALCHAVON LUIZ PINHEIRO JUNIOR

Page 8 of 8 206 words Focus Reuse files

Observe como fica em Java:

```
1: public class ex01 {
2:     public static void main(String[] args) {
3:         int num;
4:         do {
5:             System.out.println("Número: " + num);
6:             num++;
7:         } while (num < 10);
8:     }
9: }
```

Observe que, como o programa responde ao mesmo exercício, o resultado é o mesmo. Portanto, o professor pode inserir dentro de um laço de repetição. Note que as linhas 6 até 10 exibem sintaxe do comando `repita`.

Veja na videoaula a seguir, que o professor Sandro Vilela nos retorna os conteúdos estudados até o momento.

1 - Pede-se em pseudocódigo, conhecendo dois valores inteiros de A e B, maiores do que zero como obter o M.D.C(A, B).

2 - Pede-se em pseudocódigo, uma solução, Dado o valor N inteiro maior 1, como obter o valor da soma S descrita abaixo:

Reunião em "General"

59:03

File Home Insert Design Layout References Mailings Review View Help

Paste Font Paragraph Styles Editing Dictate Editor Reuse Files

Voice

Pede-se em pseudocódigo, conhecendo dois valores inteiros de A e B, maiores do que zero como obter o M.D.C.(A, B).

Pede-se em pseudocódigo, uma solução, Dado o valor N inteiro maior 1, como obter o valor da soma S descrita abaixo:

$$S = \frac{2}{2n+1} + \frac{3}{2n+1} + \dots + \frac{n}{2n+1}$$

$$= \frac{1}{2} \left(\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n} \right)$$

$$= \frac{1}{2} H_n$$

Exemplos:

1) Dados inteiros 12 e 18

$$\begin{array}{r} 12 \\ 18 \\ \hline 6 \end{array}$$

MDC = 6

Restos = 0

2) Dados inteiros 12 e 18

$$\begin{array}{r} 12 \\ 18 \\ \hline 6 \\ 12 \\ 12 \\ \hline 0 \end{array}$$

MDC = 6

Restos = 0

LUIZ PINHEIRO JUNIOR

+36

HB

CICERO EDUARDO CA...

LUIZ PINHEIRO JUNIOR

Page 10 of 10 | 247 words | Focus

1 - Maxdivi()

Var

A, B, R, DIV, DID: inteiro

Leia

A, B;

DID = A;

DIV = B;

R = DID mod DIV

Enquanto (R > 0) faça

DID = DIV;

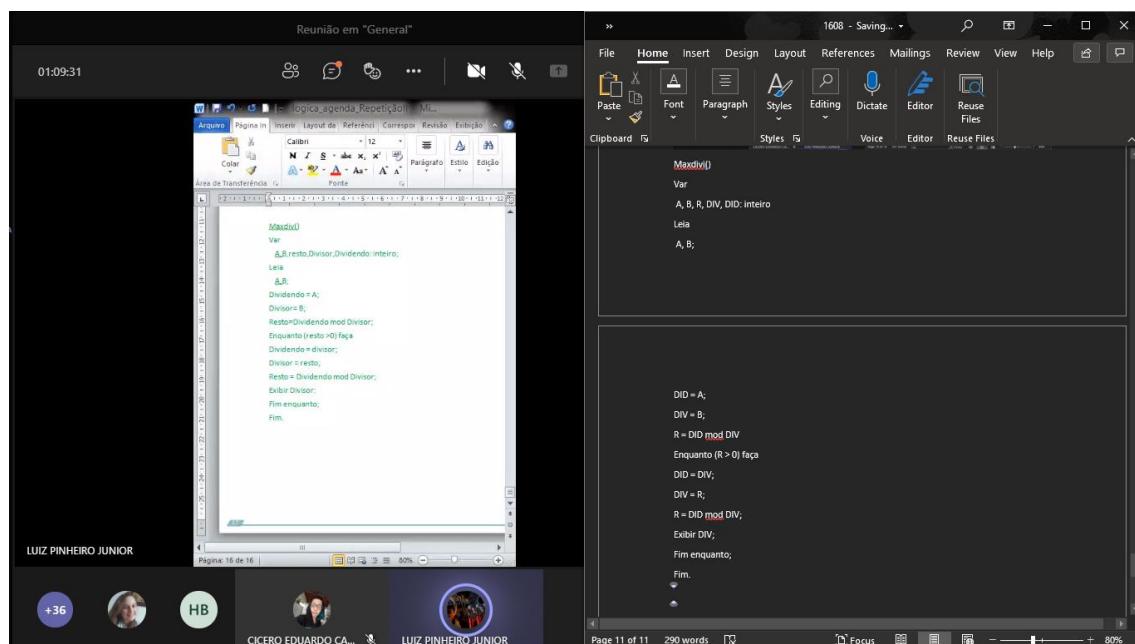
DIV = R;

R = DID mod DIV;

Exibir DIV;

Fim enquanto;

Fim.



Reunião em "General"

01:29:21

GABRIEL TAYLOR SOUZA SILVA
vou assinar prof

Modulo1

```

Var
    A;resto,Divisor,Dividendo inteiro;
Leia
    A;
Dividendo = A;
Divisor = B;
Resto=Dividendo mod Divisor;
Enquanto (resto >0)faça
    Dividendo = divisor;
    divisor = resto;
    Resto = Dividendo mod divisor;
    Escrever divisor;
    Fim enquanto;
Fim.

```

2) Pode-se em pseudocódigo, uma solução. Dado o valor N inteiro maior 1, como obter o valor da soma S descrita abaixo:

$$S = \frac{2}{1} + \frac{3}{2} + \frac{5}{3} + \dots + \frac{n}{n-1}$$

3) $(2n-3)(2n+1)$

LUIZ PINHEIRO JUNIOR

1608 - Saved

File Home Insert Design Layout References Mailings Review View Help

Clipboard Styles Editor Reuse Files

DIV = DIV;
R = DID mod DIV;
Exibir DIV;
Fim enquanto;
Fim.

Page 11 of 11 296 words Focus

2 -

Reunião em "General"

01:31:32

BEATRIZ DAMAS
aah tendii

2) Pode-se em pseudocódigo, uma solução. Dado o valor N inteiro maior 1, como obter o valor da soma S descrita abaixo:

$$S = \frac{2}{1} + \frac{3}{2} + \frac{5}{3} + \dots + \frac{n}{n-1}$$

3) $(2n-3)(2n+1)$

2)

ESTRUTURAS DE REPETIÇÃO

Pri
Seg
S= pri/seg

LUIZ PINHEIRO JUNIOR

1608 - Last Modified: Just now

File Home Insert Design Layout References Mailings Review View Help

Clipboard Styles Editor Reuse Files

Page 12 of 12 296 words Focus

2308

TPA

Reunião em "General" 13:23

Microsoft Word - lógica_agenda_Repeticão(1)

File Home Insert Draw Design Layout References Mailings Review View Help

Página inicial Insert Layout de Página Referências Correspondências Revião Edição

1) Pede-se em pseudocódigo, conhecendo dois valores inteiros de A e B, maiores do que zero como obter o M.D.C.(A,B).

Resto Inteiros

1) 0 1 2

12	18	12	6	MDC = 6
12	6	0		resto = 0

Restos resto = 0

Maxdiv()

Var

Page 4 of 4 | 490 words | Portuguese (Brazil) | Focus | + 120%

LUZ PINHEIRO JUNIOR

CICERO EDUARDO CA... LOUISY DALCHIAVON ... LUZ PINHEIRO JUNIOR

Reunião em "General" 20:00

Microsoft Word - lógica_agenda_Repeticão(1)

File Home Insert Draw Design Layout References Mailings Review View Help

Página inicial Insert Layout de Página Referências Correspondências Revião Edição

Maxdiv()

Var

```

    A,B,resto,Divisor,Dividendo: inteiro;
Leia
    A,B;
Dividendo = A;
Divisor = B;
Resto=Dividendo mod Divisor;
Enquanto (resto >0) faça
    Dividendo = divisor;
    Divisor = resto;
    Resto = Dividendo mod Divisor;
Exibir Divisor;
Fim enquanto;
Fim.

```

Page 4 of 4 | 491 words | Portuguese (Brazil) | Focus | + 120%

LUZ PINHEIRO JUNIOR

CICERO EDUARDO CA... LOUISY DALCHIAVON ... LUZ PINHEIRO JUNIOR

Reunião em "General"

52:58

File Home Insert Draw Design Layout References Mailings Review View Help

Página inicial Inserir Layout da Página Referências Correspondentes Revisão Edição

Clipboard Font Paragraph Styles Editor

Editor Reuse Files

Diagrama de fluxograma para ENQUANTO (while) loop:

```

graph TD
    Condicao{Condição} -- NÃO --> Fim
    Condicao -- SIM --> Instrucoes[Instruções]
    Instrucoes --> Condicao
    Instrucoes --> Incremento[Incremento o contador]
    Incremento --> Condicao
    Condicao -- NÃO --> Fim
    Condicao -- SIM --> Instrucoes
  
```

Imagem 03 - Representação básica de um fluxograma de estrutura de Repetição Enquanto (while).

O azul é do ENQUANTO e o branco é do PARA.

O azul é do ENQUANTO e o branco é do PARA e o azul esquerdo é o REPITE.

Reunião em "General" 01:07:02

Ferramentas de Desenvolvimento

Diagramas de Fluxo

Imagem 01: Representação básica de um fluxograma de Repetição Enquanto.

```

graph TD
    Condicao{Condição} -- SIM --> Instrucoes[Instruções]
    Instrucoes --> Condicao
    Condicao -- NÃO --> Fim
  
```

Imagem 02: Representação básica de um fluxograma de Estrutura de Repetição While.

```

graph TD
    Inicializacao[Inicialização de contador] --> Condicao{Condição}
    Condicao -- SIM --> Instrucoes[Instruções]
    Instrucoes --> Incremento[Incremento do contador]
    Incremento --> Condicao
    Condicao -- NÃO --> Fim
  
```

Imagem 03: Representação básica de um fluxograma de Estrutura de Repetição While.

```

graph TD
    Inicializacao[Inicialização de contador] --> Condicao{Condição}
    Condicao -- SIM --> Instrucoes[Instruções]
    Instrucoes --> Condicao
    Condicao -- NÃO --> Fim
  
```

Agora o "REPITA... ATÉ QUE" coincide também com a primeira e por conta disso existem vários Programadores que não o usam.

Esta estrutura é utilizada quando, também, não conhecemos o número de repetições que ocorrerá e só encerra, também, quando uma condição é satisfeita. A diferença desta para a primeira é que a condição é testada por último fazendo com que **ao menos uma vez o trecho seja executado**.

O azul é do ENQUANTO e o branco é do PARA e o azul esquerdo é o REPITE.

LUZ PINHEIRO JUNIOR

Page 3 of 17 Palavras: 1,345

2308 - Saving... MARIA EDUARDA EXPEDITA OLIVEIRA CANTO

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Paragraph Styles

Reuse Files

Image showing a video call interface with participants Cicero Eduardo Campos Leite Bispo, Louisy Dalchavon, and Luz Pinheiro Junior. The video shows Cicero walking away from the camera.

Todas são estruturas

3008

Meeting 19:34

CICERO EDUARDO CAMPOS LEITE BISPO
Professor, n seria mais fácil colocar a media dentro do soma ? Ficando
Média = soma + qd / 30 ?

Mediadiaria()

Var

Dia,soma,qd,media : inteira;

Soma = 0;

Para dia de 1 até 30 faça

 Entra qd;

 Soma = soma+qd;

 Media = soma/30;

Fim-Para;

Exibir media;

Fim.

Página 1 de 1 1 de 22 palavras

File Home Insert Design Layout Refer Mailings Review View Help

Document1 - Word

Page 1 of 1 0 words

Exercício 3:

Meeting

23:15

Leave

Document1 - Word

Arquivo Página Inserir Desenho Layout Referência Mailin Review View Ajuda

```

Mediadiaria()
Var
Dia,soma,qd,media : inteira;
Soma = 0;
Para dia de 1 até 30 faça
... enquanto (dia<=30) faça
    Entra qd;
    Soma = soma+qd;
    Media = soma/30;
Fim-Para;
... Fim-Enquanto;
Exibir media;
Fim.

```

Página 1 de 1 32 palavras 140%

Exercício 3:

Page 1 of 1 2 words Focus

com o enquanto: (amarelo é alternativa)

Meeting

01:13:39

Leave

Document2 - Word

Arquivo Página Inserir Desenho Layout Referência Mailin Review View Ajuda

```

Diamaior()
Var
Qmax,dia,qde,dmax: inteiro;
Inicio
Dia= 0;
Para dia 1 até 30 faça
    Entra qd;
    Se (qd>qmax) então
        Qmax = qd;
        Dmax = dia;
    Exibir dmax;
    Fim-se;
    Fim-Para;
Fim.

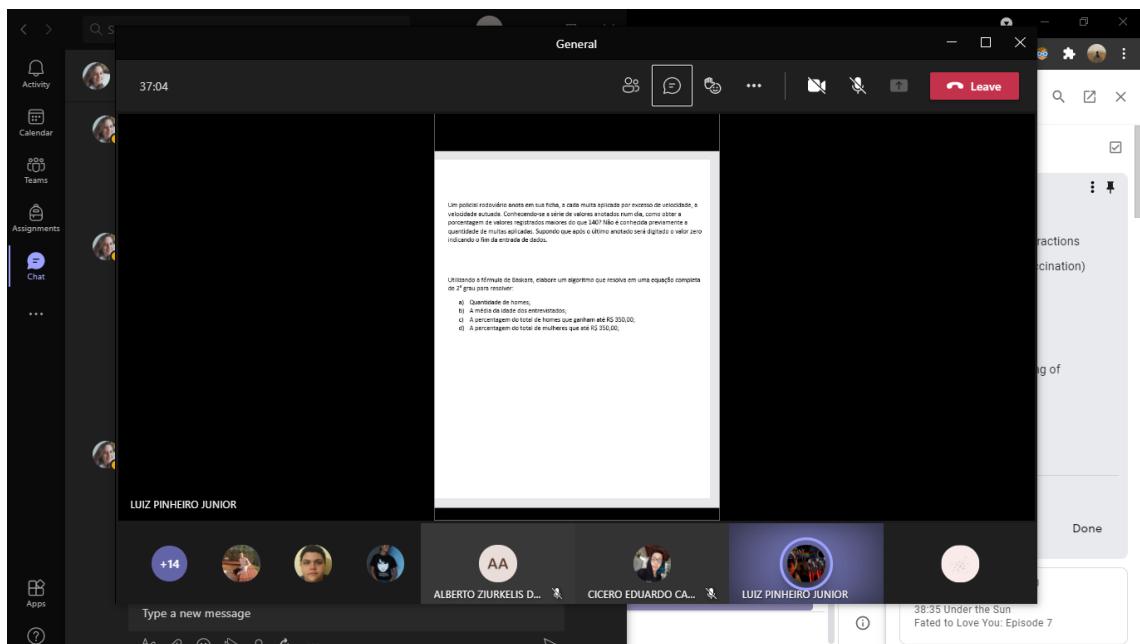
```

Página 1 de 1 30 palavras 140%

Exercício 4:

Page 2 of 2 10 words Focus

Exercício 4:



0609

1309

2009

Reunião em "General"

26:57

LOUISY DAL... LUIZ PINHEIRO ... CICERO ED... RODRIGO ... +33

Microsoft Word

Arquivo Página Inserir Layout Referência Correspondência Revisão Exibição

```

Início
soma = 0;
Escreva _" A soma dos 100 primeiros números é"
Para i de 1 até 100 faça
Leia i;
soma = soma + i;
Escreva _"Resultado" soma;
Fim-Para;
Escreva _"A soma utilizando a instrução WHILE: "
soma2 = 0;
i = 0;
Enquanto (i<=100) faça
Leia i;
soma2 = soma2 + i;
Escreva _"Resultado" soma2;
Fim-Enquanto;
Fim;

```

AutoSave 2009... Sal... 검색 Leave

File Home Insert Draw Design Layout References Mailings Review View Help

Clipboard

Font Styles

EXERCÍCIOS LIVRO DIDÁTICO DA 156 A 159
EXERCÍCIOS CADerno DE EXERCÍCIOS 56 A 59

Page 6 of 6 772 words Portuguese (Brazil)

Reunião em "General" 38:23

LOUISY DAL... LUIZ PINHEIRO... CICERO ED... RODRIGO ... +34

Leave

EXERCÍCIOS LIVRO DIDATICO DA 156 A 159 EXERCÍCIOS CADerno DE EXERCÍCIOS 56 A 59

Microsoft Word - Documento1 - Microsoft Word

Arquivo Página Início Layout da Referência Correspondência Revisão Exibição

```
Escreva_."A soma utilizando a instrução WHILE: "
soma2 = 0;
i = 0;
Enquanto (i<=100) faça
    Leia i;
    soma2 = soma2 + i;
    Escreva_."Resultado:" soma2;
Fim-Enquanto;
Fim.

Numerol()
Var
    num,soma : inteiro;
Início
    num = 1;
    Leia num;
    Para num de 1 ate n faça
        Soma = soma + num;
        Escreva_."A soma dos números é:" soma;
    Fim-Para;
Fim.
```

Página:1 de 2 Palavras: 74 100% Focus

Reunião em "General" 41:22

LOUISY DAL... LUIZ PINHEIRO... CICERO ED... RODRIGO ... +34

Leave

EXERCÍCIOS LIVRO DIDATICO DA 156 A 159 EXERCÍCIOS CADerno DE EXERCÍCIOS 56 A 59

Microsoft Word - Documento1 - Microsoft Word

Arquivo Página Início Layout da Referência Correspondência Revisão Exibição

```
Numerol()
Var
    num,soma : inteiro;
Início
    num = 1;
    Leia num;
    Para num de 1 ate n faça
        Soma = soma + num;
        Escreva_."A soma dos números é:" soma;
    Fim-Para;
Fim.
```

Página:2 de 2 Palavras: 102 100% Focus

```
Numero();
Var
    num1, num2, soma : inteiro;
Inicio
    num1 = 1;
    num2 = 1;
    Leia num1,num2;
    Para num de 1 ate n faça
        Soma = num + num2;
        Escreva " A soma dos números é: " soma;
    Fim-Para;
Fim.
```

```
Numero();
Var
    num1, num2,num3, soma : inteiro;
Inicio
    num1 = 1;
    num2 = 1;
    Leia num1,num2;
    Enquanto (num<=100) e (num2 <=100) faça
        Soma = soma + num;
        Escreva " A soma dos números é: " soma;
    Fim-Enquanto;
Fim.
```

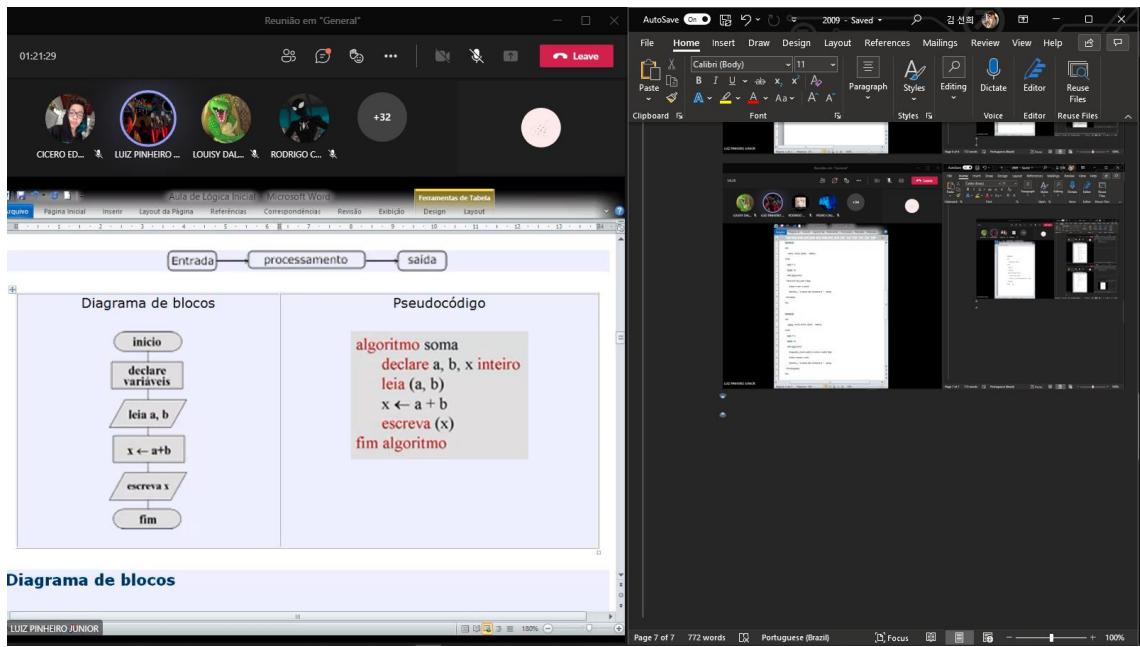
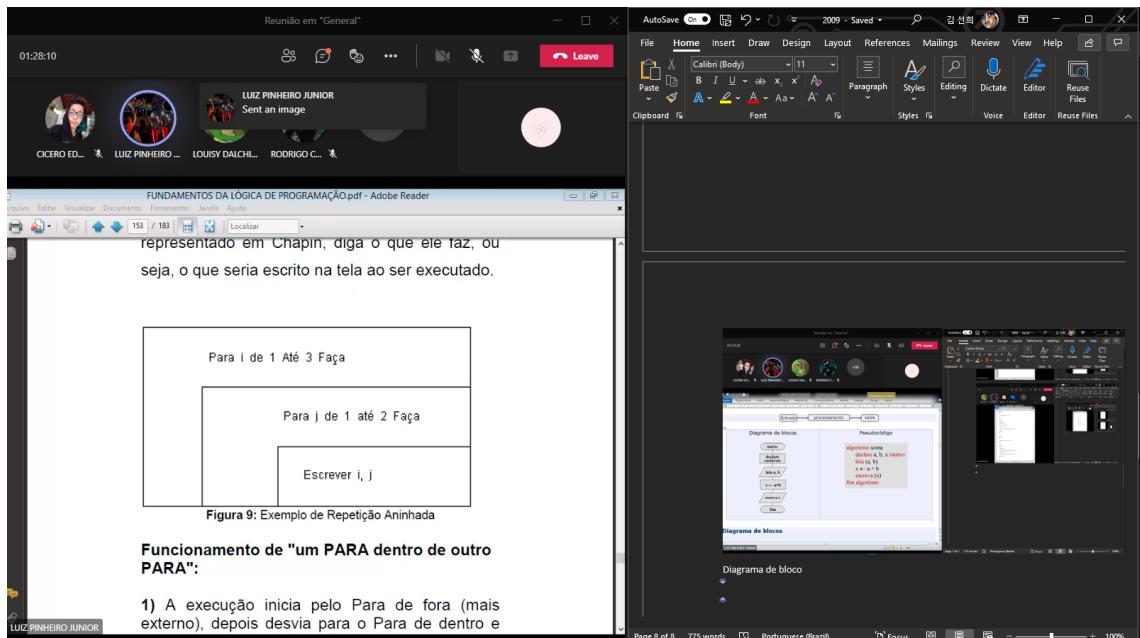


Diagrama de blocos



2709

1.2 Algoritmo Media Valor

Dado uma série de 20 valores reais, faça um algoritmo que calcule e escreva a média aritmética destes valores, entretanto se a média obtida for maior que 8 deverá ser atribuída 10 para a média.

(Ctrl)

Reunião em "General"

44:20

LUIZ PINHEIRO... CICERO ED... LOUISY DAL... FEUFE FERR...

+32

Documentos - Word

Automovel()

Variavel
Nome : char;
Precofab; precofin; imposto; real;
Inicio
 Ler Nome,precofab;
 Imposto = Precofab * (0,45% + 0,28%);
 precofin = Precofab + imposto;
 Escreva Nome,precofin;

Fim.

O preço de um automóvel é calculado pela soma do preço de fábrica com o preço dos impostos (45% do preço de fábrica) e a percentagem do revendedor (28% do preço de fábrica). Faça um algoritmo que leia o nome do automóvel e o preço de fábrica e imprima o nome do automóvel e o preço final.

1. AaE 1.1 AaE

Page 7 of 7 971 words Portuguese (Brazil) Focus + 100%

Reunião em "General"

01:03:40

LUIZ PINHEIRO... CICERO ED... LOUISY DAL... FEUFE FERR...

+29

Documentos - Word

Automovel()

Variavel
Nome : caracte;

Precofab; precofin; imposto; real;

1 Inicio
2 Ler Nome,precofab;
3 Imposto = Precofab * (0,45% + 0,28%);
4 precofin = Precofab + imposto;
5 Escreva Nome,precofin;

Fim.

O preço de um automóvel é calculado pela soma do preço de fábrica com o preço dos impostos (45% do preço de fábrica) e a percentagem do revendedor (28% do preço de fábrica). Faça um algoritmo que leia o nome do automóvel e o preço de fábrica e imprima o nome do automóvel e o preço final.

1. AaE 1.1 AaE

Page 7 of 7 971 words Portuguese (Brazil) Focus + 100%

Reunião em "General"

01:07:59

LUIZ PINHEIRO ... CICERO ED... LOUISY DAL... FELIPE FERR...

+28

Documentos - Word

Variável
Nome : caracte;
Precofab; precofin; imposto: real;
1 Início;
2 Ler Nome,precofab;
3 Imposto = Precofab * (0,45% + 0,28%);
4 precofin = Precofab + imposto;
5 Escreva Nome,precofin;
Fim.

Diagrama de fluxo:

```

graph TD
    1[1] --> 2[2]
    2 --> 3[3]
    3 --> 4[4]
    4 --> 5[5]

```

Diagrama de estrutura de dados (Diagrama de Pilha):

```

graph TD
    1[1] --- 2[2]
    2 --- 3[3]
    3 --- 4[4]
    4 --- 5[5]

```

Reunião em "General"

01:12:13

LUIZ PINHEIRO ... CICERO ED... LOUISY DAL... FELIPE FERR...

+28

Documentos - Word

Automovel()
Variável
Nome : caracte;
Precofab; precofin; imposto: real;
1 Início;
2 Ler Nome,precofab;
3 Imposto = Precofab * (0,45% + 0,28%);
4 precofin = Precofab + imposto;
5 Escreva Nome,precofin;
Fim.

Diagrama de fluxo:

```

graph TD
    1[1] --> 2[2]
    2 --> 3[3]
    3 --> 4[4]
    4 --> 5[5]

```

Diagrama de estrutura de dados (Diagrama de Pilha):

```

graph TD
    1[1] --- 2[2]
    2 --- 3[3]
    3 --- 4[4]
    4 --- 5[5]

```

Reunião em "General"

01:27:08

Meeting chat

```

Então ("element3 = Numpar")
Senão ("element3 =
Numimp1")
Escreva ("Digite o elemento 4")
Leia (element4)
Se ("element4/2 = 0")
Então ("element4 = Numpar")
Senão ("element4 =
Numimp4")
Escreva ("Digite o elemento 5")
Leia (element5)
Se ("element5/2 = 0")
Então ("element5 = Numpar")
Senão ("element5 =
Numimp1")
Soma<- Numimp1 +
Numimp2 + Numimp3 +
Numimp4 + Numimp5
Leia Totalsom;
Escreva: Totalsom;
Fim-Se
Fim.

```

CICERO EDUARDO CAMPOS LEITE
BISPO - 1425
Professor, a variável tem que usar número real, ou pode ser número inteiro também ?

Type a new message

Page 9 of 9 971 words Portuguese (Brazil) Focus 100%

Reunião em "General"

01:52:43

KPop

#	TITLE	ALBUM	DATE ADDED
30	ITZY	BORDER : CARNIVAL	Jul 5, 2021
31	Drunk-Dazed	ENHYPEN	Jun 14, 2021
32	See U Later	SQUARE UP	3:19
33	Answer	BLACKPINK	May 13, 2021
34	The Eve	ATEEZ	3:40
35	Obsession	THE WAR - The 4th Album	Apr 27, 2021
36	MOMMAE	EXO	2:56
	Left & Right	OBSSESSION - The 6th Album	Apr 27, 2021
	Dolphin	NONSTOP	3:24
	Dolphin	OH MY GIRL	3:25
	Dolphin	Heng:garæ	2:22
	Dolphin	NONSTOP	2:57

LUIZ PINHEIRO JUNIOR

Type a new message

Page 9 of 9 971 words Portuguese (Brazil) Focus

1. Elabore um programa que efetue a leitura de 5 elementos de uma matriz A do tipo vetor. No final, apresente o total da soma de todos os elementos que sejam ímpares. Pede-se Algoritmo, Diagrama de Bloco e de Chapin.

2. Elabore um programa que efetue leitura de 5 elementos, criando duas matrizes:

A tipo vetor;

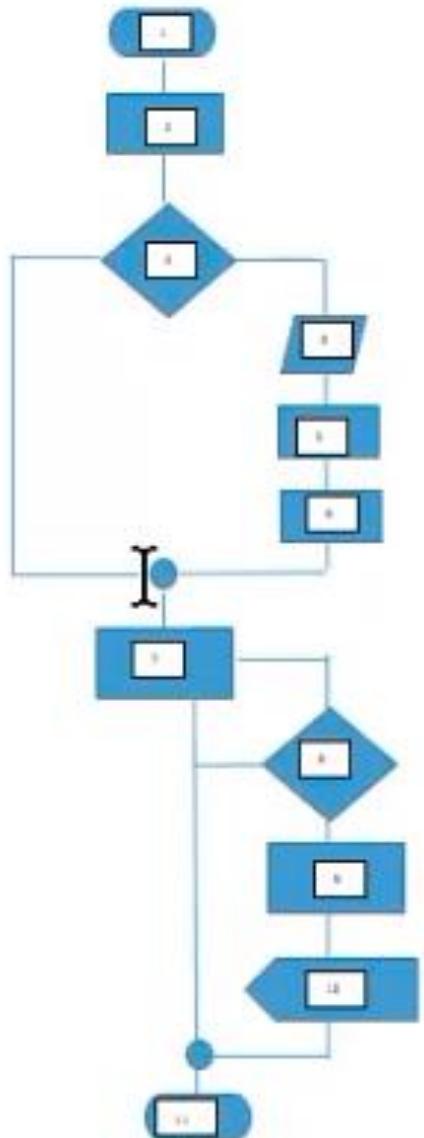
B tipo vetor, se o valor for PAR, multiplicar por 5, se for IMPAR, deverá ser somado por 5.

Deverá ser mostrado os conteúdos das duas matrizes. Pede-se Algoritmo, Diagrama de Bloco e de Chapin.

3) Elabore um programa que lê e calcule a média aritmética de dois números e solicita ao usuário, se deseja continuar.

- Um algoritmo com o digrama de bloco e outro com o de Chapin.

```
SérieU  
Variáveis  
Contador:inteiro;  
N,media,soma:real;  
1 Início  
2   contador=0;  
3   soma= 0;  
4   Enquanto (contador <20) faça  
5     Ler N;  
6     soma = soma + N;  
7     contador = contador +1;  
8   Fim-Enquanto;  
9   media=(soma/20);  
10  Se (media >8) então  
11    Media = 10;  
12  Fim-Se;  
13  Escreva media;  
14 Fim.
```



1810

Reunião em "General"

Feed

Activity

Calendar

Teams

Assignments

Chat

13:23

Exercícios de Lógica utilizando Vetor e Matriz

8,224 Crie um programa que leia um vetor A de 50 números inteiros e crie um vetor B de mesmo tamanho cujos elementos de posições ímpares são a metade do correspondente elemento do vetor A e os de posição para são o dobro.

Programa ex8

Variáveis

```
a,b: vetor [1..50] de inteiro;
j,par: inteiro;
Início
Para i de 1 até 50 faça
Leia a[i];
```

LUIZ PINHEIRO JUNIOR posted 15/10

ETESP > General

Reply

BIO posted 08/10

New conversation

17°C Light rain

13:14

Reunião em "General"

Feed

Activity

Calendar

Teams

Assignments

Chat

20:34

Exercícios de Lógica utilizando Vetor e Matriz

8,224 Crie um vetor B de mesmo tamanho cujos elementos de posições ímpares são a metade do correspondente elemento do vetor A e os de posição para são o dobro.

Programa ex8

Variáveis

```
a,b: vetor [1..50] de inteiro;
j,par: inteiro;
Início
Para i de 1 até 50 faça
Leia a[i];
```

LUIZ PINHEIRO JUNIOR posted 15/10

ETESP > General

Reply

CICERO EDUARDO CAMPOS LEITE BISPO posted 13:16

Professor, eu acharia caindo da reunião na hora que o senhor saiu da imensa da menina, o senhor falou alto?

arabéns pessoal!!!

New conversation

Reunião em "General"

13:50

Exercícios de Lógica utilizando Vetor e Matrizes.docx (Modo de Compatibilidade) - Word

Leave

g soon

tures early and send us what you think.

Show me Not now

Activity

Feed

Calendar

Teams

Assignments

Chat

LUIZ PINHEIRO JUNIOR posted 15/10

ETESP > General a dos professores!!!

BIO posted 08/10

ETESP > General antes de escrever...rsrs Parabéns pessoal!!!!

New conversation

Programa ex8

Variáveis

a,b: vetor [1..50] de inteiro;
i,par: inteiro;

Início

Para i de 1 até 50 faça
Leia a[i];

luizpinheirojr posted 15/10

quero sua opinião

Reply

LUIS VINÍCIUS FERREIRA posted 15/10

RODRIGO CARVALHO posted 15/10

LOUSY DALLA COSTA posted 15/10

JOSE FELIPE DE SOUZA posted 15/10

RAPHAEL OLIVEIRA posted 15/10

+27

focodigo

Reunião em "General"

20:20

Exercícios de Lógica utilizando Vetor e Matrizes.docx (Modo de Compatibilidade) - Word

Leave

Editor Reuse Files

Editor Reuse Files

Activity

Feed

Calendar

Teams

Assignments

Chat

LUIZ PINHEIRO JUNIOR posted 15/10

ETESP > General a dos professores!!!

CICERO EDUARDO CAMPOS LEBE BISPO posted 13:16

Professor, eu acabei caindo da reunião na hora que o senhor saiu da imaculada menina, o senhor falou algo?

New conversation

Fim-Para;

Para i de 1 até 50 faça

Par = i mod 2;

Se (par = 0) então

b[i] = 2 * a[i]

senão

b[i] = 2 / a[i];

Fim-se;

Fim-Para;

Para i de 1 até 50 faça

Escreva b[i];

Fim-Para;

Fim-

package exe1;

luizpinheirojr posted 15/10

arabéns pessoal!!!!

Back to tab

Definição de vetor

O vetor pode ser considerado uma variável multidimensional indexada, ou seja, uma variável que tem a capacidade de guardar mais de um valor simultaneamente do mesmo tipo.

Quando falamos de vetor temos que ter a seguinte visão, uma variável comum consegue guardar apenas um valor por vez, enquanto um vetor consegue armazenar diversos valores cada um em uma posição diferente.

Ao observar a tabela 1 podemos entender como um vetor funciona, por exemplo o valor 10 está localizado na posição 0.

INDICE	0	1	2	3	4	5
VALORES	10	7	8	3	9	6

Back to tab

15/10 CICERO EDUARDO CAMPOS LEITE BISPO 13:16 Professor, eu acabei caindo da reunião na hora que o senhor saiu da imanem da menina, o senhor falou algo?

Exemplificando

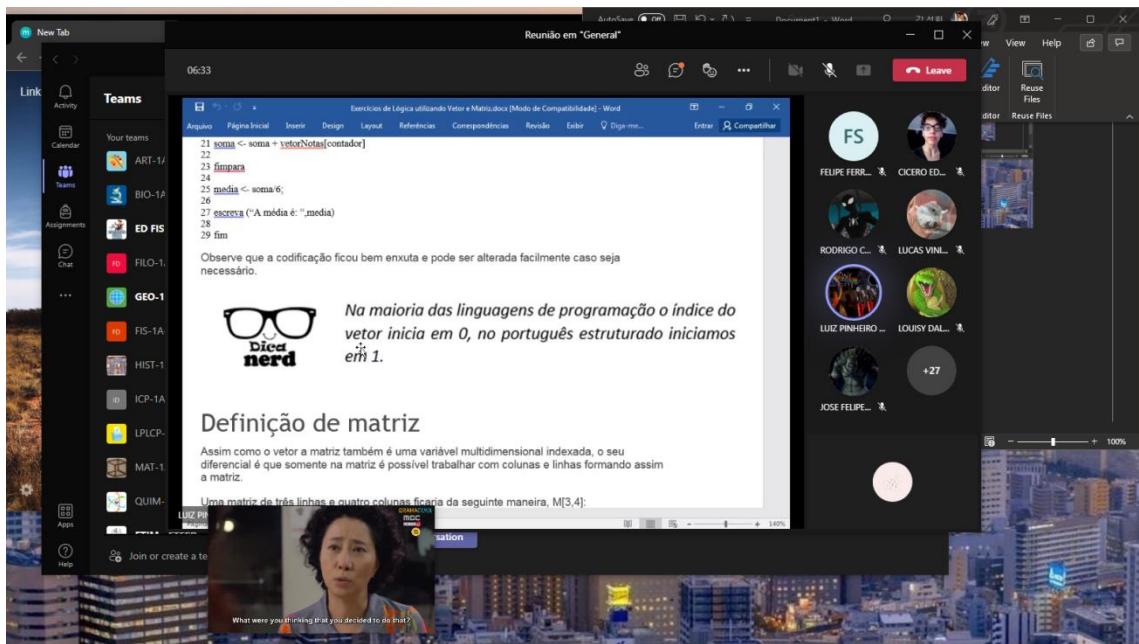
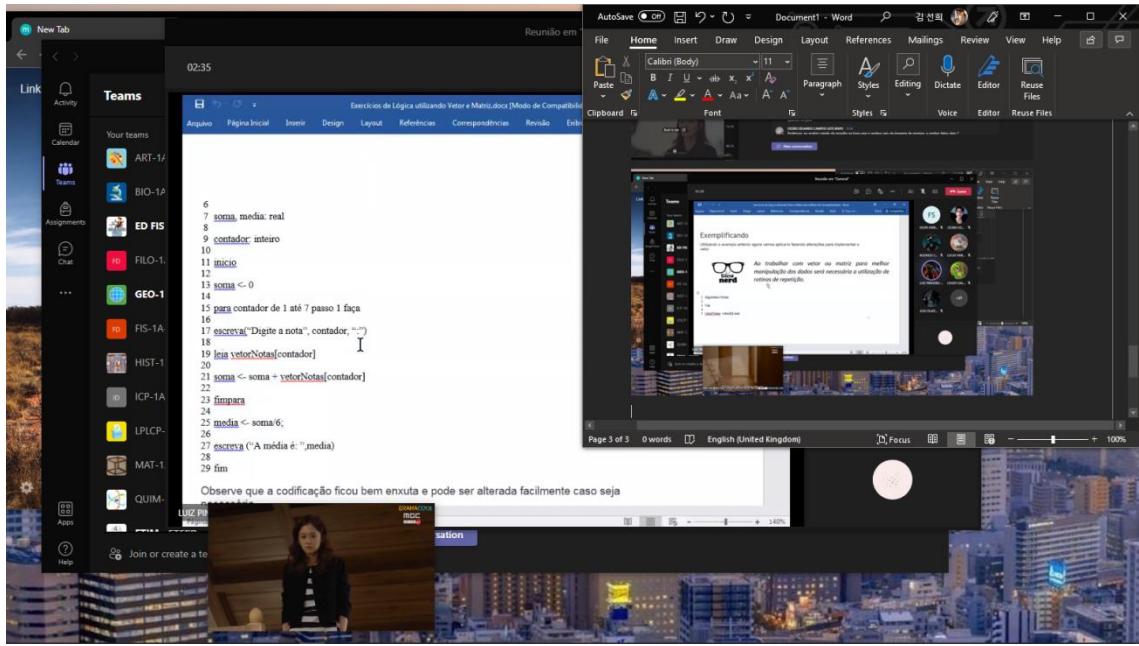
Utilizando o exemplo anterior agora vamos aplicá-lo fazendo alterações para implementar o vetor.

Dicas nerd

Ao trabalhar com vetor ou matriz para melhor manipulação dos dados será necessária a utilização de rotinas de repetição.

```
1 Algoritmo Notas
2
3 Var
4 vetorNotas: vetor[6] real
```

Join or create a team



Definição de matriz

Assim como o vetor a matriz também é uma variável multidimensional indexada, o seu diferencial é que somente na matriz é possível trabalhar com colunas e linhas formando assim a matriz.

Uma matriz de três linhas e quatro colunas ficaria da seguinte maneira, M[3,4]:

L1C1	L1C2	L1C3	L1C4
L2C1	L2C2	L2C3	L2C4
L3C1	L3C2	L3C3	L3C4

Tabela 2 - Exemplo de como uma Matriz trabalha
Ao observar a matriz representada acima podemos enxergar a linha e a coluna de cada posição, numericamente essa tabela ficaria da seguinte forma:

2510

MERCULHANDO NO TEMA...

Você já sabe que, na programação, existem variáveis especiais capazes de armazenar múltiplos dados, utilizados para organizar e facilitar a escrita do seu código, não é mesmo? Matrizes, conhecidas também por Arrays (termo originalmente de inglês), são variáveis desse tipo. Elas podem guardar múltiplos valores em mais de uma dimensão, ou seja, seus dados podem ser armazenados ou lidos de acordo com a relação entre elas.

Os elementos de uma matriz são ordenados como em um jogo de Batalha Naval: em linhas e colunas identificadas. Mas, diferente do jogo, para indicar a localização dos elementos dentro da matriz, são utilizados apenas números, chamados de índices.

Reunião em "General"

12:03

LUIZ PINHEIRO AUGUSTO F... LOUISY DAL... RODRIGO ... +34

Leave

Salvamento Automático

Início, Agentes, Matriz - Modo de Compatibilidade - Salvo no PC

Arquivo Página Inicial Insere Desenhar Design Layout Referências Mailings Review View Help

AutoSave

Clipboard

Font

Styles

Editing

Dictate

Editor

Voice

Reuse Files

Page 3 of 3 740 words Portuguese (Brazil)

90%

seus dados podem ser armazenados de forma que a relação entre essas.

Os elementos de uma matriz são ordenados como em um jogo de Batalha Naval: em linhas e colunas identificadas. Mas, diferentemente do jogo, para indicar a localização dos elementos dentro da matriz, são utilizados apenas números, chamados de índices.

Batalha Naval

Figura 4 de 21 321 páginas Portugal (Português)

LUIZ PINHEIRO JUNIOR

Reunião em "General"

12:58

LUIZ PINHEIRO AUGUSTO F... LOUISY DAL... RODRIGO ... +33

Leave

Salvamento Automático

Início, Agentes, Matriz - Modo de Compatibilidade - Salvo no PC

Arquivo Página Inicial Insere Desenhar Design Layout Referências Mailings Review View Help

AutoSave

Clipboard

Font

Styles

Editing

Dictate

Editor

Voice

Reuse Files

Page 3 of 3 740 words Portuguese (Brazil)

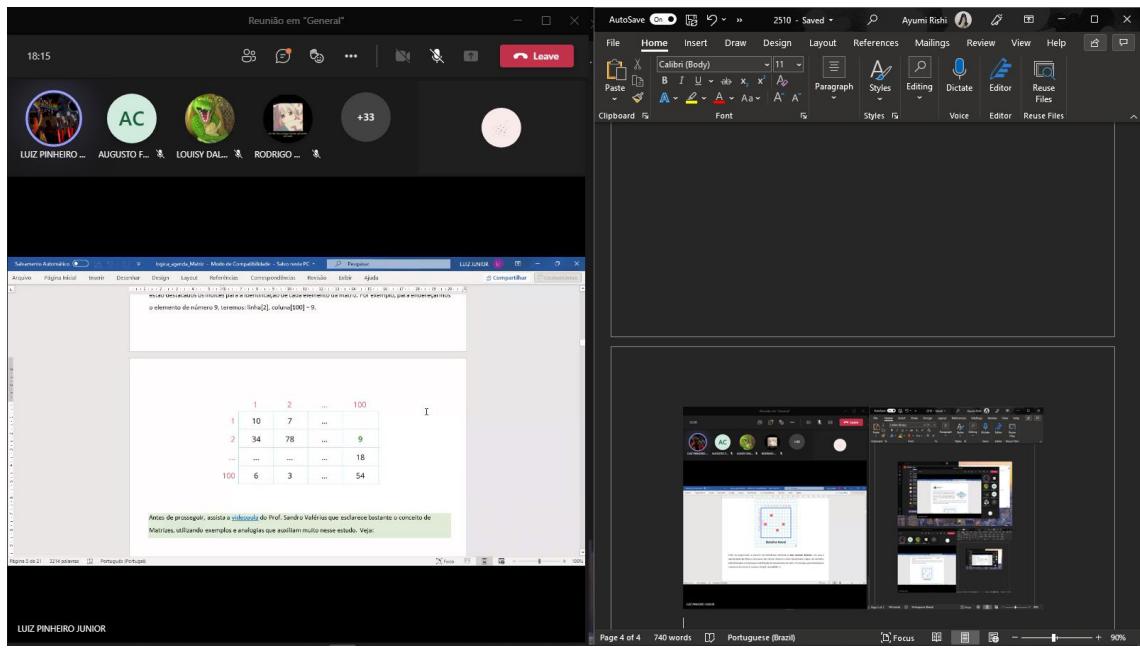
90%

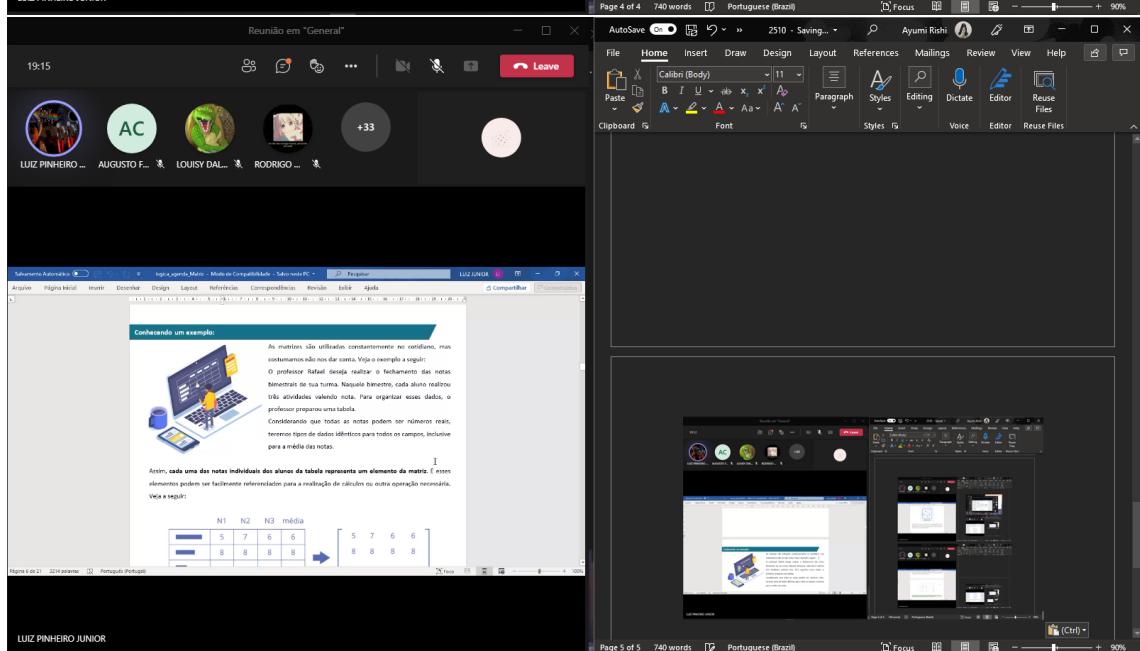
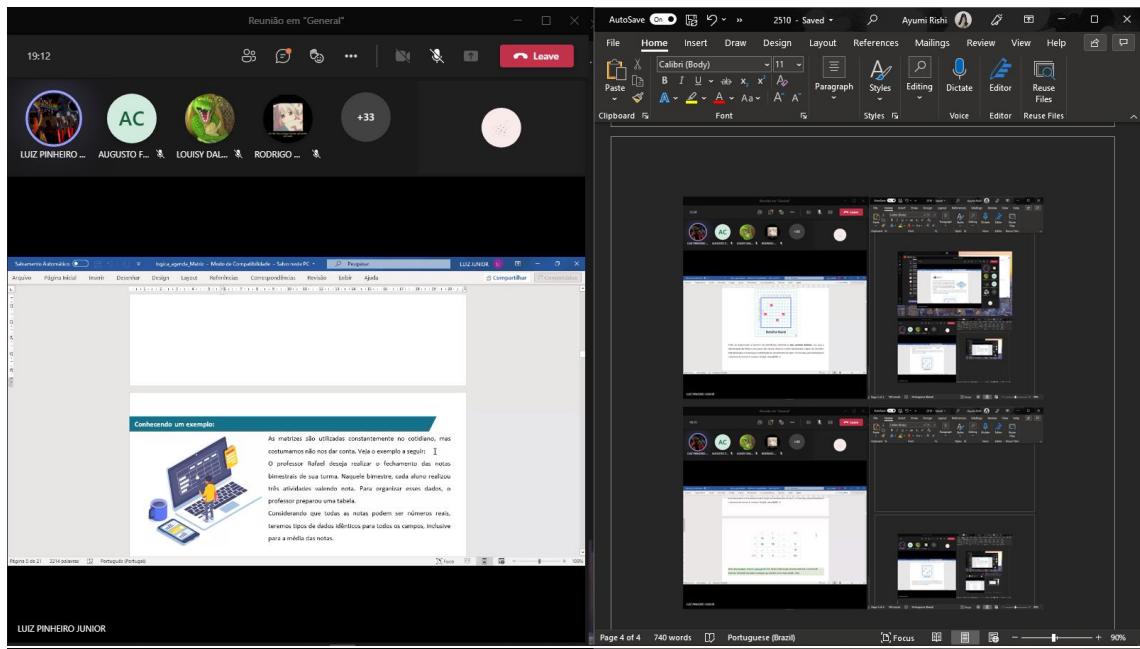
Então, na programação as matrizes são identificadas utilizando-se duas variáveis distintas: uma para a representação das linhas e outra para a das colunas. Observe a matriz representada a seguir: em vermelho, estão destacados os índices para a identificação de cada elemento da matriz. Por exemplo, para endereçarmos o elemento de número 9, teremos: linha[2], coluna[10] - 9.

Batalha Naval

Figura 4 de 21 321 páginas Portugal (Português)

LUIZ PINHEIRO JUNIOR





Reunião em "General"

23:56

LUIZ PINHEIRO, ITALO PEDR., LOUISY DALI, RODRIGO ... +34

Leave

Tutoriais Automatico

Início Aplicações Microsoft - Modo de Compatibilidade - Salvar no PC

Arquivo Página Inicial História Desenvolver Design Layout Referências Correspondência Visualizar Ajuda Ajuda

Visualizar

Veja a seguir:

	N1	N2	N3	média
1	5	7	6	6
2	8	8	8	8
3	—	—	—	—
4	10	8	9	9

Tabela de Notas

MATRIZ

Na matriz representada acima, temos as notas das 3 atividades nas três primeiras colunas e a média bimestral na última coluna. Se quisermos buscar a terceira nota do primeiro aluno, teremos: `matriz[1][2]` e é só. As matrizes são um modo muito simples e eficaz de representarmos os dados em programação.

Começar a usar uma matriz em Java?

Agora que você já compreendeu o que é uma matriz e como ela é declarada, é momento de focar na parte prática da utilização em pseudocódigo e em Java. Veja a sintaxe da declaração de uma matriz bidimensional, a seguir:

Page 4 of 21 321 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

Reunião em "General"

24:53

LUIZ PINHEIRO, ITALO PEDR., LOUISY DALI, RODRIGO ... +34

Leave

Tutoriais Automatico

Início Aplicações Microsoft - Modo de Compatibilidade - Salvar no PC

Arquivo Página Inicial História Desenvolver Design Layout Referências Correspondência Visualizar Ajuda Ajuda

Visualizar

matrizes são um modo muito simples e eficaz de representarmos os dados em programação.

Começar a usar uma matriz em Java?

Agora que você já compreendeu o que é uma matriz e como ela é declarada, é momento de focar na parte prática da utilização em pseudocódigo e em Java. Veja a sintaxe da declaração de uma matriz bidimensional, a seguir:

PSEUDOCÓDIGO		JAVA	
<code>Declarar</code>	<code><seu nome> [1..n][1..m] de tipo;</code>	<code>Object <seu nome>[] = new Object[n][m];</code>	

Onde temos `<seu nome>`, você indicará o nome da matriz. Em `código`, deve ser indicado o tipo de variável a ser armazenada. Em `n`, você deverá especificar o número de linhas e, em `m`, o de colunas.

Veja um exemplo de declaração de matriz bidimensional:

Page 5 of 6 740 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

Reunião em "General"

27:59

LUIZ PINHEIRO, ITALO PEDR., AUGUSTO F., RODRIGO ... +34

Ayumi Rishi

AutoSave 2510 - Saved

Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Styles Paragraph Editing Dictate Editor Reuse Files

Página Inicial Desenvolver Design Layout Referências Recursos Revisão Visualizar Arquivo

Tela de apresentação Automática

Início da apresentação - Modo de Compatibilidade - Salvo no PC

Prévia

LUIZ JUNIOR

Page 6 of 6 740 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

Reunião em "General"

42:44

LUIZ PINHEIRO, LOUSY DAI., ITALO PEDR., CICERO ED. ... +33

Ayumi Rishi

AutoSave 2510 - Saved

Home Insert Draw Design Layout References Mailings Review View Help

Clipboard Font Styles Paragraph Editing Dictate Editor Reuse Files

Página Inicial Desenvolver Design Layout Referências Recursos Revisão Visualizar Arquivo

Tela de apresentação Automática

Início da apresentação - Modo de Compatibilidade - Salvo no PC

Prévia

LUIZ JUNIOR

Page 7 of 7 740 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

Reunião em "General"

43:41

LUIZ PINHEIRO, LOUSY DAL, ITALO PEDR, CICERO ED...

+32

Clipboard

Page 6 of 21 | 1 de 324 páginas | Português (Brasil)

LUIZ PINHEIRO JUNIOR

Aqui valem uma observação muito importante: note que foram ilustrados duas matrizes acima. Você deve reparar que na matriz de pseudocódigo, a contagem inicia-se pelo número 0 (ímpar) já na matriz de Java, a contagem inicia-se pelo número 0 (par). Lembre-se disso! Quando utilizarmos Arrays em Java, saibam que automaticamente (pelo menos no Java) os indices iniciam a contagem sempre se inicia do zero.

Dica:

Isto mesmo! Você pode utilizar cinco tipos: **Arrays Unidimensionais** para informar o uso de vetores (matrizes unidimensionais) e **Arrays Bidimensionais** para o uso de matrizes bidimensionais. Isso é a diferença!

Veja agora outro exemplo:

PSEUDOCÓDIGO

nomes como conjunto [1..10][1..5] de caracteres

JAVA

```
String nome[][] = new String[10][5];
double num[][] = new double[4][4];
float num1[][] = new float[4][4];
```

Page 7 of 7 | 740 words | Portuguese (Brazil)

Reunião em "General"

45:37

LUIZ PINHEIRO, LOUSY DAL, ITALO PEDR, CICERO ED...

+33

Clipboard

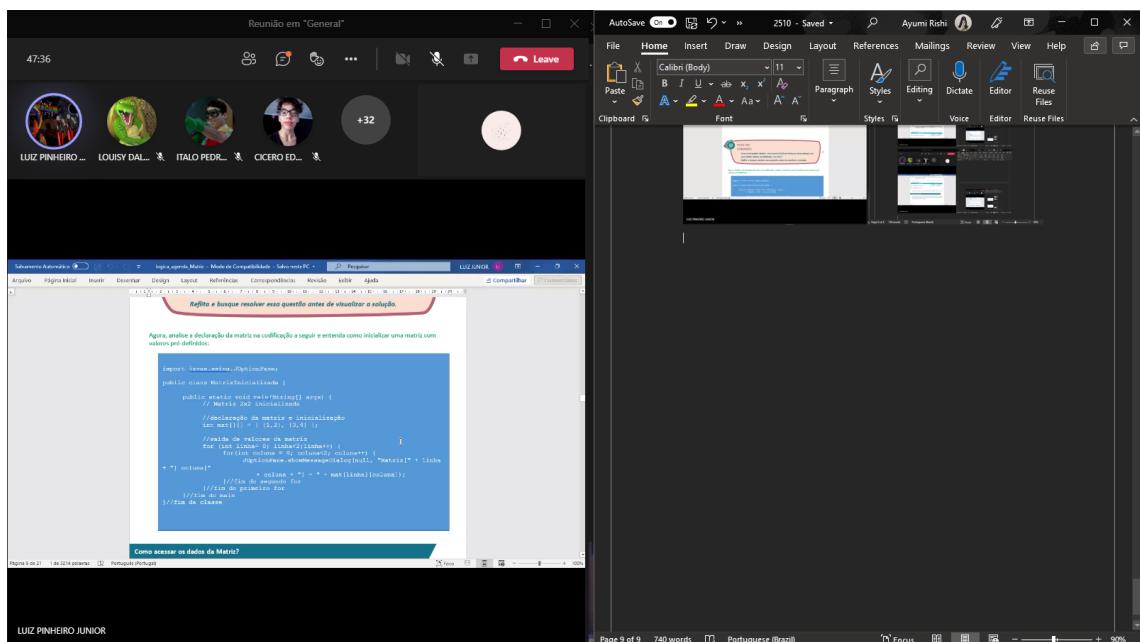
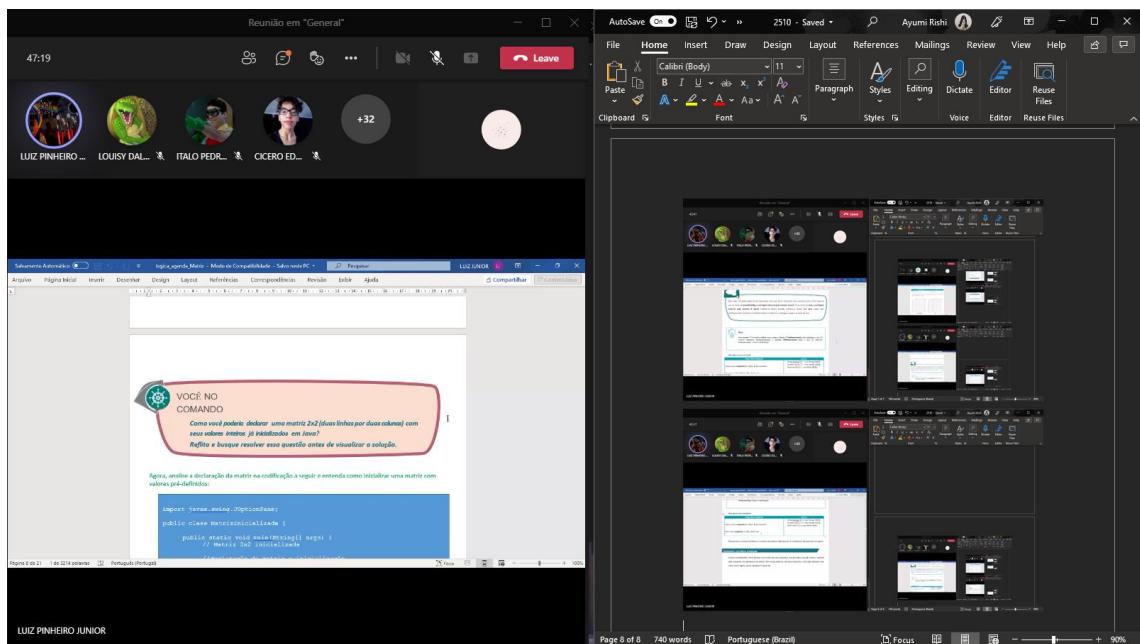
Page 8 of 8 | 1 de 324 páginas | Português (Brasil)

LUIZ PINHEIRO JUNIOR

Observa que o número de linhas e o número de colunas indicados entre [] (chaves) não precisam ser iguais.

Declarando uma Matriz inicializada

Você já compreendeu como declarar uma matriz em seu programa, mas, por favor, que não indica nenhum valor presente nos elementos da matriz? Em outras palavras, até este momento, você só pode declarar uma matriz vazia. Agora, por favor, pesquise e responda:



Reunião em "General"

52:04

LUIZ PINHEIRO — LOUSY DAL... ITALO PEDR... CICERO ED...

Leave

Como acessar os dados de Matriz?

O acesso aos elementos de uma matriz é feito utilizando comandos de repetição. Durante o curso, você viu três comandos desse tipo, sendo o para... Repetir (no Java) o mais adequado para utilização com matrizes. Esse comando percorre todos os elementos de uma matriz por um número fixo de vezes e, como você já sabe, uma matriz possui um número fixo de linhas e colunas.

O exemplo a seguir considera uma matriz A de números reais, com as variáveis linha e coluna referentias ao acesso à linha e à coluna das matrizes. Nota: é necessário declarar os valores presentes em cada elemento da matriz, ou seja, trata-se de uma matriz não-inicializada. Veja como ficaria a sintaxe do comando em pseudocódigo e em Java:

```

PSEUDOCÓDIGO           JAVA
Programa MatrizExemplo
Declara
    num como conjunto [1..2]..4 de real
    linha, coluna como inteto
    inicio
    fim
    Para linha = 1 Até 4 Fazer
        Escreva ("Entre com os dados na Matriz")
        Para coluna = 1 Até 4 Fazer
            Escreva ("Entre com um número")
            Entrar ("Entrada", linha)
            Entrar ("Entrada", coluna)
            Letra num (linha, coluna)
        Fim Para
    Fim Para
    Escreva ("Mostrando os dados na Matriz")
    Para linha = 1 Até 4 Fazer
        Escreva ("Entrada", linha)
        Para coluna = 1 Até 4 Fazer
            Escreva ("Entrada", coluna)
            Escreva ("Valor", num (linha, coluna))
        Fim Para
    Fim Para

```

Page 9 of 9 740 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

Reunião em "General"

53:20

LUIZ PINHEIRO — LOUSY DAL... ITALO PEDR... CICERO ED...

Leave

Como acessar os dados de Matriz?

O acesso aos elementos de uma matriz é feito utilizando comandos de repetição. Durante o curso, você viu três comandos desse tipo, sendo o para... Repetir (no Java) o mais adequado para utilização com matrizes. Esse comando percorre todos os elementos de uma matriz por um número fixo de vezes e, como você já sabe, uma matriz possui um número fixo de linhas e colunas.

O exemplo a seguir considera uma matriz A de números reais, com as variáveis linha e coluna referentias ao acesso à linha e à coluna das matrizes. Nota: é necessário declarar os valores presentes em cada elemento da matriz, ou seja, trata-se de uma matriz não-inicializada. Veja como ficaria a sintaxe do comando em pseudocódigo e em Java:

```

PSEUDOCÓDIGO           JAVA
Programa MatrizExemplo
Declara
    num como conjunto [1..2]..4 de real
    linha, coluna como inteto
    inicio
    fim
    Para linha = 1 Até 4 Fazer
        Escreva ("Entre com os dados na Matriz")
        Para coluna = 1 Até 4 Fazer
            Escreva ("Entre com um número")
            Entrar ("Entrada", linha)
            Entrar ("Entrada", coluna)
            Letra num (linha, coluna)
        Fim Para
    Fim Para
    Escreva ("Mostrando os dados na Matriz")
    Para linha = 1 Até 4 Fazer
        Escreva ("Entrada", linha)
        Para coluna = 1 Até 4 Fazer
            Escreva ("Entrada", coluna)
            Escreva ("Valor", num (linha, coluna))
        Fim Para
    Fim Para

```

Page 10 of 10 740 words Portuguese (Brazil)

LUIZ PINHEIRO JUNIOR

Reunião em "General"

01:01:02

LUIZ PINHEIRO, LOUSY DAL, ITALO PEDR, CICERO ED...

+31

Leave

Salvo no Arquivo

Impresso, Móvel, Modo de Compatibilidade, Salvo no PC, Página

Arquivo Fórum Início Desenvolver Design Layout Referências Recursos Ajuda Design da Tabela Logout Compartilhar

Refira e busque resolver essa questão antes de visualizar a solução.

Percorra que, para resolver o problema, basta mover o JOptionPane para fora do laço for:

```

for(int linha = 0; linha < 10; linha++) {
    for(int coluna = 0; coluna < 4; coluna++) {
        mat[linha][coluna] = random.nextInt(100);
        JOptionPane.showMessageDialog(null, "Número sorteado com o número " + mat[linha][coluna]);
    }
}

```

Outra opção seria utilizar o comando System.out.print que realiza a impressão no console do programa.

Page 10 of 10 740 words Portuguese (Brazil) Focus

LUIZ PINHEIRO JUNIOR

Reunião em "General"

01:26:41

ITALO PEDR, CICERO ED..., LUIZ PINHEIRO, LOUSY DAL...

+30

Leave

Salvo no Arquivo

Impresso, Móvel, Modo de Compatibilidade, Salvo no PC, Página

Arquivo Fórum Início Desenvolver Design Layout Referências Recursos Ajuda Design da Tabela Logout Compartilhar

Retomando ao Exemplo

No começo desse texto, você viu que o professor Rafael descreve calcular a média bimestral de um encontro aluno x, para isso, organizou os dados em uma matriz. Considerando que a turma tem 10 alunos, cada um realizou 5 atividades e uma coluna deve ser reservada para a média, sabemos que essa matriz tem 10 linhas e 6 colunas.

Como podemos codificar um programa que realize esse cálculo?

Veja o pseudocódigo e a codificação em Java:

```


```

Page 11 of 11 740 words Portuguese (Brazil) Focus

LUIZ PINHEIRO JUNIOR

```


```

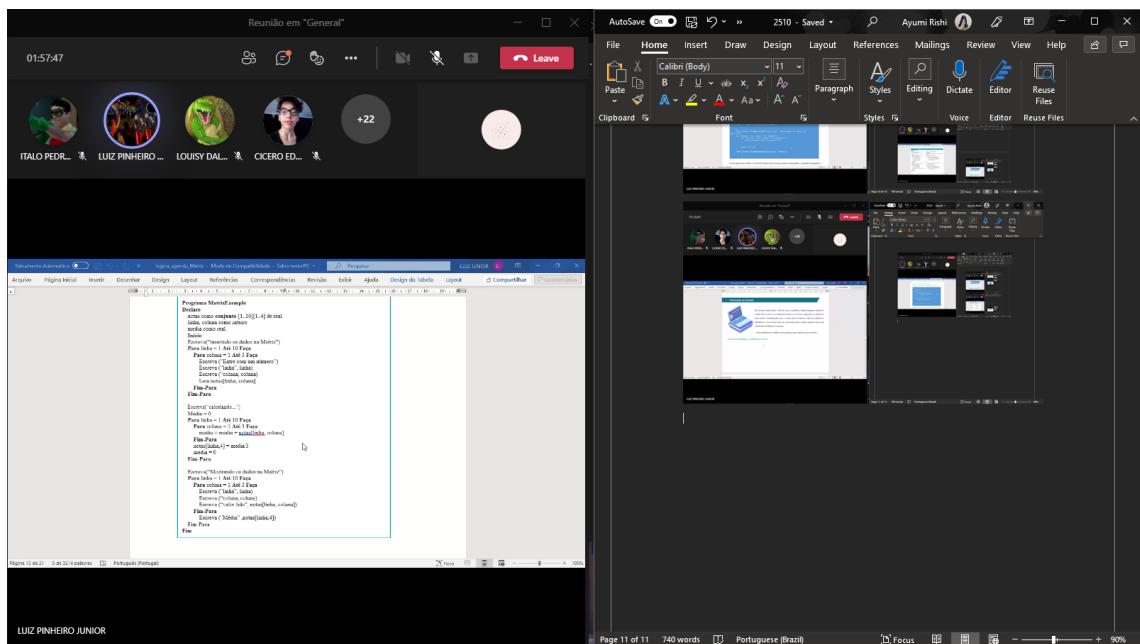


Figura 12 de 21 - 5 de 22 (4 slides) [2] Portuguese (Brazil)

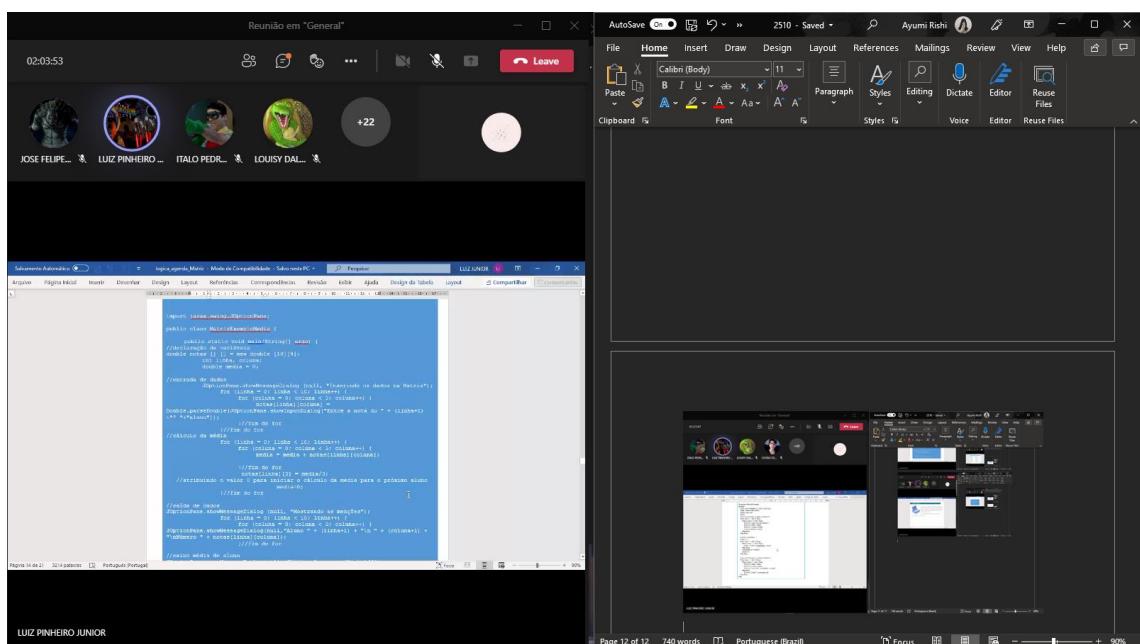
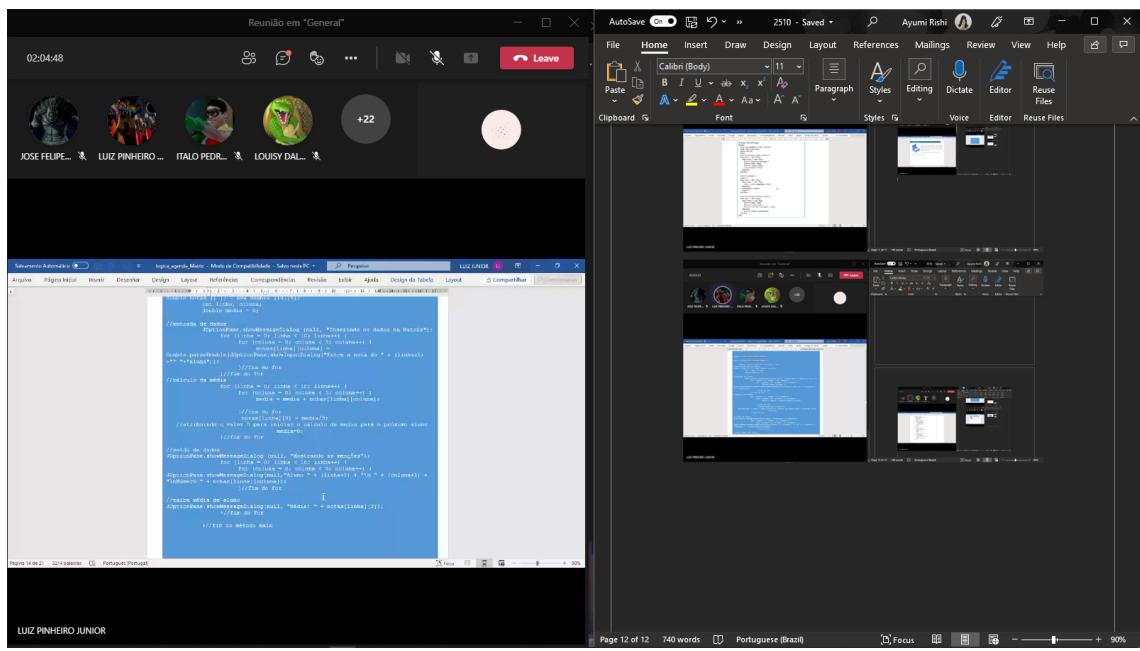


Figura 12 de 21 - 5 de 22 (4 slides) [2] Portuguese (Brazil)



0811

Exercícios de Lógica

Escreva um programa que leia e exiba nomes para um vetor de 100 posições.

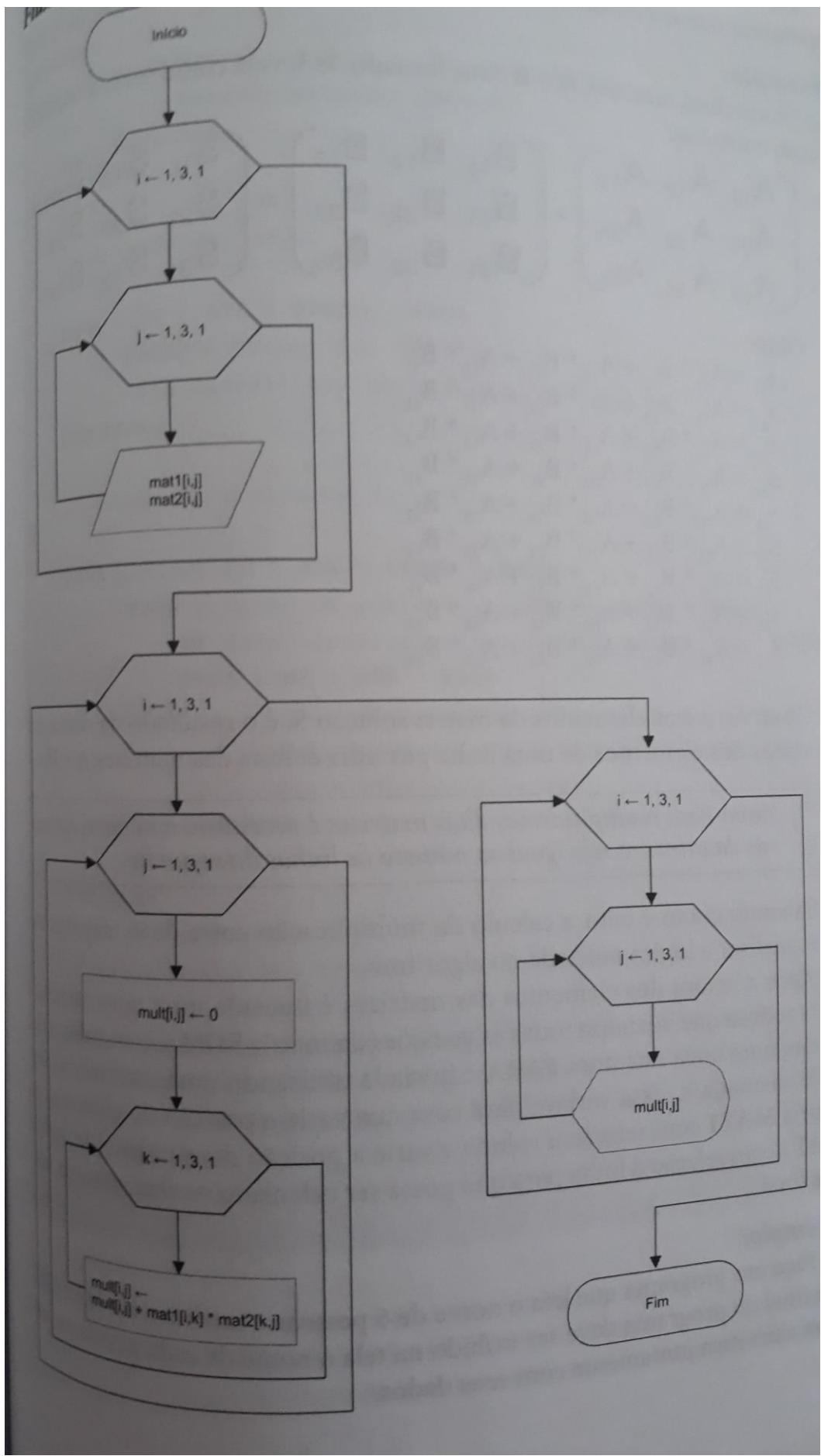
Crie um programa que leia nome de cidades para um vetor de 15 posições, coloque os nomes em ordem crescente e exiba para o usuário.

Crie um programa que leia números inteiros para dois vetores A e B, de 20 posições, calcule a soma entre os dois e exiba o resultado em ordem decrescente em um terceiro vetor.

2211

2911

```
FzrMatriz()
Variaveis
    Mat1: matriz[1..3,1..3] de inteiro;
    Mat2: matriz[1..3,1..3] de inteiro;
    Multi: matriz[1..3,1..3] de inteiro;
    I,J,K : inteiro;
Inicio
    /* Entrada de Dados */
    Para I de 1 até 3 faça
        Para J de 1 até 3 faça
            Leia Mat1[I,J], Mat2[I,J]
        Fim-para;
    Fim-Para;
    /*Multiplicação dos Dados */
    Para I de 1 até 3 faça
        Para J de 1 até 3 faça
            Multi[I,J] =0;
            Para K de 1 até 3 faça
                Multi[I,J] = Multi[I,J] + Mat1[I,K] * Mat2[K,J];
            Fim-para;
        Fim-Para;
    Fim-para;
    /*Saída dos Dados */
    Para I de 1 até 3 faça
        Para J de 1 até 3 faça
            Escreva Multi [I,J];
        Fim-para;
    Fim-Para;
Fim.
```



```

package tabuadagenerator;
import java.util.Scanner;
public class TabuadaGenerator {
    public static void main(String[] args) {
        int contador,num;
        String resposta;
        Scanner n = new Scanner(System.in);
        do {
            System.out.println("Insira o operando");
            num = n.nextInt();
            System.out.println("=====");
            for (contador = 1; contador <=10; contador++) {
                System.out.println(num + " x " + contador + " = " + num*contador);
            }
            System.out.println("=====");
            System.out.println("Deseja fazer outra tabuada? sim/nao?");
            resposta = n.next();
        }while ("sim".equals(resposta));
    }
}

```

- Tabuada do 10

```

- package fatorial;
- import java.util.Scanner;
- /**
- *
- * @author Etesp
- */
- public class Fatorial {
-     /**
-      * @param args the command line arguments
-      */
-     public static void main(String[] args) {
-         // TODO code application logic here
-         Scanner ler = new Scanner(System.in);
-         int fat=1;
-         int auxiliar = 1;
-         System.out.println("Digite um número n:");
-         int n = ler.nextInt();
-         for (int i=1; i<=n; i++){
-             fat = fat*i;
-         }
-         if (fat <= 1){
-             System.out.println("O valor do fatorial de " + "(!" + n + ")" + " " + "é: 1 " + "\n" +
- "!" + n + " = 1" + "\n");
-         }
-         else {
-             System.out.println("O valor do fatorial de " + "(!" + n + ")" + " " + "é: " + fat + "\n" +
- "!" + n + " = " + fat + "\n");
-         }
-     }
- }

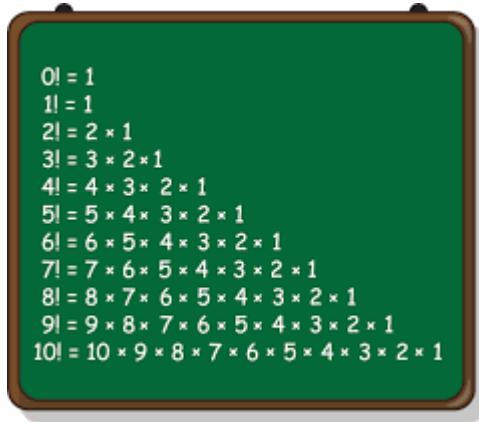
```

- }

- }

- Exercício factorial

Calcular o factorial de um número só faz sentido quando estamos trabalhando com números naturais. Essa operação é bastante comum na análise combinatória, facilitando o cálculo de arranjos, permutações, combinações e demais problemas envolvendo contagem. O factorial é representado pelo símbolo “!”. Definimos como $n!$ (n factorial) a multiplicação de n por todos os seus antecessores até chegar em 1. $n! = n \cdot (n - 1) \cdot (n - 2) \cdot \dots \cdot 3 \cdot 2 \cdot 1$.



0612