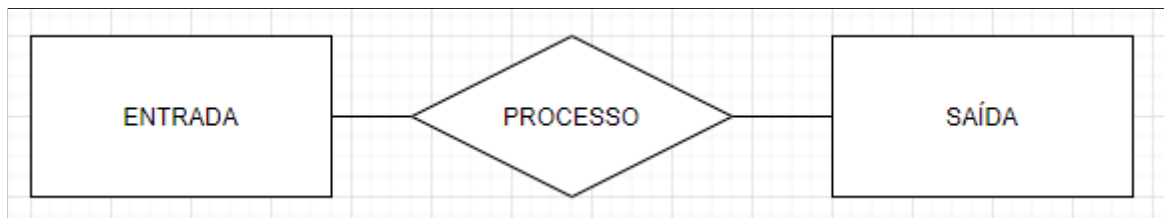


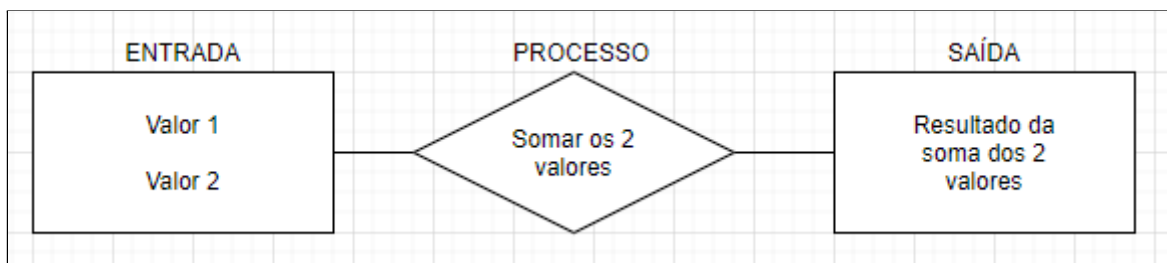
Lógica de Programação

O que é? O C# é uma linguagem de programação multiparadigma (diversos propósitos) criada pela Microsoft, sendo a principal da plataforma .NET.

Como funciona o C#? Toda lógica de programação tem que ter uma entrada, processo e saída:



Exemplo Fluxograma:



Exemplo Sintaxe:

```
//Entrada de dados
Console.WriteLine("Soma de dois números");
Console.WriteLine("-----");
Console.WriteLine("Digite o primeiro valor");
int valorUm = int.Parse(Console.ReadLine());
Console.WriteLine("Digite o segundo valor");
int valorDois = int.Parse(Console.ReadLine());

//Processamento
int soma = valorUm + valorDois;

//Exibir
Console.WriteLine("A soma dos valores é = "+soma);
```

Variáveis: É como se fossem gavetas, eu defino uma gaveta para armazenar números e dentro dela não poderei armazenar textos. É uma linguagem de tipagem forte, ou seja, quando definimos uma variável de um tipo, ela só irá armazenar aquele tipo. (Só armazena um tipo de valor)

```
string nomeCompleto = Console.ReadLine();
```

variável	nome_da_variavel	tipo_da_variavel
----------	------------------	------------------

Operadores Lógicos:

Operador	Descrição
==	Igualdade
>	Maior
<	Menor
<=	Menor igual
>=	Maior igual
!=	Diferente

Estruturas Condicionais: Precisam de uma condição para acontecer, ou seja, o sistema testa se uma condição é verdadeira e então executa comandos de acordo com esse resultado.

Estrutura Condicional	Descrição	Sintaxe
IF / ELSE	Apenas uma condição (se o aluno tiver nota acima de 5 ele está aprovado, se não tiver ele está reprovado)	<pre>if(mediaNotas>5){ Console.WriteLine("O aluno está APROVADO"); }else{ Console.WriteLine("O aluno está REPROVADO"); }</pre>
SWITCH CASE	Quando tem várias condições (Se a opção for soma, ele realizará tal processo. Se a opção for a subtração, ele realizará tal processo. Se a opção for multiplicação, ele realizará tal processo. Se a opção for divisão ele realizará tal processo. Se não for nenhuma das opções, ele mostrará a mensagem "Operação Inválida")	<pre>switch(operacao){ case "Soma": resultado = numeroUm + numeroDois; break; case "Subtracao" : resultado = numeroUm - numeroDois; break; case "Multiplicacao" : resultado = numeroUm * numeroDois; break; case "Divisao" : resultado = numeroUm / numeroDois; break; default: Console.WriteLine("Operação Inválida!"); break; }</pre>

Laços de Repetição: As estruturas de repetição são usadas para controlar a execução de códigos repetidamente até que uma condição seja verdadeira.

<i>Estrutura</i>	<i>Descrição</i>	<i>Sintaxe</i>
FOR (Contador)	Usamos o laço FOR quando sabemos exatamente quantas vezes queremos repetir um bloco de instruções.	<pre>for (valor inicial; teste booleano; incremento) for (var contador = 0; contador < 10; contador++) { Console.WriteLine("Contador: "+ contador); }</pre>
WHILE (Condicional)	Ele vai se repetir enquanto a condição que ele analisa for verdadeira	<pre>while (resposta == "sim") { Console.WriteLine("Bom dia!"); }</pre>
DO WHILE (Condicional)	Esta estrutura de repetição, garante que o bloco de instruções seja executado no mínimo uma vez,	<pre>do { Console.WriteLine("BOM DIA!!!!"); Console.WriteLine("Quer receber outro bom dia?"); resposta = Console.ReadLine(); } while (resposta == "sim");</pre>

Vetor: O vetor é usado para armazenar vários valores. (Como se fosse uma super variável, ou seja, armazena mais de um valor).

<i>Estrutura</i>	<i>Descrição</i>	<i>Sintaxe</i>
ARRAY (Vetor)	Elemento que comporta vários valores, e cada valor fica posicionado em um índice diferente.	<pre>string[] nomes = new string[4]; nomes[0] = "Gabriele Brito"; nomes[1] = "Pedro Felipe"; nomes[2] = "Mariana Matias"; nomes[3] = "Gustavo Viveiros"; Console.WriteLine("Nome: "+nomes[2]);</pre>

Comandos:

<i>Comando</i>	<i>Sintaxe</i>	<i>Terminal</i>
Criar um projeto na console	<pre>dotnet new console</pre>	<pre>dotnet new console</pre>

Executar a aplicação	<code>dotnet run</code>	<code>dotnet run</code>
Mostra o que está escrito	<code>Console.WriteLine("Digite o seu nome");</code>	<div>Digite o seu nome</div> <div>■</div>
Variável do tipo texto	<code>string nomeCompleto = Console.ReadLine();</code>	<div>Digite o seu nome</div> <div>Gabriele</div>
Mostrar o que está escrito + o que ele armazenou na variável	<code>Console.WriteLine("Bem Vindo " + nomeCompleto);</code>	<div>Digite o seu nome</div> <div>Gabriele</div> <div>Bem Vindo Gabriele</div>