**Programação em c# iniciante. Primeiros conceitos.**

**Namespaces:**

- Usado para organizar tipos (classes), permitindo evitar colisões de nomes.

Basicamente um namespace é uma coleção nomeada de classes.

Boa Pratica: todas as classes devem ser definidas dentro de um namespace

**Ex:**

Classe: Console

Nome completo: System.Console

Classe Console (definida pelo usuário)

Namespace MinhasClasses

MinhasClasses.Console

**Diretiva using**

Diretivas using trazem os namespaces para o escopo do seu projeto; desta forma não é necessário qualificar explicitamente os objetos com o respectivo namespace.

Ou seja, a diretiva using inclui outros namspaces no programa.

**Assembly / Montagem**

**-** As classes são compiladas em Assemblies – são arquivos, usualmente com a extensão .dll

As classes mais comuns como System.Console pertecem ao assembly *mscorlib.dll.*

Uma montagem pode conter classes definidas em muitos namespaces, e um namespace pode ocupar vários assemblies.

Para usarmos as classes de um assembly, devemos adicionar uma referencia no projeto, e incluir a diretiva “using” adequada.

**Referencias:**

Uma referencia permite que utilizemos tipos ou classes disponíveis em outros assemblies, tanto do FCL quanto de terceiros

FCL = framework class library

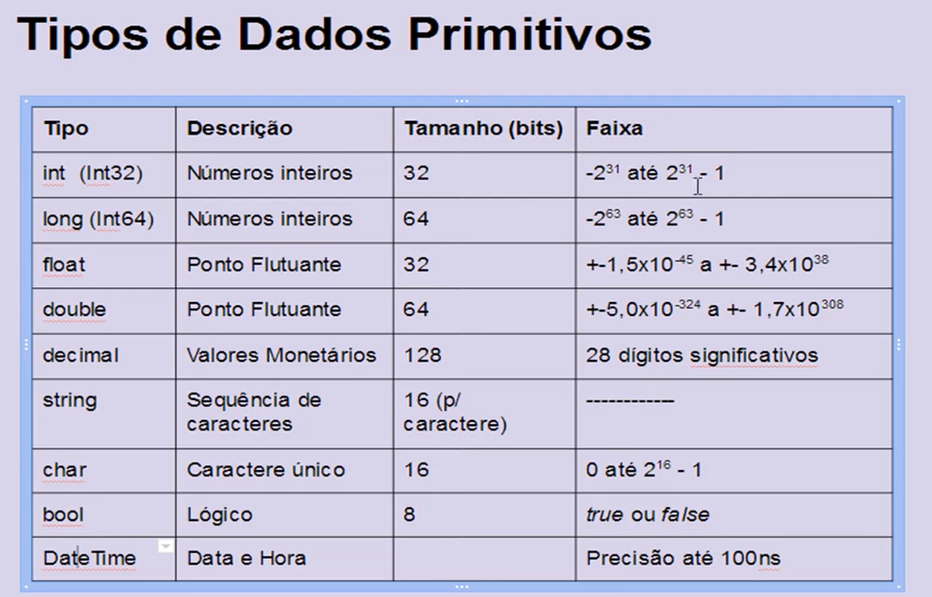
**Tipos de variáveis**

**Declaração: tipo\_variável nome\_variavel**

**Ex:** int num1 = 0;

Str nome = “lucas”

Bool trava = True

****

**Arrays**

Um array (“vetor”) é uma sequencia não-ordenada de itens.

Todos os itens dentro de um array são do mesmo tipo.

Os itens do array são armazenados em um array um bloco contiguo da memoria RAM e são acessados por meio de um número de índice.

**Sintaxe:** tipo[] nome\_array;

Ex: int[] números\_loteria;

Obs.: O tamanho do array não faz parte da declaração.

Podemos criar array de qualquer tipo, tanto de valor quanto de referência.

**Criar uma instancia do array:**

Os arrays são tipos de referencia, independentemente do tipo de seus elementos.

Ou seja, uma variável array faz referencia a um bloco de memoria que armazenara os elementos do array no heap

Assim como com as variáveis de classe, a memoria não é alocada para o array até que criemos uma instancia dele usando o operador new.

O array somente recebe espaço na memória quando a instancia é criada – e é quando especificamos o tamanho do array.

**Preenchendo um array**

Podemos inicializr os elementos de um array fornecendo uma lista de elementos separados por virgulas, entre chaves:

Int[] números\_loteria = new int[6] {2, 23, 34, 12, 29, 44}

Podemos também usar essa sintaxe:

Int[] números\_loteria = {2, 23, 34, 12, 29, 44};

Omitindo a palavra new – o compilador calcula o tamanho do array a partir da lista de elementos inicializadores e gera o código que cria o array