**Aula 1 – Lógica de programação**

**Porque um software é criado?**

1. **Para criar soluções para problemas do dia a dia.**
2. **Automação e otimização de processos.**
   1. **Automatizar tarefas na web (interagir, preencher ou buscar informações em sites)**
   2. **Automatizar criação, preenchimento, duplicação, e envio de arquivos (planilhas, docs, textos, pdfs, imagens, etc...)**
   3. **Criação, leitura e envio de e-mails.**
   4. **Tarefas que usem seu teclado e mouse.**
   5. **Praticamente qualquer processo que é feito em um dispositivo pode ser automatizado parcial ou totalmente usando a programação.**

**Aula 2 – Como um software é criado do zero e onde a lógica encaixa nisso?**

***‘Antes de existir a lógica, havia um processo... um ciclo’***

* **Estágio 1: Idealização**
* **Estágio 2: Especificação de requisitos**
* **Estágio 3: Validação da solução**
* **Estágio 4: Desenvolvimento e testes**
* **Estágio 5: Implantação e entrega**

**Aula 3 – O problema que todo iniciante enfrenta.**

***‘Me solta! Deixa eu começar a codar!’ – Todo iniciante inexperiente***

* **Erro #1 -> Querer começar a escrever o código sem entender o problema apresentado.**

**Todos passam por isso:**

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

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

**Aula 4 -> Aprenda a resolver problemas através da análise crítica.**

*“Saber programas é secundário o importante é saber criar soluções bem definidas”*

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

**Problema 1:**

* **Copo d’água: Em uma receita, precisamos exatamente de 2ml de água para prosseguir, porém, só temos disponível um copo de 5ml e um de 3ml. Crie um algoritmo para obtermos os 2ml de água necessários para prosseguir com a receita.**
  1. **Encher o copo de 5ml totalmente.**
  2. **Despejar 3ml no outro copo de 3ml.**
  3. **No copo de 5ml ficará somente 2ml.**

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

**Problema 2 ->**

* **Escreva um programa que retorne o valor hora de um funcionário com base no seu salário mensal e horas trabalhadas por mês.**
  1. **Descobrir o total de horas trabalhadas semanalmente e salário total.**
  2. **Multiplicar as horas trabalhadas semanalmente e multiplicar por 5 (5 semanas no mês), para descobrir as horas trabalhadas no mês.**
  3. **Dividir o salário total pelo total de horas trabalhadas no mês.**

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

* **Iniciante ao perceber que código é secundário.**

**Aula 5 – O que são algoritmos e como montar algoritmos do zero.**

* **O que são algoritmos?**

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

* **Quando algoritmos devem ser criados?**

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

* **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 a sequência de passos a serem feitas para chegar ao resultado esperado?**

**Problema 1 – Ligar para alguém**

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

1. **Quais sãos os dados de entrada necessários?**
   1. **Um telefone.**
   2. **Um número de celular.**
2. **O que devo fazer com esses dados?**
   1. **Pegar o telefone.**
   2. **Digitar o número do celular.**
3. **Quais são as restrições deste problema?**
   1. **Caso meu amigo não atenda, devo deixar uma mensagem dizendo “me ligue de volta”.**
4. **Qual é o resultado esperado?**
   1. **Conseguir falar com meu amigo.**
5. **Qual é a sequência de passos a serem feitas para chegar ao resultado esperado?**
   1. **Pegar o telefone.**
   2. **Entrar na agenda.**
   3. **Ir até o número do meu amigo**
   4. **Colocar para discar.**
   5. **Se meu amigo atender, conversar com ele.**
   6. **Se não, entrar em contato mais tarde.**

**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.**

1. **Quais são os dados de entrada necessários?**
   1. **Salário mensal**
   2. **Horas trabalhadas por mês**
2. **O que devo fazer com estes dados?**
   1. **Fazer o cálculo do valor hora (salário mensal / horas trabalhadas).**
3. **Quais são as restrições deste problema?**
   1. **A pessoa estar desempregada**
   2. **Não digitar o salário mensal e as horas trabalhadas.**
4. **Qual é o resultado esperado?**
   1. **Mostrar o valor hora do funcionário**
5. **Qual é a sequência de passos a serem feitas para chegar ao resultado esperado?**
   1. **Digitar o salário mensal**
   2. **Digitar as horas trabalhadas por mês**
   3. **Fazer o cálculo (salário mensal / horas trabalhadas)**
   4. **Se o valor digitado for diferente de salário e horas trabalhadas, não retornar nada.**
   5. **Exibir o valor.**

**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.**

1. **Quais são os dados de entrada necessários?**
   1. **Um valor aleatório de 1 a 10**
   2. **Chute do usuário**
2. **O que devo fazer com estes dados?**
   1. **Verificar se o número do usuário é o mesmo do valor aleatório.**
   2. **Se o chute for maior ou menor que o valor gerado, alertar o usuário sobre isso e o deixar jogar novamente até que acerte o número que foi gerado.**
3. **Quais são as restrições deste problema?**
   1. **O valor aleatório só pode ser um número entre 1 e 10.**
   2. **O tipo de dado do usuário é somente números de 1 a 10**
   3. **O programa não deve ser finalizado até que um valor chutado seja chutado corretamente.**
   4. **O usuário deve ser capaz de jogar quantas vezes quiser.**
4. **Qual é o resultado esperado?**
   1. **O programa deve identificar que o valor chutado pelo usuário é o mesmo do valor aleatório.**
5. **Qual é a sequência de passos a serem feitas para chegar ao resultado esperado?**
   1. **Gerar um número aleatório de 1 a 10**
   2. **Salvar o número**
   3. **Pegar o chute do usuário e verificar se é um número.**
   4. **Salvar o número.**
   5. **Verificar se o chute do usuário e o mesmo que o número aleatório.**
   6. **Se o chute seja menor, informar ao usuário**
   7. **Permitir o usuário digitar um número novamente.**
   8. **Se o chute seja maior, informar ao usuário**
   9. **Permitir o usuário digitar um número novamente.**
   10. **Se o chute for igual ao número gerado, informar ao usuário que ele acertou.**

**Aula 06 – 4 Conceitos OBRIGATÓRIOS para ser capaz de resolver problemas!**

* **Variáveis**
* **Condicionais**
* **Laços de repetição**
* **Coleções**

**---------------------------------------------------------------------------------------------------**

**Variáveis**

* **Quando você ouve o nome de alguém, o que você faz com este nome?**
* **Você tenta guardar ele, certo?**
* **Onde você o guarda? Você guarda na sua memória.**
* **Quando alguém te passa uma data de uma consulta, o que você faz com essa data?**
* **Você anota 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 uma barra de chocolate, você guarda ou anota essa informação em algum lugar também, certo?**

**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:**
  + **Valor hora.**
  + **Salário mensal.**
  + **Horas trabalhadas.**

**------------------------------------------------------------------------------------------**

**Condicionais**

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

**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.**

* **Utilizar condicional se o número for maior;**
* **Se o número for menor;**
* **E se o usuário acertou o número.**

**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 a um lugar.**

**Cenário real:**

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

**Cenário real #2:**

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

* **Valor = 10**
* **1ª repetição | valor = valor + 1 | 11**
* **2ª repetição | valor = valor + 1 | 11**
* **3ª repetição | valor = valor + 1 | 11**
* **4ª repetição | valor = valor + 1 | 11**
* **5ª repetição | valor = valor + 1 | 11**
* **6ª repetição | valor = valor + 1 | 11**
* **7ª repetição | valor = valor + 1 | 11**
* **8ª repetição | valor = valor + 1 | 11**
* **9ª repetição | valor = valor + 1 | 11**
* **10ª repetição | valor = valor + 1 | 11**

**Pontos 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.**

**Pontos chave**

* **Uteis para conseguir automatizar processos.**
* **É possível 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’];**

* **numerosPremiados = [12, 67, 34, 100, 55];**

**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 pelo 1º valor e somar ele ao total**
* **Passar pelo 2º valor e somar ele ao total**
* **Passar pelo 3º valor e somar ele ao total**
* **Passar pelo 4º valor e somar ele ao total**
* **Passar pelo 5º valor e somar ele ao total**
* **Exibir o valor total.**

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

**Aula 7 – Criando soluções em Pseudocódigo do básico ao avançado.**

1. **Como podemos demonstrar a solução para um problema?**
   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 toda do pseudocódigo é descrever algo em uma linguagem natural que pode ser lida por qualquer pessoa.**
* **Pseudocódigo não é padronizado, pode ser escrito de várias maneiras, não há um padrão certo ou errado de 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 (Python, JavaScript, C#, Java, etc...);**
* **Não vale a pena se tornar um especialista em pseudocódigo ou em linguagem de “aprendizado” como a linguagem de programação “portugol”.**

**Sintaxe do Pseudocódigo**

* **Input: palavra usada para receber dados do usuário**
* **Print: exibir o resultado no console**
* **If condição: condicional que controle 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 uma condição for verdadeira.**

**Exemplo 1# - Pseudocódigo**

**(com uso de variáveis)**

**Valor por hora**

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

**1º passo**

* **Salário mensal.**
* **Horas trabalhadas por mês.**

**2º passo**

* **Devo usar os valores para calcular o valor hora do funcionário utilizando a formula salário mensal / horas trabalhadas.**

**3º passo**

* **Os valores devem ser entregues somente em formato salário por hora.**

**4º passo**

* **O programa deve exibir o valor hora do funcionário.**

**5º passo**

* **Input salarioMensal**
* **Input horasTrabalhadasPorMes**
* **valorHora = salarioMensal / horasTrabalhadasPorMes**
* **print valorHora**

**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(a)”, onde X é o nome do usuário.**

**1º Passo**

* **O dado de entrada é o nome do usuário**

**2º Passo**

* **O nome do usuário deve ser exibido com a mensagem de boas-vindas**

**3º Passo**

* **O campo de nome deve conter apenas letras.**
* **O campo deve ter um nome.**

**4º Passo**

* **O programa deve mostrar a mensagem de boas-vindas com o nome do usuário digitado.**

**5º Passo**

* **Input nome**
* **Print “nome, seja bem-vindo(a)”**

**Exemplo #3 – Pseudocódigo**

**(exemplo de uso de variável)**

**(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.**

* **1º Passo -Dados de entrada**
  + **Primeiro valor**
  + **Segundo valor**
* **2º Passo – O que fazer com os dados**
  + **Comparar e exibir o maior valor entre eles**
* **3º Passo – Restrições**
  + **Devo possuir somente dois números para realizar a comparação.**
* **4º Passo – Resultado esperado**
  + **Mostrar qual o número maior entre a comparação dos dois números.**
* **5º Passo – Passos**
  + **Input primeiroNumero**
  + **Input segundoNumero**
  + **If primeiroNumero > segundoNumero**
    - **Print primeiroNumero**
  + **Else**
    - **Print segundoNumero**

**Exemplo #4 – Pseudocódigo**

**(exemplo de uso de variáveis)**

**(exemplo de uso de laço de coleções)**

**(exemplo de uso de laços de repetição)**

**Fatorial de um número**

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

* **1º Passo – Dados de entrada**
  + **Um número inteiro**
* **2º Passo – O que fazer com os dados**
  + **Multiplicar o valor de entrada por todos os seus antecessores até chegar a um.**
* **3º Passo – Restrições**
  + **O número deve ser um inteiro**
  + **O número deve ser positivo.**
* **4º Passo – Resultado**
  + **Exibir o fatorial do número recebido.**
* **5º Passo – Passos**
  + **Input numero**
  + **If número < 0**
    - **Print ‘Digite apenas números positivos’**
  + **Fatorial = 1**
  + **Loop de 1 a numero**
    - **Fatorial = fatorial \* numero**
  + **Print resultado**

**Exemplo #4 – 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.**

* **1º Passo – Dados de entrada**
  + **Array “idades”**
* **2º Passo – O que fazer com estes dados**
  + **Somar todos os valores do array e exibir.**
* **3º Passo – Restrições**
  + **Somente os valores dentro do array podem ser somados.**
* **4º Passo – Resultado.**
  + **Mostrar o resultado da soma de todos os valores do array.**
* **5º Passo – Passos**
  + **Idades = [15, 46, 75, 34, 23]**
  + **Total = 0**
  + **Loop idade em idades**
    - **Total += idade**
  + **Print total**

**Exemplo #6 – Pseudocódigo**

**(exemplo de uso de variáveis)**

**(exemplo de uso de laços de repetição)**

**(exemplo de uso de condicionais)**

**Chute o número**

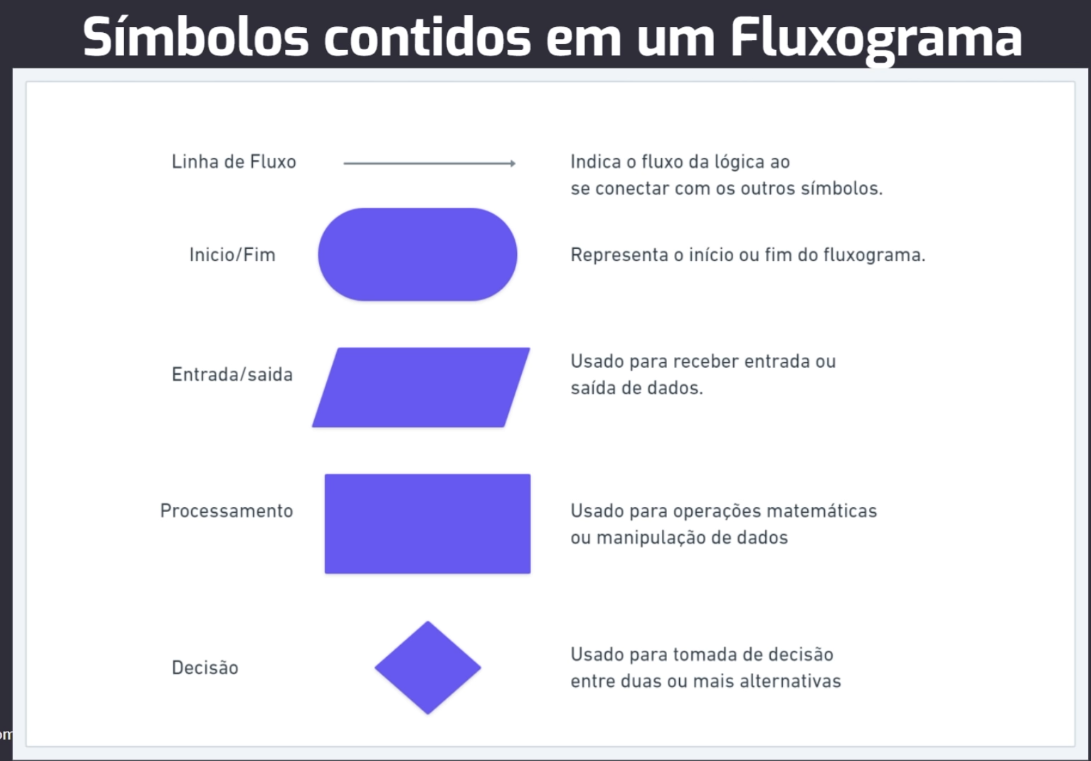
**Escreva um programa que, ao iniciar gera um valor aleatório de 1 a 10 e permite que o usuário chute um número até o valor aleatório gerado no início do programa seja chutado corretamente. O programa deve informar caso o chute tenha sido acima, abaixo ou igual ao valor aleatório gerado no início do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

* **1º Passo – Dados de entrada**
  + **Valor inicial de 1 a 10**
  + **Valor do usuário**
* **2º Passo – O que fazer com estes dados**
  + **Verificar se o número aleatório gerado é maior, menor ou igual ao valor inicial digitado pelo usuário**
* **3º Passo – Restrições**
  + **O número gerado só pode ser entre 1 e 10**
  + **O usuário só pode digitar números inteiros entre 1 e 10.**
* **4º Passo – Resultado.**
  + **Se o número digitado pelo usuário for igual ao número gerado, mostrar na tela que o usuário acertou o número.**
* **5º Passo – Passos**
  + **Input valorAleatorio entre 1 e 10**
  + **Acertou = falso**
  + **While acertou = falso**
    - **Input chute**
    - **If chute > valorAleatorio**
      * **Print “Chute é maior que o valor gerado”**
    - **If chute < valorAleatorio**
      * **Print “Chute é menor que o valor gerado”**
    - **If chute = valorAleatorio**
      * **Print “Acertou o chute!”**
      * **Acertou = verdadeiro**
    - **Else** 
      * **Print “Você digitou um valor inválido”**

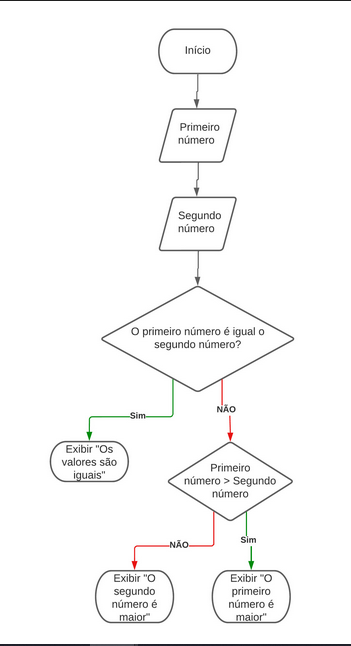
**2 avisos pra você que chegou aqui:**

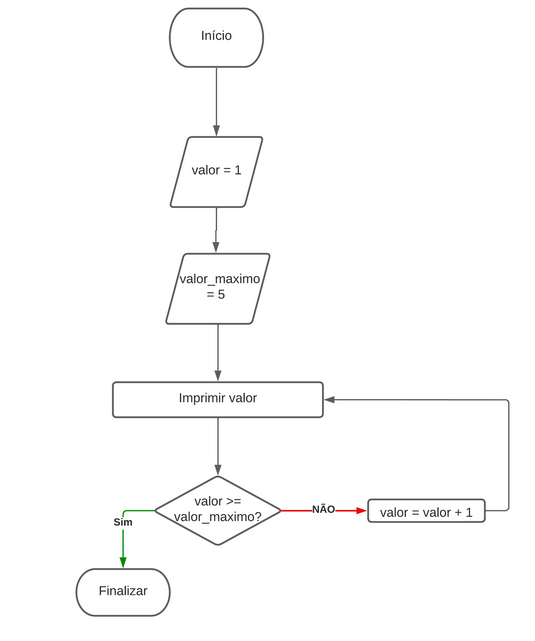
* **Não existe pseudocódigo perfeito.**
* **É absolutamente normal achar difícil no 1º contato.**

**Aula 8 – Criando soluções com fluxogramas**

* **Fluxogramas – uma representação visual gráfica de um algoritmo**

**Problema 1 – Exibir o maior dos dois números**

****

**Problema 2 – Exiba na tela todos os números de 1 a N**

**Vantagens e desvantagens do fluxograma.**

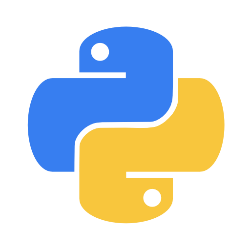
|  |  |
| --- | --- |
| **Vantagens** | **Desvantagens** |
| **Fluxogramas podem ser mais intuitivos de entender para algumas pessoas.** | **Caso tenha um processo altamente complexo, tentar o quebrar em passos em um fluxograma pode levar muito mais tempo que o pseudocódigo** |
| **Fluxogramas podem 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.** |

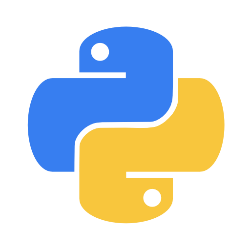
**Só existem 2 estados na mente de um programador:**

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

***-Jhonatan de Souza***

**Aula 9 – Seu primeiro programa em Python.**

**O que é Python?**

* **2º Linguagem de programação mais popular em 2021 (fonte: talentgarden.com)**
* **Permite criar scripts, sites, apps para celular, sistemas web, aplicações de desktop até aplicações que analisam dados em massa, inteligência artificial e muito mais.**
* **Extremamente amigável para iniciantes**
* **Uma das linguagens mais simples de se aprender e começar a usar. O que iremos aprendeer agora?**
* **Como criar seu 1º programa Python.**
* **Como criar e usar variáveis em Python.**
* **Como criar e usar condicionais em Python.**
* **Como criar e usar laços de repetição em Python.**
* **Como usar e criar listas em Python.**

**Aula 10 – Variáveis**

**Tipos de dados**

* **Strings**
* **Numbers**
* **Booleans**
* **F**