FICHA 3

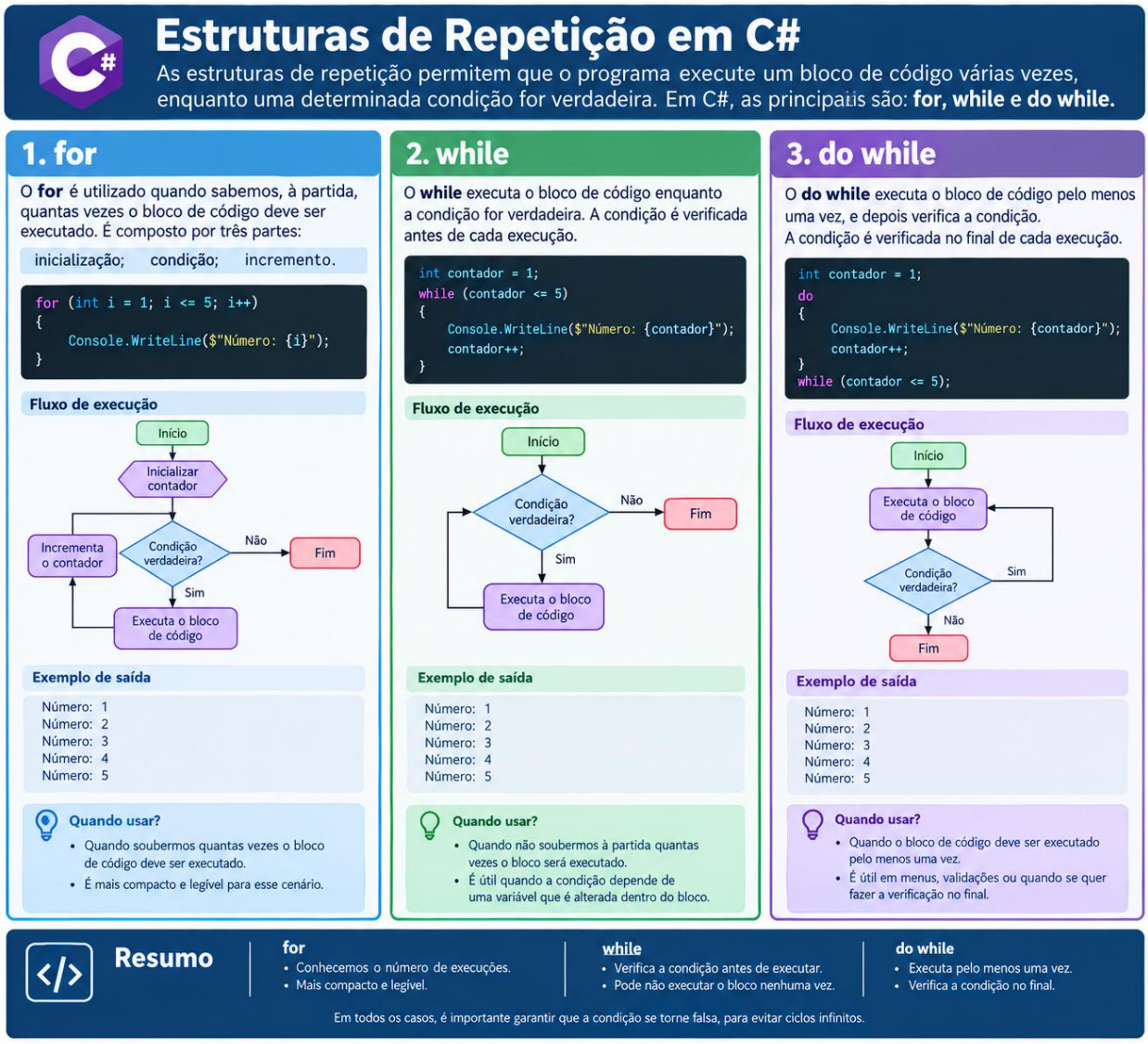
ESTRUTURAS DE REPETIÇÃO

MODALIDADE:	Aprendizagem +	Escolha um item.	
CURSO:	Tecnologias de Programação de Sistemas de Informação		
UC:	Desenvolver programas complexos em linguagem de programação estruturada	CÓDIGO UC:	UC00607
FORMADOR/A:	Bruno Silva	DATA:	

OBJETIVOS

- Utilização das estruturas de repetição com a linguagem C#

- ***Estrutura de repetição com contador <u>automático</u>:***
  - Implementação de estruturas de repetição ***For*** (Para);
- **Estrutura de repetição com contador <u>manual</u>:**
  - Implementação de estruturas de repetição ***While*** (Enquanto);
  - Implementação de estruturas de repetição ***Do .. While*** (Faz ... Enquanto);



1 – Estruturas de Repetição FOR (contador automático)

Esta estrutura permite a ***repetição duma determinada instrução ou bloco de instruções***, durante ***um número de vezes predefinido no código*** e é a única repetição de forma ***automática***.

Por defeito, **cada repetição** corresponde **a um ciclo de instruções**, mas pode ser modificado (caso necessário).

Esta estrutura deverá ter 3 parâmetros:

FOR(<condição de inicio>; <condição de fim>; <ação no final do ciclo>)

- ***condição de inicio -»*** iniciar a variável
- ***condição de fim -»*** qual a condição que irá satisfazer a condição
- ***ação no final do ciclo -»*** normalmente vamos dizer o que acontece a variável da condição de inicio: ***incrementa*** ou ***decrementa;***
- ***Exemplo Prático:***

```
for (int contador = 1; contador <= 10; contador++)
{
    // calcula a soma
    soma = soma + contador;
}
```

Exemplo 1: Percorrer todos os números de 1 até 9:

```
//Estrutura de repetição FOR para percorrer os valores
for (int i = 1; i < 10; i++)
{
    Console.WriteLine($"Valor da variavel i = {i}");
}
```

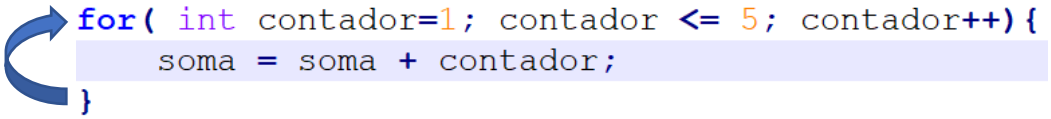
Console de Depuração do Micrc

```
Valor da variavel i = 1
Valor da variavel i = 2
Valor da variavel i = 3
Valor da variavel i = 4
Valor da variavel i = 5
Valor da variavel i = 6
Valor da variavel i = 7
Valor da variavel i = 8
Valor da variavel i = 9
```

Exemplo 2: Somar todos os números de 1 a 5 (inclusive) de forma automática

Neste exemplo, quando começamos a estrutura de repetição, vamos ter ***2 valores*** em ter em conta e que serão determinantes para o ciclo do programa:

- ***a soma*** (pois será o nosso ***acumulador*** das somas);
- ***o contador*** (***variável i***, pois devemos dar ***+1*** ao contador de cada vez que percorre a estrutura de repetição, ou seja, de cada vez que corre o ciclo da estrutura de repetição);



```
for( int contador=1; contador <= 5; contador++){
    soma = soma + contador;
}
```

```
//Estrutura de repetição FOR para somar todos os valores
//de 1 a 5 (inclusive)
int soma = 0;

for (int i = 1; i <= 5; i++)
{
    soma = soma + i;
}

Console.WriteLine($"Soma Final = {soma}");
```

```
Console de Depuração do Microsoft Visual Studio
Soma Final = 15
```

2 – Estruturas de Repetição While e Do … While (contador automático)

Estas diferem da estrutura de repetição FOR, porque utilizam uma condição manual. Dentro do bloco de instruções ***terá que ocorrer uma alteração que interfira com a condição***, caso contrário, ***entraria em ciclo infinito***.

Temos dois tipos:

- ***Enquanto*** *(While)*
- ***Faz … Enquanto*** *(Do … While)*

2.1 – Estruturas de Repetição WHILE (contador manual)

O teste à condição é realizado ***antes* da execução de qualquer instrução contida numa estrutura de repetição** e enquanto a ***condição for verdadeira***, esta volta a executar a instrução ou bloco de instruções, caso contrário, este termina e sai do ciclo.

```
int contador = 1, soma = 0;

while (contador <= 10){
    soma = soma + contador;

    contador++; /* contador = contador + 1*/
}
```

Exemplo da soma de todos os números de 1 a 10 com WHILE:

```
//estrutura de reptição while
int contador = 1;
int soma = 0;

while (contador < 10)
{
    Console.WriteLine($"Valor da variavel contador = {contador}");

    //calculo da soma
    soma = soma + contador;

    //gestão do contador de forma manual (caso contrário entra num ciclo infinito)
    contador++;
}

Console.WriteLine($"Soma Final = {soma}");
```

Console de Depuração do Microsoft Visual Studio

```
Valor da variavel contador = 1
Valor da variavel contador = 2
Valor da variavel contador = 3
Valor da variavel contador = 4
Valor da variavel contador = 5
Valor da variavel contador = 6
Valor da variavel contador = 7
Valor da variavel contador = 8
Valor da variavel contador = 9
Soma Final = 45
```

2.2 – Estruturas de Repetição DO …WHILE (contador manual)

O teste à condição ***só é realizado <u>depois</u> de executar as instruções*** contidas numa estrutura de repetição e ***enquanto a condição for verdadeira***, volta a executar a instrução ou bloco de instruções, caso contrário, termina e sai do ciclo.

```
int contador = 1, soma = 0;

do{ /* vai funcionar sempre 1 vez, pois a condição está no fim */
    soma = soma + contador;

    contador++; /* contador = contador + 1*/
}while (contador <= 10) /*só testa a condição no final*/
```

Exemplo da soma de todos os números de 1 a 10 com Do ... While:

```
//estrutura de reptição do ... while
int contador = 1;
int soma = 0;

do
{
    Console.WriteLine($"Valor da variavel contador = {contador}");

    //calculo da soma
    soma = soma + contador;

    //gestão do contador de forma manual (caso contrário entra num ciclo infinito)
    contador++;

} while (contador < 10);

Console.WriteLine($"Soma Final = {soma}");
```

Console de Depuração do Microsoft Visual Studio

```
Valor da variavel contador = 1
Valor da variavel contador = 2
Valor da variavel contador = 3
Valor da variavel contador = 4
Valor da variavel contador = 5
Valor da variavel contador = 6
Valor da variavel contador = 7
Valor da variavel contador = 8
Valor da variavel contador = 9
Soma Final = 45
```

Este tipo de estrutura de repetição é ideal para mostrar menus, ou seja, **faz** mostrar os menus **enquanto** o utilizador não coloca uma opção válida. O objetivo é **fazer** mostrar os menus **enquanto** o utilizador não carregue na tecla 4:

```
//menu de escolha
int escolha = 0;

do
{
    Console.WriteLine("***** MENU *****");
    Console.WriteLine("1 - Inserir Dados");
    Console.WriteLine("2 - Alterar Dados");
    Console.WriteLine("3 - Eliminar Dados");
    Console.WriteLine("4 - Sair");

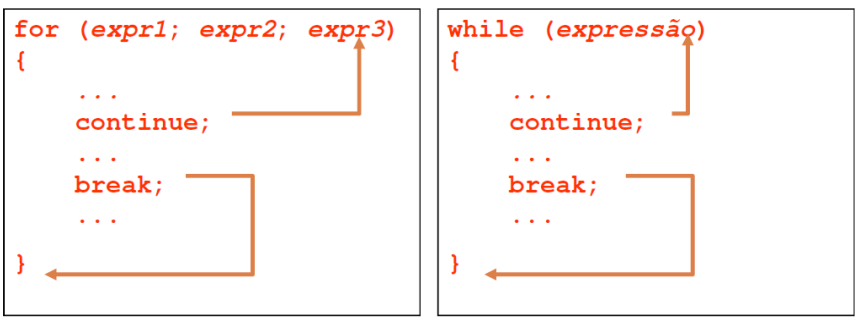
    Console.WriteLine("Escolha opção: ");
    escolha = Convert.ToInt32(Console.ReadLine());

    switch (escolha)
    {
        case 1: Console.WriteLine("Escolheu a opção inserir dados"); break;
        case 2: Console.WriteLine("Escolheu a opção alterar dados"); break;
        case 3: Console.WriteLine("Escolheu a opção eliminar dados"); break;
        case 4: Environment.Exit(0); break;

        default: Console.WriteLine("Opção Inválida"); break;
    }
} while (escolha != 4);
```

3 – Instruções Break e Continue

- ***break*** **-** sai do ciclo ou ***switch*** envolvente (mais interior)
- ***continue*** - salta para a iteração seguinte do ciclo envolvente (mais interior) no caso do ciclo for (expr1; expr2; expr3) ***salta para expr3***
    - no caso do ciclo ***while*** ou ***do (…) while*** salta para a expressão-argumento do ***while***.



Exercícios

Exercício 1 – Construa um programa para **percorrer todos os números de 1 até 10** e devolva o ***soma de todos os números;***

Exercício 2 – Construa um programa para pedir ao utilizador para inserir ***um número mínimo*** e ***um número máximo***, e de seguida, com uma estrutura de repetição automática FOR, devolva o ***soma de todos os números desse intervalo.***

Exercício 3 – Construa um programa para percorrer todos os números de 1 até 50 e devolva o ***soma de todos os números pares e ímpares*** (**dica**: utilize o resto da divisão);

Exercício 4 – Construa um programa para pedir um número inteiro ao utilizador e de seguida fazer a tabuada desse número (de 1 até 10). ***Exemplo***:

```
Insira um numero: 4
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
4 * 10 = 40
```

Exercício 5 – Construa um programa para percorrer todos os números até 10 com uma estrutura de ***repetição manual While (Enquanto)*** e devolva o ***soma de todos os números desse intervalo;***

Exercício 6 – Construa um programa para pedir somente números inteiros positivos com uma estrutura de ***repetição manual While (Enquanto)***. Se colocar números positivos o programa deve o fluxo do programa com a instrução **continue** e pedir outros números, senão, deve parar o programa com a instrução **break**;

Exercício 7 – Construa um programa para mostrar um menu de 4 opções. O programa deve ***fazer*** a exibição do menu ***tantas vezes quanto sejam necessárias***, ***enquanto*** o utilizador ***não digitar uma opção entre os valores que definiu no menu***.

Passos da conceção do programa:

1. Deve utilizar uma estrutura de repetição do tipo **Do …While** para fazer repetir o menu tantas vezes quantas necessárias até que peça para sair do programa (neste caso a opção 4);

2. Deve mostrar o menu mais abaixo com as seguintes opções:

    1. Inserir Cliente
    2. Alterar Cliente
    3. Remover Cliente
    4. Sair do programa

3. Pedir para inserir a opção (ler valor do teclado);

4. Utilize uma estrutura de decisão seletiva para mostrar uma mensagem para a opção que selecionou (com exceção da opção 4);

Exercício 8 – Construa o jogo da adivinha: o programa deve gerar um número de 1 até 10. De seguida, deve criar uma estrutura de repetição manual **do … while** para que o utilizador possa adivinhar o número. Enquanto não acertar, deve repetir o processo até que adivinhe o número.

Para gerar números aleatórios, deve instanciar a classe Random:

```
//invocação da classe Random (números aleatórios)
Random aleatorio = new Random();

//Gerar números aleatórios sem limites
Console.WriteLine(aleatorio.Next());

//Gerar números aleatórios (com limite mínimo e limite máximo)
int valor_inteiro_com_limite = aleatorio.Next(0, 50);
Console.WriteLine(valor_inteiro_com_limite);

//Gerar números decimais (entre 0 e 1)
double valor_decimal = aleatorio.NextDouble();
Console.WriteLine(valor_decimal);

//Gerar números decimais (entre 0 e 1)
//e com arredonadamento Math.Round(número, nº casas decimais)
double valor_decimal_arredondado = Math.Round(aleatorio.NextDouble(), 2);
Console.WriteLine(valor_decimal_arredondado);
```