

LibrasKê



Convenções



Definições

Pascal Case: Cada palavra começa com letra maiúscula.

Ex: SomenteASolidaoEstaPantera

Camel Case: Cada palavra começa com letra maiúscula, menos a primeira palavra.

Ex: foiTuaCompanheiraInseparavel

Estruturas

```
public struct Slide {...}  
private struct Cidade {...}  
private struct Pessoa {...}
```

Todas as estruturas devem usar [Pascal Case](#).

[Ver referência](#)

Classes

```
public class Lapis {...}  
private class Javali {...}  
private static class Treino {...}
```

Todas as classes devem usar Pascal Case.

[Ver referência](#)

Herança

```
public class Fruta {}  
public class AbacaxiFruta : Fruta {}  
private class LaranjaFruta : Fruta {}
```

As classes filhas devem portar o nome da classe pai.

Interfaces

```
public interface IFuncionario {}  
public interface IPessoa {}  
public class Robo : IFuncionario {}  
public class Humano : IFuncionario, IPessoa {}
```

Interfaces devem ter o prefixo “I” e usar [Pascal Case](#).

[Ver referência](#)

Enumerações

```
public enum Opcoes  
{  
    Pedra,  
    Papel,  
    Tesoura  
}
```

As enumerações devem usar Pascal Case, tanto no nome quanto nas variáveis.

[Ver referência](#)

Corroutines

```
public IEnumerator FecharAppCoroutine(...){...}  
private IEnumerator JogarDadoCoroutine(...){...}  
public static IEnumerator ColherFrutaCoroutine(...){...}
```

As coroutines devem usar Pascal Case e ter o sufixo Coroutine.

ScriptableObjects

```
public class DadosSO : ScriptableObject {...}  
public class SavesSO : ScriptableObject{...}  
public class MusicasSO : ScriptableObject{...}
```

Os ScriptableObjects devem usar Pascal Case e ter o sufixo SO.

Métodos

```
public int GetNome() {...}  
private void SetNome(...) { ... }  
public static ValidarEntrada(...) { ... }
```

Os métodos devem usar Pascal Case.

[Ver referência](#)

Propriedades

```
public int Id {get; set;}  
private string Nome { get; private set;}  
private static Fruta Laranja { get; set;}
```

As propriedades devem usar Pascal Case.

[Ver referência](#)

Parâmetros

```
public void SetNome(string nome) {...}  
private string Add(int valor) {...}  
private static void SetCor(Cor cor){...}
```

Parâmetros devem usar [Camel Case](#).

[Ver referência](#)

Atributos Constantes

```
public const float GravidadeDaTerra = 9.8f;  
public const string Nome = "Libraskê";  
private const int NumeroDoSlide = 10;
```

Constantes devem usar Pascal Case.

[Ver referência](#)

Atributos Estáticos

```
public static float Pi = 3.14f;  
public static string Programa = "GSlides";  
private static Data DataAtual = new Data(28/04/21);
```

Atributos estáticos devem usar Pascal Case.

Atributos Protected

```
protected string fonte = "Arial";  
protected static int Idade = "19";  
protected const string Laboratorio = "LAVID";
```

Atributos protected devem usar Camel Case. Não se aplica em casos de atributos estáticos ou constantes.

Atributos Privados

```
private string _projeto = "VLibras";  
private bool _encerrado = false;  
private static int Ano = 2021;  
private const string Engine = "Unity";
```

Atributos privados devem ter o prefixo “_” e usar Camel Case. Não se aplica em casos de atributos estáticos ou constantes.

Encapsulamento

Evite usar **atributos públicos**. Além de ser um **mal hábito**, pois afeta o princípio Open-Closed, a IDE **Visual Studio** não **identifica** quais classes usam esses atributos, **dificultando** a **manutenção** do código.

Encapsulamento

Em vez de:

```
public string nome;
```

Você pode usar:

```
private string _nome;
```

```
public string GetNome(...) {}
```

```
public void SetNome(...) {}
```

Encapsulamento

Em vez de:

```
public string nome;
```

Você pode usar:

```
private string _nome;
```

```
public string Nome {  
    get => _nome;  
    set => _nome = value;  
}
```

Encapsulamento

Se quiser alterar algum campo no Inspector,
em vez de:

```
public string nome;
```

Use:

```
[SerializeField] private string _nome;
```

Documento criado no dia 28/04/2021
para ser aplicado no projeto LibrasKê.

Versão usada da Unity: 2020.1

Versão usada do C#: 8.0

