# Lógica de Programação

# Entendendo o que é lógica

**Lógica** é coerência de raciocínio de ideias.

**Sequencia coerente, regular e necessária de acontecimentos, coisas.**

**Lógica de programação** significa apenas contextualizar a lógica na programação de computadores, buscando a melhor sequência de ações para solucionar um problema.

**Metacognição**: Pensar como você pensa

**Abstração** é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes ou acidentais.

Exercício1 :

Crie um mapa mental para resolver um determinado problema, por exemplo, calcular a média aritmética de 4 notas, sabendo que as notas são as seguintes:

Nota 1: 5

Nota 2: 7

Nota 3: 10

Nota 4: 3

Exercício 2:

É possível fazer uma abstração a partir do seu mapa mental?

# O que é algoritmos e pseudocódigo

Pseudocódigo é uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples (nativa, ou seja, em português a quem o escreve, de forma a ser entendida por qualquer pessoa)”.

Exercício 1:

Crie um algoritmo do seu dia

Exercício 2:

Abra o site

<https://studio.code.org/s/mc/stage/1/puzzle/1>

e tente resolver todos os problemas em menos de uma hora

# Aprendendo fluxograma, variáveis e constantes

**Fluxograma** é uma ferramenta utilizada para representar graficamente algoritmo, isto é, a sequência lógica e coerente do fluxo de dados.

“Um fluxograma é um tipo de diagrama e pode ser entendido como uma representação esquemática de um processo. Podemos entende-lo, na prática, como a documentação dos passos necessários para a execução de um processo qualquer”

**Diagrama de blocos**: Utilizado para representar o método do fluxograma.

Variáveis e Constantes

Na programação, uma **variável** é um objeto (uma posição, frequentemente localizada na memória) capaz de reter e representar um valor ou expressão”.

Uma **variável** é um espaço na memória do computador destinado a um dado que é alterado durante a execução do algoritmo”

Tipos de variáveis

As variáveis e as constantes podem ser classificadas basicamente de quatro tipos: Numéricas, Caracteres, Alfanuméricas ou Lógicas.

As **constantes** são valores imutáveis e não são alterados durante a vida útil do programa.

# Tomadas de decisões e expressões

**Expressões aritméticas** são expressões que utilizam operadores aritméticos e funções aritméticas envolvendo constantes e variáveis.

Expressões literais

São expressões com constantes e/ou variáveis que tem como resultado valores literais. Iremos utilizar as expressões literais na atribuição de valor para uma variável ou constante.

Exemplo:

nome=”José da Silva”

nome<-“José da Silva”

media=(nota1+nota2+nota3+nota4)/4

Operadores relacionais

**São expressões compostas por outras expressões ou variáveis numéricas com operadores relacionais. As expressões relacionais retornam valores lógicos (verdadeiro/falso)**

> Maior que

>= Maior ou igual

< Menor que

<= Menor ou igual

== Igualdade

!= Diferente de

Tomadas de decisão

Quando escrevemos programas, geralmente ocorre a necessidade de decidir o que fazer dependendo de alguma condição encontrada durante a execução.

# Como utilizar a concatenação

**Concatenação é um termo usado em computação para designar a operação de unir o conteúdo de duas strings.**

String é uma sequência de caracteres

Agrupamento de duas ou mais células que, incluindo fórmulas, textos ou outras informações contida no seu interior, dá origem a um único resultado.

# Introdução a Portugol

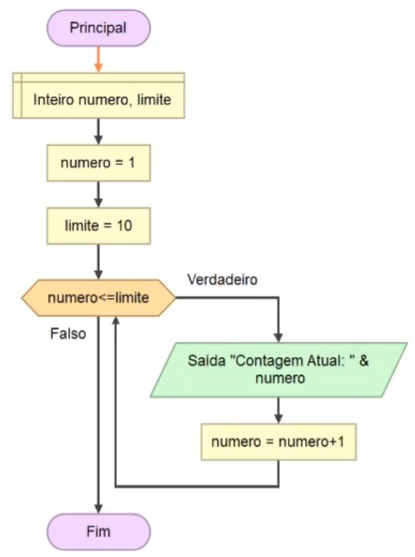
# Aprenda como utilizar uma estrutura de repetição

Objetivos da Aula:

1. Aprender o que é uma estrutura de repetição e como utilizá-la

**Dentro da lógica de programação é uma estrutura que permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador.**

Exemplo: Contar de 0 a 10.



Exercício:

Exibir a tabuada de 9

# O que são linguagens de programação?

Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software). Um software pode ser desenvolvido para rodar em um computador, dispositivo móvel ou em qualquer equipamento que permita sua execução.

O que é óbvio para você, certamente não é óbvio para uma máquina. E se você quer que a máquina faça algo pra você, você precisa, “falar com ela”.

**A função das linguagens de programação é servir de um meio de comunicação entre computadores e humanos.**

Alto nível

Essas são aquelas cuja sintaxe se aproxima mais da nossa linguagem e se distanciam mais da linguagem de máquina.

Baixo nível

É aquela que se aproxima mais da linguagem de máquina. Essas são as que você precisa ter o conhecimento direto de arquitetura do computador para fazer alguma coisa. Exemplo: Assembly

**Compiladas**

É uma linguagem de programação em que o código fonte, é executado diretamente pelo sistema operacional ou pelo processador, após ser traduzido por meio de um processo chamado compilação.

Exemplos: C, C++, Visual Basic, Delph, C#

**Interpretadas**

É uma linguagem de programação em que o código fonte é executado por um programa de computador chamado interpretador, que em seguida é executado pelo sistema operacional ou processador.

Exemplos: JavaScript, PHP e Python

Portugol é uma pseudolinguagem que permite ao leitor desenvolver algoritmos estruturados em português de forma simples e intuitiva, independente de linguagem de programação.

É uma pseudolinguagem que permite ao programador pensar no problema em si e não no equipamento que irá executar o algoritmo.

IDE: Ambiênte gráfico de desenvolvimento.

<https://github.com/UNIVALI-LITE/Portugol-Studio/releases/>

real, inteiro

cadeia, caracter

leia(variável)

“\n” quebra linha

// Comentário

Escolha

Pare

Caso contrario

Exercício:

Fazer um programa para retornar a média de vendas de um vendedor para os meses de janeiro, fevereiro e março. Exibir a média, o nome do vendedor e o total vendido.

escreva("Sua média foi " + media)

# Aprenda a atualizar desvios condicionais e boas práticas em programação

Objetivos:

1) Aprender a utilizar os desvios condicionais (estruturas de decisão) no portugol.

2) Boas práticas de programação – comentários

Desvio condicional – SE

É utilizada a palavra reservada se, a condicional a ser testada entre parênteses e as instruções que devem ser executadas entre chaves caso o desvio seja verdadeiro.

Exemplo:

se(medai >= 7) {

escreva(“Parabéns! Você foi aprovado”)

}

Se – senao

Agora vamos imaginar que se a condição for falsa um conjunto de comandos deve ser executado.

Quando iremos encontrar essa situação?

se(medai >= 7) {

escreva(“Parabéns! Você foi aprovado”)

} senao {

Escreva(“Você foi reprovado.”)

}

// Função: Fazer aquilo

// Autor: Fulano

Desvio Condicional – caso

Este comando é similar aos comandos se e senao, e reduz a complexidade na escolha de diversas opções. Apesar de suas similaridades com o se, ele possui algumas diferenças. Nesse comando não é possível o uso de operadores lógicos, ele apenas trabalha com valores definidos.

Exemplo:

Inteiro valor==0

Escolha(valor)

{

Caso1: // Testar se o valor é igual a 1

Escreva (“Ok! Abrir Netflix!”)

Pare

Caso2: // Testar se o valor é igual a 2

Escreva(“Ok! Abrir Amazon prime”)

Pare

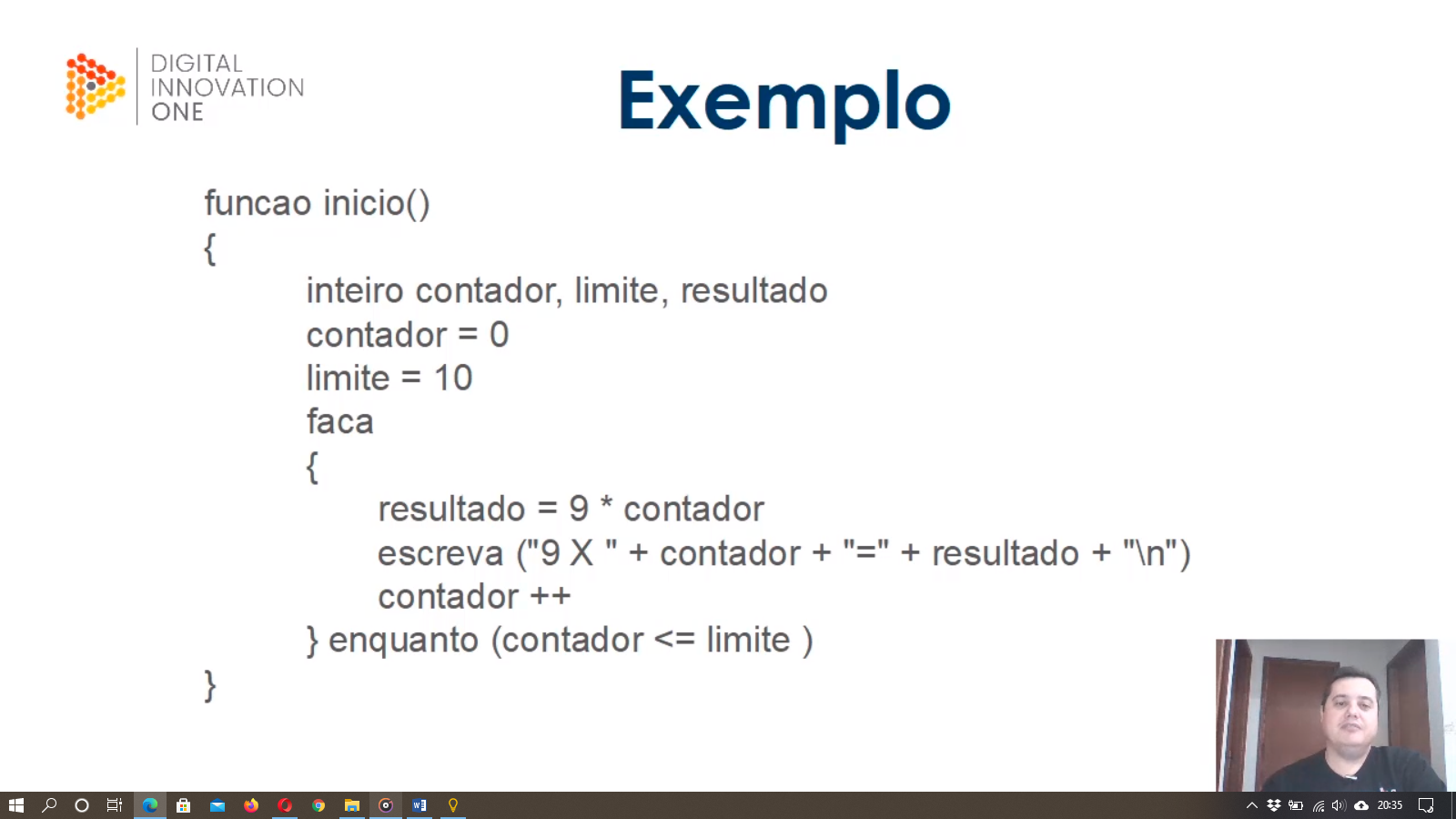
Caso contrario:

Escreva(“Você deve escolher as opções 1 ou 2”)

}

Trabalhando com laços de repetição em Portugol

Dentro da lógica de programação é uma estrutura que permite executar mais de uma vez o mesmo comando ou conjunto de comandos, de acordo com uma condição ou com um contador.



Exercício:

Fazer um programa que execute a tabuada de um número.

programa

{

funcao inicio()

{

inteiro numeroTabuada, i, limite, resultado

i = 0

limite = 10

escreva("Digite o número da tabuada: ") leia(numeroTabuada)

faca {

resultado = numeroTabuada \* i

escreva(numeroTabuada + " x " + i + " = " + resultado + "\n")

i ++

} enquanto (i <= limite)

}

}

# Matrizes e Vetores

**Uma matriz é uma coleção de variáveis do mesmo tipo, acessíveis com um único nome e armazenados contiguamente na memória.**

A individualização de cada variável de um vetor é feita através do uso de **índices**.

Os **vetores** são matrizes de uma só dimensão.

Exercício:

