O que mais precisamos aprender

Cenas dos próximos capítulos:

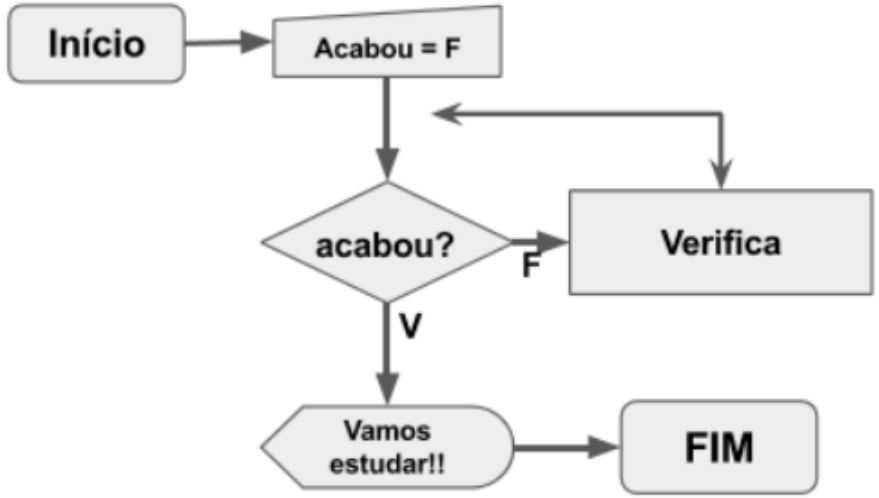
- Laços de repetição (enquanto);
- Subrotinas (Funções);
  - Bibliotecas.
  - Recursividade;
- Estruturas de dados (Vetores, Matrizes, Filas e Pilhas);
- Estatística Básica;
- Regra de três;
- Introdução a armazenamento de dados ;
- Git;

Laços de repetição

- Podemos usar **laços de repetição** para sabermos se podemos sair de casa ou não?

```
programa
{
        funcao inicio () {
                logico acabou_coronavirus = falso
                enquanto (acabou_coronavirus == falso){
                        acabou_coronavirus = verifica_pandemia()
                }
                escreva("Vamos para a Residencia de software!!")
        }
}
```

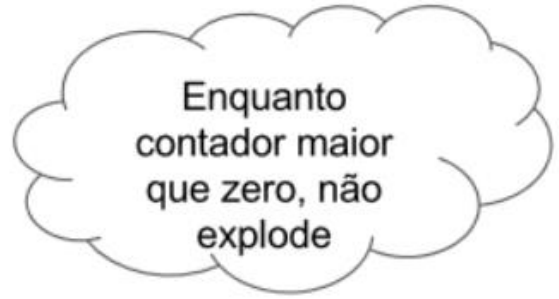
Note que o programa ainda está incompleto pois precisamos programar como verificar a pandemia



Laços de repetição

- Podemos colocar condições dentro da estrutura **enquanto**

```
programa
{
        funcao inicio() {
                inteiro contador = 10
                enquanto (contador > 0)
                {
                        limpa()
                        escreva ("Detonação em: ", contador)
                        contador = contador - 1
                        aguarde(1000) // Aguarda 1000 milisegundos (1 segundo)
                }
                limpa()
                escreva ("Booom!\n")
        }
}
```



Laços de repetição

- Além do enquanto, temos o para... até ... faça;
- Imagine que queremos saber a tabuada de um número. Quais são os requisitos?
  - Escolher um número;
  - Multiplicar o número escolhido por 1 até 10;
- Então para 1 até 10 multiplique o número escolhido.

Laços de repetição

```
programa{
    funcao inicio(){
        inteiro numero, resultado, contador
        escreva("Informe um número para ver sua tabuada: ")
        leia(numero)
        limpa()
        para (contador = 1; contador <= 10; contador++){
            resultado = numero * contador
            escreva (numero, " X ", contador, " = ", resultado , "\n")
        }
    }
}
```

Laços de repetição

```
programa{
    funcao inicio(){
        inteiro numero, resultado, contador
        escreva("Informe um número para ver sua tabuada: ")
        leia(numero)
        limpa()
        para (contador = 1; contador <= 10; contador++){
            resultado = numero * contador
            escreva (numero, " X ", contador, " = ", resultado , "\n")
        }
    }
}
```

Laços de repetição

- Se uma ação se repete em um algoritmo, em vez de escrevê-la várias vezes, podemos resumir anotando uma só vez e solicitando que ela se repita, usando umas das estruturas de repetição;

- Podemos pedir que uma ação (ou um conjunto de ações) seja executada um número definido ou indefinido de vezes, ou enquanto um estado permanecer ou até que um estado seja atingido;

- Fora do Portugol, essas estruturas são denominadas do inglês , while (enquanto), do...while (faça...enquanto), e for (para).

Laços de repetição

Voltando ao caso do coronavírus

- Lembra do código que verificava se já podíamos retornar às aulas?

```
programa
{
        funcao inicio () {
                logico acabou_coronavirus = falso
                enquanto (acabou_coronavirus == falso){
                        acabou_coronavirus = verifica_pandemia()
                }
                escreva("Vamos para a Residencia de software!!")
        }
}
```

Ficou faltando programarmos como verificaríamos se o coronavírus já está contido

Laços de repetição

Voltando ao caso do coronavírus

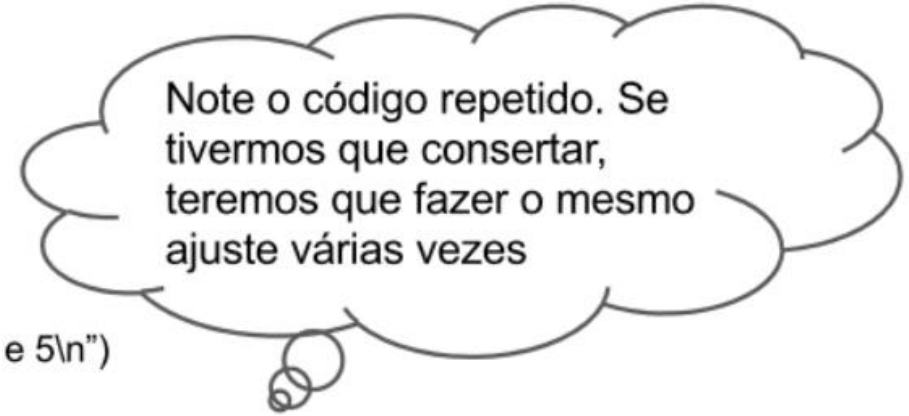
- Podemos escrever a execução da subrotina (ou função) abaixo do programa início. A lógica é semelhante à função **inicio**

```
programa
{
	funcao inicio () {
		logico acabou_coronavirus = falso
		inteiro dias_parados = 0
		enquanto (acabou_coronavirus == falso){
			acabou_coronavirus = verifica_pandemia(dias_parados)
			dias_parados ++
		}
		escreva("Vamos para a Residencia de software!!")
	}
	funcao logico verifica_pandemia(inteiro dias_parados){
		se(dias_parados>15){
			retorne verdadeiro
		 }
		retorne falso
	}
}
```

Laços de Repetição

Mais alguns exemplos - Repetição de código

```
programa {
        funcao inicio(){
                inteiro i
                para(i=0;i<20;i++)
                        escreva("*")
                escreva("\n")
                escreva("Numeros entre 1 e 5\n")
                para(i=0;i<20;i++)
                        escreva("*")
                escreva("\n")
                para(i=1; i<=5; i++)
                        escreva(i,"\n")
                para(i=0;i<20;i++)
                        escreva("*")
                escreva("\n")
        }
}
```



Saída:

```
**********
Numeros entre 1 e 5
**********
1
2
3
4
5
**********
```

Laços de Repetição

Mais alguns exemplos - Repetição de código

```
programa {
    funcao inicio(){
        inteiro i
        para(i=0;i<20;i++)
            escreva("*")
        escreva("\n")
        escreva("Numeros entre 1 e 5\n")
        para(i=0;i<20;i++)
            escreva("*")
        escreva("\n")
        para(i=1; i<=5; i++)
            escreva(i,"\n")
        para(i=0;i<20;i++)
            escreva("*")
        escreva("\n")
    }
}
```

Note o código repetido. Se tivermos que consertar, teremos que fazer o mesmo ajuste várias vezes

Saída:

```
********************
Numeros entre 1 e 5
********************
1
2
3
4
5
********************
```

Funções

- Definição : Sequência de instruções executadas somente quando chamadas por um programa em execução:
  - Devem executar uma tarefa específica
  - Um programa pode conter diversas funções, além da função principal início() , que é obrigatória;
  - As funções executam somente quando chamadas à partir da função inicio();
  - Após a execução, o fluxo retorna ao ponto imediatamente após o da chamada da função;
  - Uma função pode (ou não) retornar um valor ao bloco que a chamou;
  - Uma função pode (ou não) necessitar de um ou mais argumentos ao ser chamada;