

Aula 1 - Por que um software é criado?

Chegou na aula 1 ?
Digite “level 1” nos comentários

Aula 1 - Porque um software é criado?

1. Para criar soluções para problemas do dia a dia

Aula 1 - Porque um software é criado?

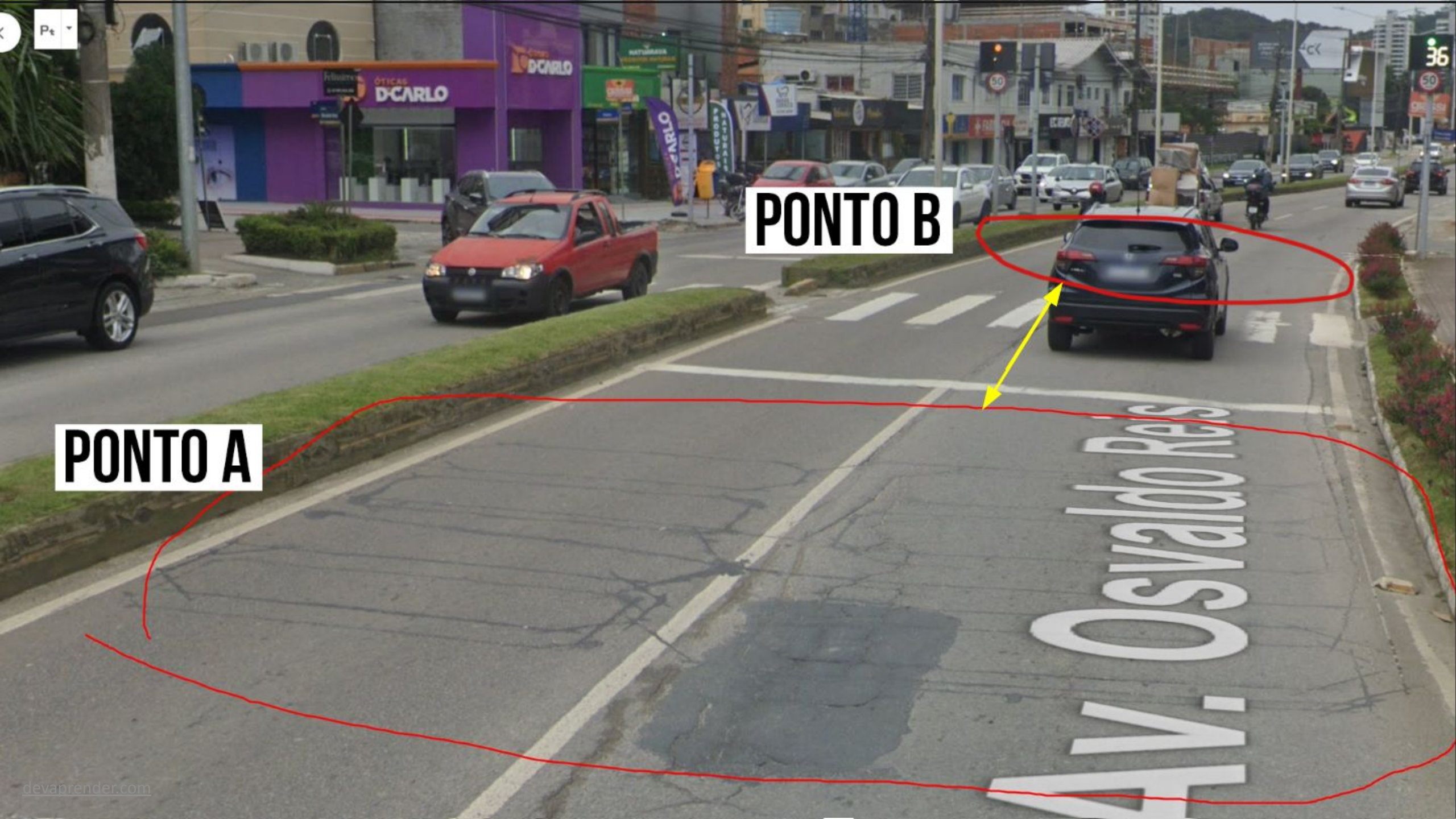
1. Para criar soluções para problemas do dia a dia



Então **todo software** é grandioso assim?

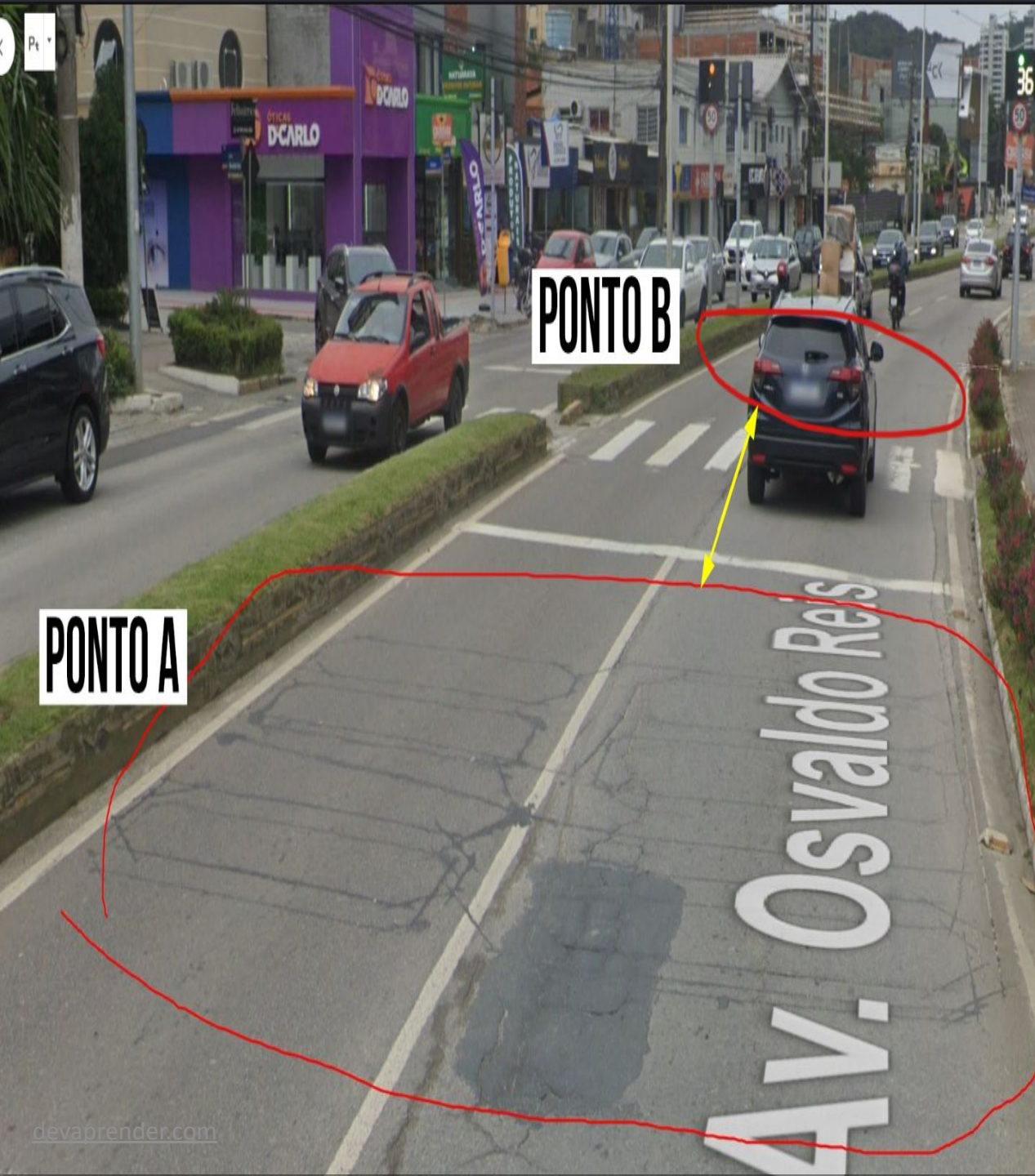
Aula 1 - Porque um software é criado?

- 1. Para criar soluções para problemas do dia a dia**
- 2. Automação e otimização de processos**



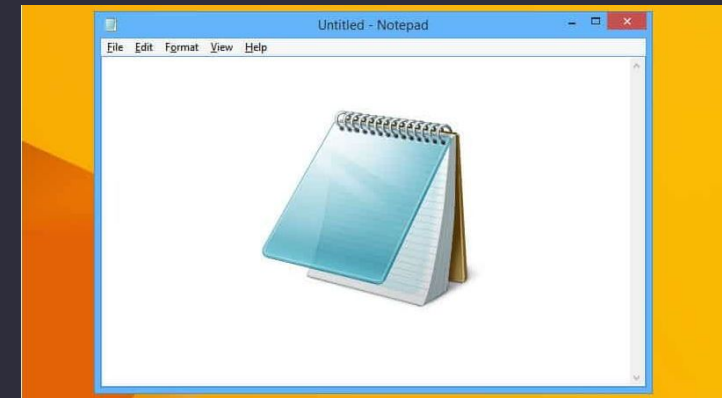
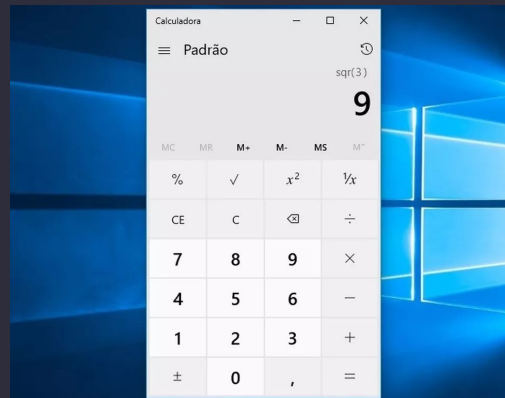
PONTO A

PONTO B



Aula 1 - Porque um software é criado?

1. Para criar soluções para problemas do dia a dia
2. Automação e otimização de processos



2. Automação e otimização de processos

- ✓ Automatizar tarefas na web(interagir, preencher ou buscar informações em sites)
- ✓ Automatizar criação, preenchimento, duplicação, e envio de arquivos(planilhas,docs,textos,pdf, imagens, etc...)
- ✓ Criação, leitura e envio de e-mails
- ✓ Tarefas que use seu teclado e mouse
- ✓ Praticamente qualquer processo que é feito em um dispositivo pode ser automatizado parcial ou totalmente usando a programação

Aula 1 - Porque um software é criado?

1. Para criar soluções para problemas do dia a dia



1. Automação e otimização de processos



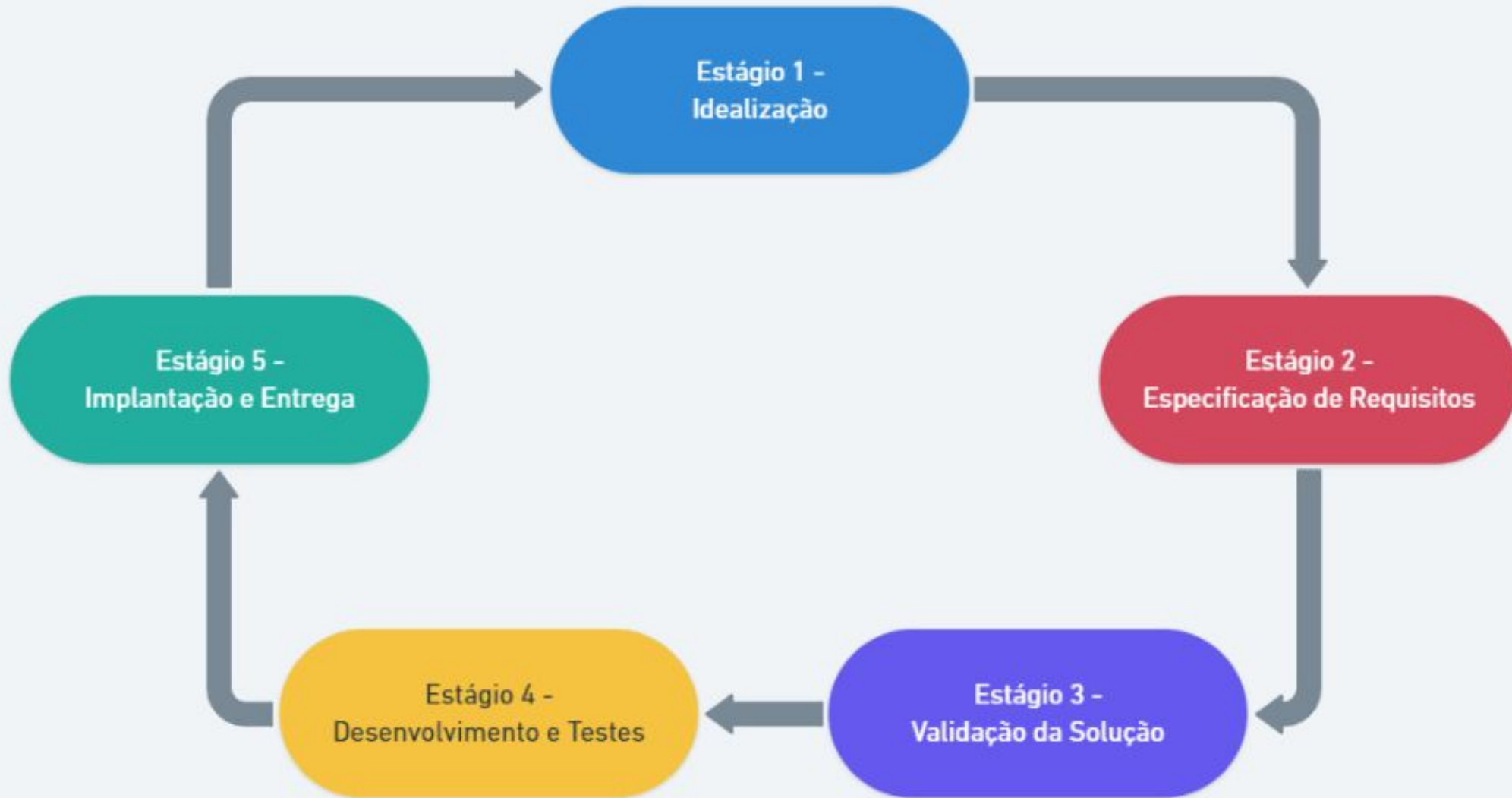
- ✓ Automatizar tarefas na web(interagir, preencher ou buscar informações em sites)
- ✓ Automatizar criação, preenchimento, duplicação, e envio de arquivos(planilhas,docs,textos,pdf, imagens, etc...)
- ✓ Criação, leitura e envio de e-mails
- ✓ Tarefas que use seu teclado e mouse
- ✓ Praticamente qualquer processo que é feito em um dispositivo pode ser automatizado parcial ou totalmente usando a programação

Aula 2 - Como software é criado do zero e onde a lógica encaixa nisso?

“Antes de existir a lógica, havia um processo... um ciclo!”

-Jhonatan de Souza

Ciclo de Desenvolvimento de Software



Aula 2 - Como software é criado do zero e onde a lógica encaixa nisso?

Agora **você conhece o ciclo!**
Digite **“Conheço o ciclo!”**

Aula 3 - O problema que todo iniciante enfrenta

“Me solta! Deixa eu começar a codar!”

-Todo iniciante inexperiente

Erro #1 de quase todo iniciante

Todos passam por isso:

- **Normal #1** - vai escrever código lentamente
- **Normal #2** - Vai resolver os mesmos problemas em situações diferentes
- **Normal #3** - Vai se achar incapaz/insuficiente
- **Normal #4** - Vai gradualmente conseguir solucionar problemas mais facilmente

“Tudo começa na análise, na análise crítica do problema”

-Jhonatan de Souza

Aula 4 - Aprenda resolver problemas através da análise crítica

Chegou na aula 4?
Digite “Sou nível 4”

Aula 4 - Aprenda resolver problemas através da análise crítica

“Saber programar é secundário o importante é saber criar soluções bem definidas!”

-Jhonatan de Souza

Aula 4 - Aprenda resolver problemas através da análise crítica

“Saber analisar e resolver um problema é mais importante que decorar os comandos de uma linguagem de programação!”

-Jhonatan de Souza

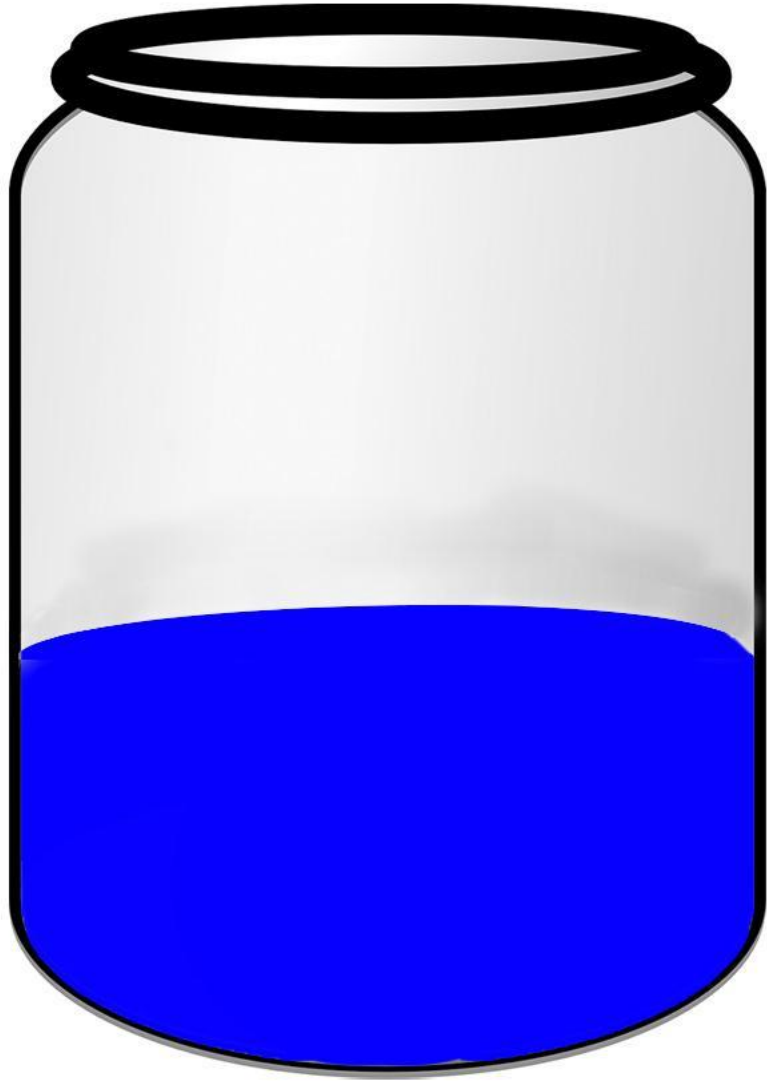


5 ml

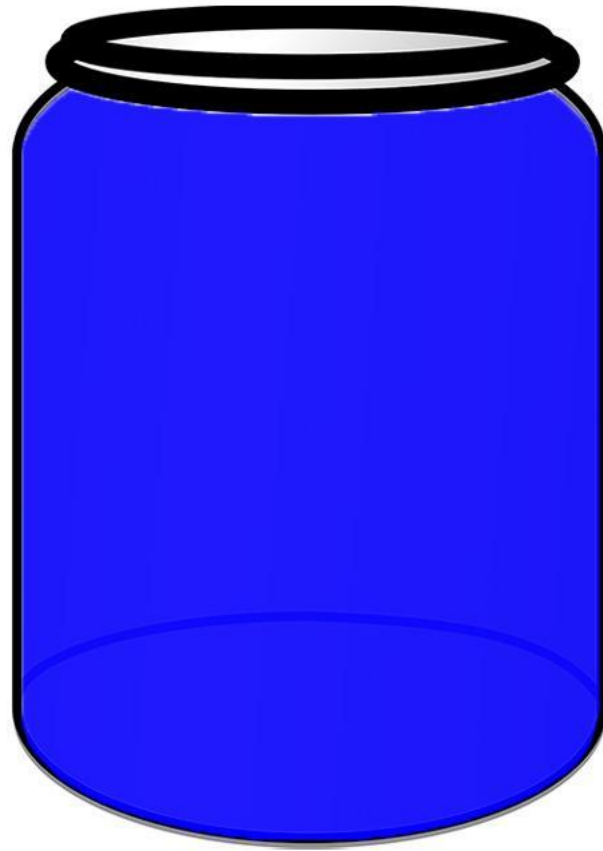


3 ml

***objetivo:
2 ml***

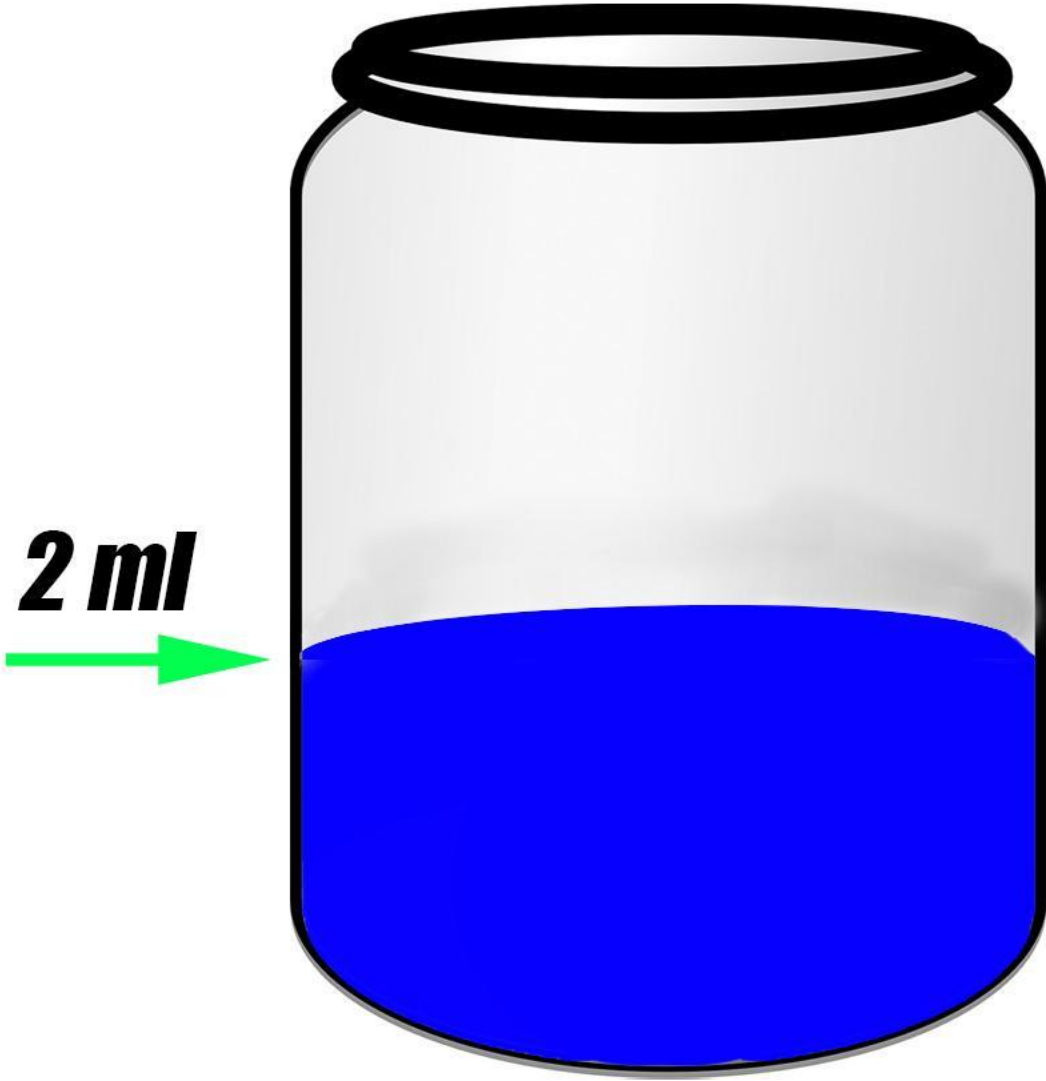


5 ml

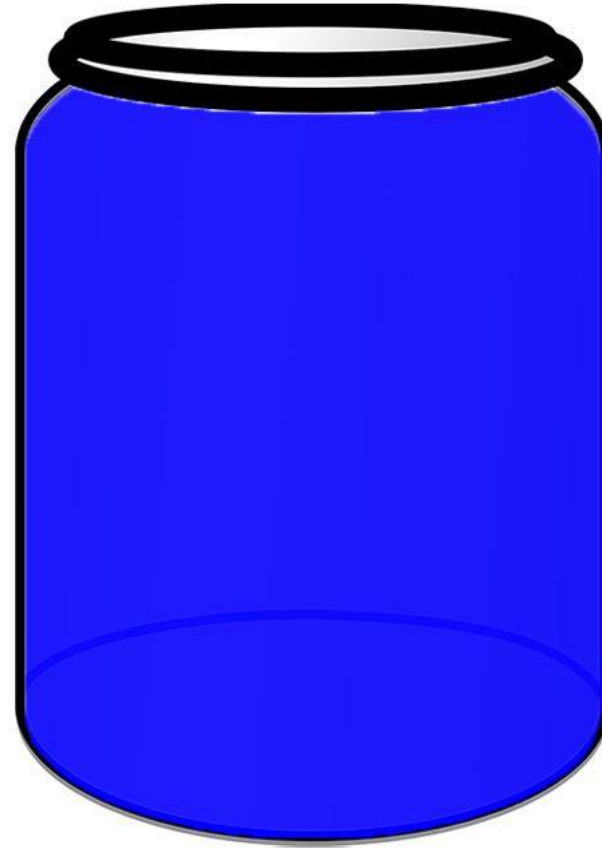


3 ml

***objetivo:
2 ml***



5 ml



3 ml

***objetivo:
2 ml***

“Quanto mais experiente se tornar, mais óbvio ficam os caminhos que levam a uma solução”

-Jhonatan de Souza

Problema #2 - Meu salário por hora

- Escreva um programa que retorne o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

Problema #2 - Meu salário por hora

- Escreva um programa que retorne o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.
- Temos que fazer o seguinte cálculo:
 $\text{salário mensal} / \text{total de horas trabalhadas}$

Solução

- salário mensal: R\$2500,00
- horas trabalhadas: 176 horas
- $2500/176 = \text{R\$}14.20$
- salário hora = R\$14.20

“Resolvi esse problema antes de escrever uma linha de código!”

- Iniciante ao perceber que código é secundário!**



Na próxima aula...

Está curtindo essas aulas?

Aula 5 - O que são algoritmos e como montar algoritmos do zero

Aula 5 - O que são algoritmos e como montar algoritmos do zero

1. O que são algoritmos?

“Um algoritmo é simplesmente uma série de instruções a serem seguidas, para resolver um problema.”

1. Quando algoritmos devem ser criados?

Sempre que queremos montar uma sequência de passos necessários para solucionar um problema

3. Qual é a estratégia para montar um algoritmo?

Independente se:

1. Quando alguém te apresenta um problema a ser resolvido.

2. Quando você encontra um problema a ser resolvido.

Método 5Q's para montar um algoritmo:

Analise criticamente o problema e descubra:

(Tente explicar este problema para você mesmo em voz alta e peça mais informações/investigue mais até você compreender completamente o problema.)

- 1. Quais são os dados de entrada necessários?**
- 2. O que devo fazer com estes dados?**
- 3. Quais são as restrições deste problema?**
- 4. Qual é o resultado esperado?**
- 5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado?**

Vamos aplicar esse método a um problema real!

Problema 1 - Ligar para alguém

**Monte um algoritmo necessário para
ligar para um amigo**

Problema 1 - Ligar para alguém

1. Quais são os dados de entrada necessários?

**Monte um algoritmo necessário para
ligar para um amigo**

- um telefone
- um número de celular

Problema 1 - Ligar para alguém

2. O que devo fazer com estes dados?

**Monte um algoritmo necessário para
ligar para um amigo**

- devo usar o celular para discar o número do meu amigo

Problema 1 - Ligar para alguém

3. Quais são as restrições deste problema?

**Monte um algoritmo necessário para
ligar para um amigo**

- Caso meu amigo não atenda, devo deixar uma mensagem dizendo: “me ligue de volta”

Problema 1 - Ligar para alguém

4. Qual é o resultado esperado?

**Monte um algoritmo necessário para
ligar para um amigo**

- Conseguir falar com meu amigo

Problema 1 - Ligar para alguém

5. Qual é sequência de passos a serem feitas para chegar ao resultado esperado?

Monte um algoritmo necessário para ligar para um amigo

1. Pegar o telefone
2. se estiver travado por senha, destrave o celular
3. verifique se há sinal da operadora
4. navegue até o discador do celular
5. digite o número do seu amigo
6. aperte no botão de "ligar"
7. aguarde a ligação completar
8. se a ligação completar, conversar com ele
9. se a ligação não completar, deixar uma mensagem dizendo "me ligue de volta"

Este processo faz sentido?

 **Compartilhe sua opinião abaixo**  :

Problema 2 - Valor por hora

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

Problema 1 - Valor por hora

1. Quais são os dados de entrada necessários?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- salário mensal
- horas trabalhadas por mês

Problema 1 - Valor por hora

2. O que devo fazer com estes dados?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- devo usar eles para calcular o valor hora que um funcionário recebe usando o cálculo $\text{salário mensal} / \text{horas trabalhadas}$

Problema 1 - Valor por hora

3. Quais são as restrições deste problema?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- Os valores devem ser entregues somente em formato de salário por hora

Problema 1 - Valor por hora

4. Qual é o resultado esperado?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- O valor hora que um funcionário recebe

Problema 1 - Valor por hora

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- perguntar quanto a pessoa ganha por mês
- guardar essa informação
- pergunta quantas horas ela trabalha por mês
- guardar essa informação
- calcular o valor hora da pessoa (salário mensal / horas trabalhadas)
- exibir o valor hora daquela pessoa

Este começando a fazer sentido?

 **Compartilhe sua opinião abaixo**  :

Problema 3 - Chute o número

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima, abaixo ou igual ao valor aleatório gerado no início do programa.

Problema 3 - Chute o número

1. Quais são os dados de entrada necessários?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- um valor aleatório de 1 a 10
- um chute do usuário

Problema 3 - Chute o número

2. O que devo fazer com estes dados?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- Devo pegar o valor aleatório de 1 a 10 que foi gerado e comparar com o valor que foi chutado pelo usuário.
- Se o chute for maior ou menor que valor gerado, alertar o usuário sobre isso e o deixar jogar novamente até que acerte o número que foi gerado.

Problema 3 - Chute o número

3. Quais são as restrições deste problema?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- o programa não deve ser finalizado até que um valor seja chutado corretamente.
- O usuário deverá ser capaz de jogar quantas vezes quiser.

Problema 3 - Chute o número

4. Qual é o resultado esperado?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- O programa identificar que o valor chutado pelo usuário é igual ao valor gerado no início do programa.

Problema 3 - Chute o número

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima ou abaixo do valor aleatório gerado no início do programa.

1. gerar um valor aleatório de 1 a 10.
2. armazenar este valor.
3. receber o chute do usuário.
4. comparar o chute do usuário com o valor aleatório gerado.
5. se o chute for maior que o valor gerado, exibir “chutou alto” e voltar ao passo 3.
6. se o chute for menor que o valor gerado, exibir “chutou baixo” e voltar ao passo 3.
7. se o chute for igual que o valor gerado, exibir “acertou!” e voltar ao passo 1.

Entendeu o processo dos 5Q's ?

👉 Deixe sua opinião abaixo 👉 :

Aula 06 - 4 Conceitos OBRIGATÓRIOS ser capaz de resolver problemas!

 Está gostando dessa aula?

Compartilhe ela com pelo menos 1
pessoa hoje 

Aula 06 - 4 Conceitos OBRIGATÓRIOS ser capaz de resolver problemas!

- 1. Variáveis**
- 2. Condicionais**
- 3. Laços de repetição**
- 4. Coleções**

Aula 06 - 4 Conceitos OBRIGATÓRIOS ser capaz de resolver problemas!

Variáveis

- Quando você ouve o nome de alguém, o que você faz com este nome?
- Você tenta guardar ele certo?
- Onde você guarda ele? Você guarda ele na sua memória.
- Quando alguém te passa uma data de uma consulta, o que você faz com essa data?
- Você anota ela em algum lugar para se lembrar daquilo ou simplesmente tenta guardar aquela informação na sua mente certo?
- Quando alguém te diz para ir ao supermercado e comprar 1 barra de chocolate, você guarda ou anota essa informação em algum lugar também certo?

Variáveis

- Quando você ouve o nome de alguém, o que você faz com este nome?
- Você tenta guardar ele certo?
- Onde você guarda ele? Você guarda ele na sua memória.
- Quando alguém te passa uma data de uma consulta, o que você faz com essa data?
- Você anota ela em algum lugar para se lembrar daquilo ou simplesmente tenta guardar aquela informação na sua mente certo?
- Quando alguém te diz para ir ao supermercado e comprar 1 barra de chocolate, você guarda ou anota essa informação em algum lugar também certo?

O que essas situações têm em comum?

Variáveis

Problema 2 - Valor por hora

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

Variáveis

Problema 2 - Valor por hora

Escreva um programa que retorna o **valor hora** de um funcionário com base no seu **salário mensal** e **horas trabalhadas por mês**.

- **salário mensal**
- **horas trabalhadas por mês**
- **valor hora**

Condicionais

- Quando alguém te oferece um panfleto na rua você pega ele sem pensar?
- Quando descobre que algo que você quer muito comprar está na promoção, você compra ela direto sem pensar antes?

Condicionais

Problema 3 - Chute o número

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima, abaixo ou igual ao valor aleatório gerado no início do programa.

Condicionais

Problema 3 - Chute o número

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente.

O programa deve informar se o chute foi acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- Neste caso podemos identificar algumas condicionais:
- se o número chutado for menor que o número gerado, então avise que deve chutar mais alto
- se o número chutado for maior que o número gerado, então avise que deve chutar mais baixo
- se o número chutado for igual que o número gerado, então avise que o ele acertou

Laços de repetição

No mundo real e na programação, temos diversas situações onde precisamos fazer uma ação várias vezes ou por uma determinada quantidade de vezes. Ex:

- Cortar um pão
- Varrer o chão
- Dar vários passos para chegar até um lugar

Laços de repetição

Cenário real:

- Vamos dizer que para derrubar uma árvore, tenha que dar 11 machadas nela.
- Então você teria que criar um comando que mande a pessoa dar 11 machadadas na arvore e com isso feito a árvore cairia.

Laços de repetição

Cenário real:

- Então no exemplo anterior, temos uma repetição que é executada por 11 vezes para conseguir derrubar a árvore.
- Esse é um exemplo de repetição usada na lógica de programação dentro de jogos.

Laços de repetição

Cenário real #2

- Crie um programa que recebe um número e incrementa ele por 1 por 10 vezes

Laços e repetição

Cenário real #2

Crie um programa que recebe um número e o incrementa por 1 por 10 vezes.

- valor = 10
- 1° repetição | valor = valor + 1 | 11
- 2° repetição | valor = valor + 1 | 12
- 3° repetição | valor = valor + 1 | 13
- 4° repetição | valor = valor + 1 | 14
- 5° repetição | valor = valor + 1 | 15
- 6° repetição | valor = valor + 1 | 16
- 7° repetição | valor = valor + 1 | 17
- 8° repetição | valor = valor + 1 | 18
- 9° repetição | valor = valor + 1 | 19
- 10° repetição | valor = valor + 1 | 20

Laços de repetição

Ponto em comum

Normalmente possuem um ponto inicial e um ponto final, ou em alguns casos uma condição que irá significar o final daquele laço de repetição, ex:

- Deixe a torneira aberta até que o copo esteja cheio de água
- Nade até chegar ao outro lado da piscina
- Some todos os valores até enquanto o valor for menor que 100
- Dê machadadas na árvore até que ela caia

Laços de repetição

pontos chaves

- Úteis para conseguir automatizar processos
- É possível definir por quantas vezes algo deve ser executado
- Possuem ponto de início e fim
- Podem estar atrelados a uma condição

Coleções

Em muitos casos, você terá a necessidade de trabalhar com coleções de valores que estão armazenados em um local só, ex:

- convidados = ['Jeff', 'Amanda', 'Carol', 'Robert']
- numeros_premiados = [12,67,34,100,55]

Coleções

Exemplo de uso de coleções - Chute o número

Dado uma coleção de dados "idades" [15,46,75,34,23] imprima na tela a soma destes valores.

- Definir o total como 0
- Passar por 1º valor e somar ele ao total
- Passar por 2º valor e somar ele ao total
- Passar por 3º valor e somar ele ao total
- Passar por 4º valor e somar ele ao total
- Passar por 5º valor e somar ele ao total
- Exibir o valor total

Classes

Em alguns casos, quando você percebe que está criando **inúmeras variáveis** e **funções** que tratam da **mesma funcionalidade**, talvez seja a hora de criar uma classe!

Classes

Exemplo de uso de classe - Classe Gerador de Senha

- Classe **GeradorDeSenha**


Variáveis

- quantidade_de_caracteres
- usar_numeros
- usar_letras
- usar_caracteres_especiais

Funções

- gerar_senha

**“Dominar estes 4 conceitos te permitirá
criar algoritmos muito mais facilmente!”**

 Agora que tem uma noção dos conceitos mais importantes, podemos criar nossos próprios algoritmos através das estratégias que irei te mostrar nas próximas 2 aulas.

Está curtindo este curso? Então envie ele para pelo menos mais um conhecido 

Aula 7 - Criando soluções em Pseudocódigo do básico ao avançado

Aula 7 - Criando soluções em Pseudocódigo do básico ao avançado

1. Como podemos demonstrar a solução para um problema?

Aula 7 - Criando soluções em Pseudocódigo do básico ao avançado

1. Pseudocódigo

2. Fluxogramas

- Qual é a melhor? / Qual devo usar?

O pseudocódigo:

- O pseudocódigo é uma descrição dos passos necessários para resolver um determinado problema em uma linguagem natural, que não está ligada diretamente a uma linguagem de programação

alertas sobre o pseudocódigo

- Pseudocódigo não é uma linguagem de programação
- A ideia todo do pseudocódigo é de escrever algo em uma linguagem natural que pode ser lida por qualquer pessoa.
- Pseudocódigo não é padronizado, pode ser escritos de várias maneiras. Não há um padrão certo ou errado de o escrever.
- Não será processado por um computador
- Não substitui uma linguagem de programação real
- Deve ser usado apenas para criar uma lógica para resolver um problema e depois escrito em uma linguagem de programação real como Python, Javascript, C#, ou Java
- Não vale a pena na minha visão se tornar um especialista em pseudocódigo ou em linguagens de "aprendizado" como a linguagem de programação "portugol".

Exemplo #1 - Pseudocódigo

(com uso de variáveis)

Valor por hora

1. Crie um programa que retorne o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

Método 5Q's para montar um algoritmo:

Analise criticamente o problema e descubra:

(Tente explicar este problema para você mesmo em voz alta e peça mais informações/investigue mais até você compreender completamente o problema.)

1. Quais são os dados de entrada necessários?
2. O que devo fazer com estes dados?
3. Quais são as restrições deste problema?
4. Qual é o resultado esperado?
5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado? **(faça essa parte em pseudocódigo)**

Problema 1 - Valor por hora

1. Quais são os dados de entrada necessários?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- salário mensal
- horas trabalhadas por mês

Problema 1 - Valor por hora

2. O que devo fazer com estes dados?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- devo usar eles para calcular o valor hora que um funcionário recebe usando o cálculo $\text{salário mensal} / \text{horas trabalhadas}$

Problema 1 - Valor por hora

3. Quais são as restrições deste problema?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- Os valores devem ser entregues somente em formato de salário por hora

Problema 1 - Valor por hora

4. Qual é o resultado esperado?

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

- O valor hora que um funcionário recebe

Problema 1 - Valor por hora

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado?(faça essa parte em pseudocódigo)

Escreva um programa que retorna o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.

Pseudocódigo sintaxe

- **input:** palavra usada para receber dados do usuário
- **print:** exibir o resultado no console
- **if condição:** condicional que controla se algo deve ou não ser feito
- **else:** cláusula a ser executada caso nenhuma condicional if seja executada
- **loop de X a Y:** laço de repetição que irá iterar de X a Y
- **loop X em Y:** laço de repetição que irá iterar X em uma coleção Y
- **while X:** laço de repetição que acontecerá enquanto um condição for verdadeira

Problema 1 - Valor por hora

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado? (faça essa parte em pseudocódigo)

- **input:** palavra usada para receber dados do usuário
- **print:** exibir algo
- **if condição:** condicional que controla se algo deve ou não ser feito
- **else:** cláusula a ser executada caso nenhuma condicional if seja executada
- **loop de X a Y:** laço de repetição que irá iterar de X a Y
- **loop X em Y:** laço de repetição que irá iterar X em uma coleção Y
- **while X:** laço de repetição que acontecerá enquanto um condição for verdadeira

```
input salario_mensal
input horas_trabalhadas_por_mes
valor_hora = salario_mensal /
horas_trabalhadas_por_mes
print valor_hora
```

Exemplo #2 - Pseudocódigo (com uso de variáveis)

Gerador de boas-vindas

Crie um programa que pede ao usuário seu nome e depois o dá as boas-vindas dizendo "X seja bem-vindo", onde X é o nome do usuário

Problema 2 - Gerador de boas vindas

1. Quais são os dados de entrada necessários?

Crie um programa que pede ao usuário seu nome e depois o dá as boas-vindas dizendo "X seja bem-vindo", onde X é o nome do usuário

- nome do usuário

Problema 2 - Gerador de boa vindas

2. O que devo fazer com estes dados?

Crie um programa que pede ao usuário seu nome e depois o dá as boas-vindas dizendo "X seja bem-vindo", onde X é o nome do usuário

- após receber o nome do usuário, eu devo exibir uma mensagem que diz nome do usuário + seja bem vindo.

Problema 2 - Gerador de boas vindas

3. Quais são as restrições deste problema?

Crie um programa que pede ao usuário seu nome e depois o dá as boas-vindas dizendo "X seja bem-vindo", onde X é o nome do usuário

- Devo possuir um nome para que possa exibir quem estou apresentando

Problema 2 - Gerador de boa vindas

4. Qual é o resultado esperado?

Crie um programa que pede ao usuário seu nome e depois o dá as boas-vindas dizendo "X seja bem-vindo", onde X é o nome do usuário

- uma mensagem que exibe o nome do usuário + a mensagem "seja bem-vindo"

Problema 2 - Gerador de boa vindas

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado? (faça essa parte em pseudocódigo)

- **input:** palavra usada para receber dados do usuário
- **print:** exibir algo
- **if condição:** condicional que controla se algo deve ou não ser feito
- **else:** cláusula a ser executada caso nenhuma condicional if seja executada
- **loop de X a Y:** laço de repetição que irá iterar de X a Y
- **loop X em Y:** laço de repetição que irá iterar X em uma coleção Y
- **while X:** laço de repetição que acontecerá enquanto um condição for verdadeira

```
input nome_do_usuario  
print nome_do_usuario + "seja bem-vindo"
```


Exemplo #3 - Pseudocódigo

(exemplo de uso de variáveis)

(exemplo de uso de condicionais)

Exibir o maior dos dois números

Crie um programa que recebe dois valores e exibe qual é o maior entre eles

Problema 3 - Exibir o maior dos dois números

1. Quais são os dados de entrada necessários?

Crie um programa que recebe dois valores e exibe qual é o maior entre eles

- primeiro valor
- segundo valor

Problema 3 - Exibir o maior dos dois números

2. O que devo fazer com estes dados?

Crie um programa que recebe dois valores e exibe qual é o maior entre eles

- após receber os dois valores, devo os comparar e exibir qual é o maior entre eles

Problema 3 - Exibir o maior dos dois números

3. Quais são as restrições deste problema?

Crie um programa que recebe dois valores e exibe qual é o maior entre eles

- Eu devo possuir 2 números para que a comparação seja realizada

Problema 3 - Exibir o maior dos dois números

4. Qual é o resultado esperado?

Crie um programa que recebe dois valores e exibe qual é o maior entre eles

- exibir na tela o maior valor entre os dois

Problema 3 - Exibir o maior dos dois números

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado? (faça essa parte em pseudocódigo)

- **input:** palavra usada para receber dados do usuário
- **print:** exibir algo
- **if condição:** condicional que controla se algo deve ou não ser feito
- **else:** cláusula a ser executada caso nenhuma condicional if seja executada
- **loop de X a Y:** laço de repetição que irá iterar de X a Y
- **loop X em Y:** laço de repetição que irá iterar X em uma coleção Y
- **while X:** laço de repetição que acontecerá enquanto um condição for verdadeira

```
input primeiro_valor
input segundo_valor
if primeiro_valor > segundo_valor
    print "o primeiro valor é maior"
if primeiro_valor < segundo_valor
    print "o segundo valor é maior"
else
    print "os números são iguais"
```

Exemplo #4 - Pseudocódigo

(exemplo de uso de variáveis)

(exemplo de uso de laço de coleções)

(exemplo de uso e laços de repetição)

Fatorial de um número

Crie um programa que recebe um número e imprime seu fatorial.

Problema 4 - Fatorial de um número

1. Quais são os dados de entrada necessários?

Crie um programa que recebe um número e imprime seu fatorial.

- numero

Problema 4 - Fatorial de um número

2. O que devo fazer com estes dados?

Crie um programa que recebe um número e imprime seu fatorial.

- após receber um número devo o multiplicar por todos seus antecessores. Feito isso devo pegar este resultado e o exibir.

Problema 4 - Fatorial de um número

3. Quais são as restrições deste problema?

Crie um programa que recebe um número e imprime seu fatorial.

- o número deve ser um numero inteiro
- o número deve ser positivo

Problema 4 - Fatorial de um número

4. Qual é o resultado esperado?

Crie um programa que recebe um número e imprime seu fatorial.

- é esperado que o fatorial do número recebido seja exibido.

Problema 4 - Fatorial de um número

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado?(faça essa parte em pseudocódigo)

- **input:** palavra usada para receber dados do usuário
- **print:** exibir algo
- **if condição:** condicional que controla se algo deve ou não ser feito
- **else:** cláusula a ser executada caso nenhuma condicional if seja executada
- **loop de X a Y:** laço de repetição que irá iterar de X a Y
- **loop X em Y:** laço de repetição que irá iterar X em uma coleção Y
- **while X:** laço de repetição que acontecerá enquanto um condição for verdadeira

```
input numero
if numero < 0
    print "digite apenas números positivos"
fatorial = 1
loop de 1 a numero
    fatorial = fatorial * numero
print fatorial
```

número = 5	
fatorial = 1	
fatorial = 1	1 * 1
fatorial = 2	1 * 2
fatorial = 6	2 * 3
fatorial = 24	6 * 4
fatorial = 120	24 * 5

Exemplo #5 - Pseudocódigo

(exemplo de uso de variáveis)

(exemplo de uso de laço de coleções)

(exemplo de uso de laços de repetição)

Some os valores de uma lista

Dados uma coleção de dados "idades"
[15,46,75,34,23] imprima na tela a soma
destes valores

Problema 5 - Some os valores de uma lista

1. Quais são os dados de entrada necessários?

**Dados uma coleção de dados "idades"
[15,46,75,34,23] imprima na tela a soma
destes valores**

- a lista de idades

Problema 5 - Some os valores de uma lista

2. O que devo fazer com estes dados?

Dados uma coleção de dados "idades" [15,46,75,34,23] imprima na tela a soma destes valores

- Após receber a lista de idades, devo somar cada idade com a idade anterior até chegar ao final da lista e exibir a soma de todas as idades.

Problema 5 - Some os valores de uma lista

3. Quais são as restrições deste problema?

Dados uma coleção de dados "idades" (15,46,75,34,23) imprima na tela a soma destes valores

- apenas os valores contidos na lista devem ser adicionados

Problema 5 - Some os valores de uma lista

4. Qual é o resultado esperado?

Dados uma coleção de dados "idades" [15,46,75,34,23] imprima na tela a soma destes valores.

- é esperado que a soma de todas as idades seja exibido.

Problema 5 - Some os valores de uma lista

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado?(faça essa parte em pseudocódigo)

- **input:** palavra usada para receber dados do usuário
- **print:** exibir algo
- **if condição:** condicional que controla se algo deve ou não ser feito
- **else:** cláusula a ser executada caso nenhuma condicional if seja executada
- **loop de X a Y:** laço de repetição que irá iterar de X a Y
- **loop X em Y:** laço de repetição que irá iterar X em uma coleção Y
- **while X:** laço de repetição que acontecerá enquanto um condição for verdadeira

```
idades = [15,46,75,34,23]
total = 0
loop idade em idades
    total = total + idade
print total
```

total = 0	
total = 15	total + 15
total = 61	total + 46
total = 136	total + 75
total = 170	total + 34
total = 193	total + 23

Exemplo #6 - Pseudocódigo

(exemplo de uso de variáveis)

(exemplo de uso de laços de repetição)

(exemplo de uso de condicionais)

Chute o número

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente. O programa deve informar caso o chute tenha sido acima, abaixo ou igual ao valor aleatório gerado no início do programa.

Problema 6 - Chute o número

1. Quais são os dados de entrada necessários?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente. O programa deve informar caso o chute tenha sido acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- um valor aleatório de 1 a 10
- um valor de chute do usuário

Problema 6 - Chute o número

2. O que devo fazer com estes dados?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente. O programa deve informar caso o chute tenha sido acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- após ter gerado e armazenado o valor aleatório, devo pedir que o usuário chute um número. Ao receber o chute do usuário deve comparar se o chute é maior, menor ou igual ao valor gerado e exibir a mensagem correspondente a comparação. (“chute é maior que o valor gerado”, “chute é menor que o valor gerado”, “acertou o chute”)

Problema 6 - Chute o número

3. Quais são as restrições deste problema?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente. O programa deve informar caso o chute tenha sido acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- o valor aleatório deve estar entre 1 e 10

Problema 6 - Chute o número

4. Qual é o resultado esperado?

Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até que o valor aleatório gerado no início do programa seja chutado corretamente. O programa deve informar caso o chute tenha sido acima, abaixo ou igual ao valor aleatório gerado no início do programa.

- Exibir na tela 1 de 3 possíveis mensagens: “Chute é maior que o valor gerado”, “Chute é menor que o valor gerado”, “Acertou o chute!”

Problema 6 - Chute o número

5. Qual é sequência de passos a ser feitas para chegar ao resultado esperado?(faça essa parte em pseudocódigo)

- **input:** palavra usada para receber dados do usuário
- **print:** exibir algo
- **if condição:** condicional que controla se algo deve ou não ser feito
- **else:** cláusula a ser executada caso nenhuma condicional if seja executada
- **loop de X a Y:** laço de repetição que irá iterar de X a Y
- **loop X em Y:** laço de repetição que irá iterar X em uma coleção Y
- **while X:** laço de repetição que acontecerá enquanto um condição for verdadeira

```
input valor_aleatorio entre 1 a 10
acertou = falso
while acertou = falso
    input chute
    if chute > valor_aleatorio
        print "Chute é maior que valor gerado"
    if chute < valor_aleatorio
        print "Chute é menor que valor gerado"
    if chute = valor_aleatorio
        print 'Acertou o chute!'
        acertou = verdadeiro
    else
        print 'Você chutou um valor inválido'
```


2 avisos para você que chegou aqui:

- Não existe pseudocódigo perfeito
- É absolutamente normal achar difícil no 1º contato
- Iremos ver 3 exemplos reais em Python

Aula 8 - Criando soluções com Fluxogramas

Aula 8 - Criando soluções com Fluxogramas

- Fluxogramas - uma representação visual gráfica de um algoritmo.

Símbolos contidos em um Fluxograma

Linha de Fluxo



Indica o fluxo da lógica ao se conectar com os outros símbolos.

Início/Fim



Representa o início ou fim do fluxograma.

Entrada/saída



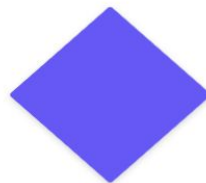
Usado para receber entrada ou saída de dados.

Processamento



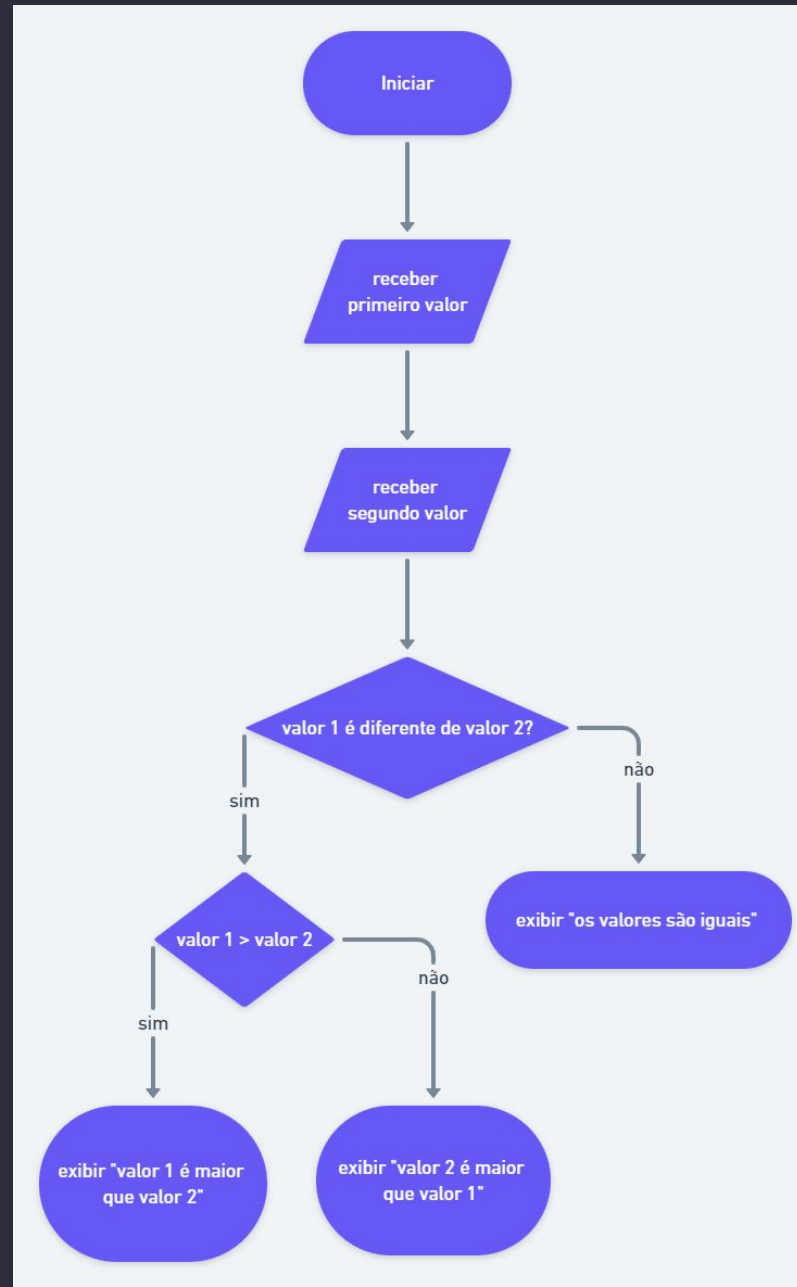
Usado para operações matemáticas ou manipulação de dados

Decisão



Usado para tomada de decisão entre duas ou mais alternativas

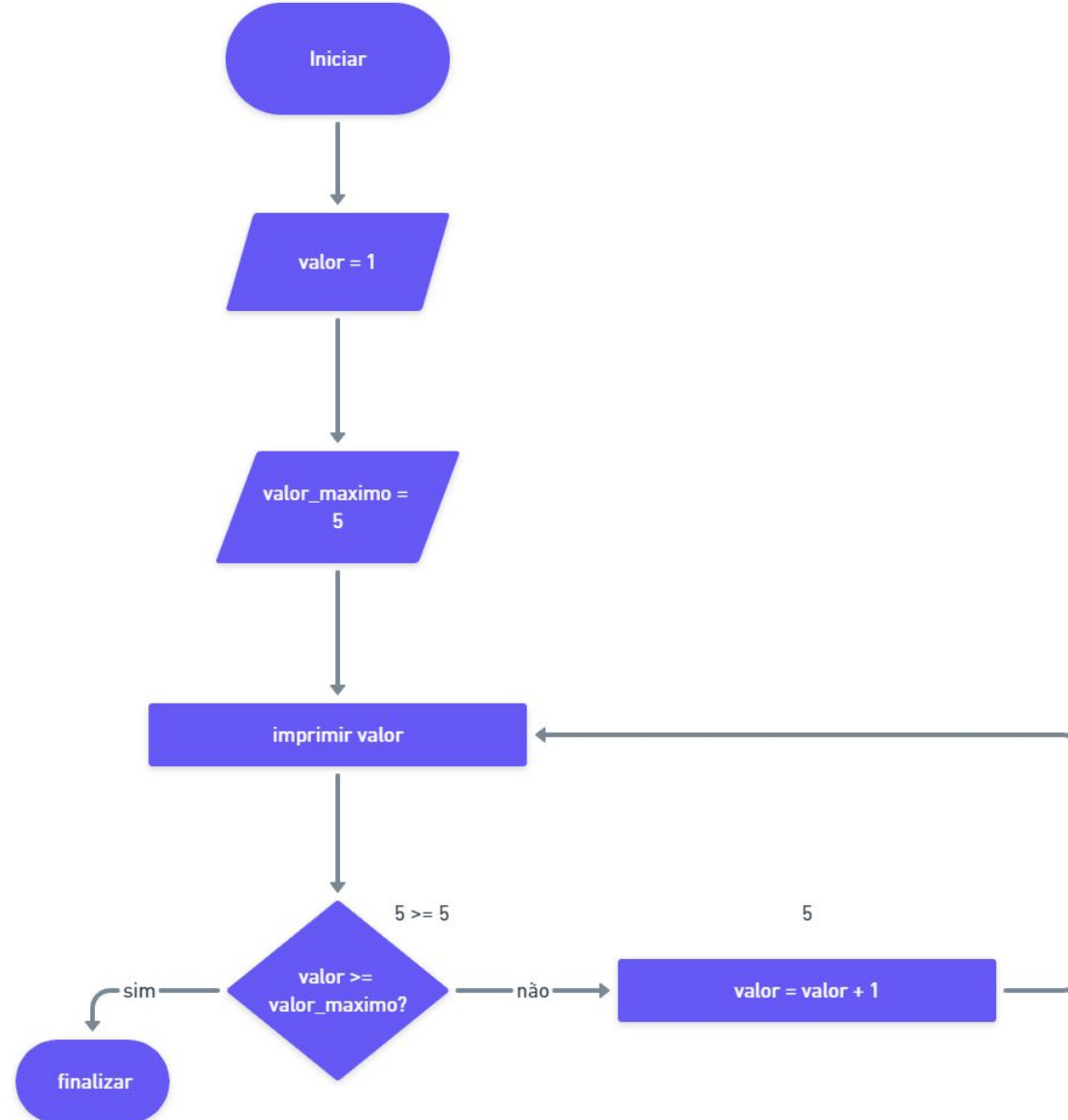
Problema 1 - Exibir o maior dos dois números



Problema 2 - De 1 a N

Exiba na tela todos os números de 1 a N

Problema 2 - De 1 a N



Vantagens e Desvantagens de Fluxogramas

Vantagens	Desvantagens
✧ Fluxogramas podem ser mais intuitivos de entender para algumas pessoas.	✧ Caso tenha um processo altamente complexo, tentar o quebrar em passos em um fluxograma pode levar muito mais tempo que o pseudocódigo.
✧ Fluxogramas podem abrir caminho para que pessoas não técnicas entrem no processo de design da solução.	✧ O nível de detalhe da informação que é colocada em cada campo pode ter que ser simplificada para deixar o fluxograma mais legível.
✧ Caso tenha um processo altamente complexo, tentar o quebrar em passos menores em um fluxograma pode deixar a lógica mais explícita e de fácil compreensão.	✧ Ao tentar representar a lógica de um programa maior, terá um fluxograma possivelmente gigantesco e que não é facilmente lido, a não ser que tenha múltiplos monitores ou imprima o fluxograma em tamanhos grandes o suficiente para serem lidos em uma mesa no trabalho, por exemplo.

Só existem 2 estados na mente de um programador:

**“mas como eu vou resolver isso???” e
“Caramba! Era isso???!?”**

-Jhonatan de Souza

 Isso é verdade?

Compartilhe sua opinião nos comentários
abaixo 