

Recapitulando

- Já aprendemos:
 - Valores
 - Como nos organizamos como turma
 - Valores em trabalho em equipe e desenvolvimento de software
 - Conteúdo
 - O que é um algoritmo
 - O que é um programa
 - Qual ferramenta utilizaremos
 - Representações de algoritmos
 - Operações de entrada e saída
 - O que são variáveis e constantes
 - Desvios condicionais (se e senão);
 - Operadores lógicos (E, OU ...).

Próximos Passos

- Laços de repetição (enquanto);
- Estruturas de dados (Vetores, Matrizes, Filas e Pilhas);
- Subrotinas (Funções);
 - Recursividade;
 - Bibliotecas.

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()
            espera(1 dia)
        }// fim enquanto
        escreva("Podemos sair!!")
    }// fim inicio
}// fim programa
```

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")
    }
}
```

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

Além do enquanto, temos o para... até ... faça

```
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")
        }
    }
}
```

Além do enquanto, temos o para... até ... faça

- Se uma ação se repete em um algoritmo, em vez de escrevê-la várias vezes, em certos casos podemos resumir anotando uma vez só 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 Portugal, essas estruturas são denominadas do inglês , **while** (enquanto) e **for** (para)