

## Interface de Dano e Morte

Você pode fazer download do projeto Unity como ele está ao final desta aula [clicando aqui](#) ([https://github.com/alura-cursos/unity-shooter-2/raw/master/Final\\_Aula03.zip](https://github.com/alura-cursos/unity-shooter-2/raw/master/Final_Aula03.zip)).

O Jogador já tem método de dano mas o Inimigo não, então vamos fazer a mesma coisa de uma forma mais dinâmica?

Para isso vamos criar uma Interface, mas não é Interface de visualização, é Interface no código! Elas servem para definirmos métodos que uma classe deve implementar, assim facilitando o nosso desenvolvimento.

Na interface vamos definir os métodos, parâmetros e retornos para tudo que precisarmos nos personagens para torná-los matáveis.

```
public interface IMatavel
{
    void TomarDano (int _dano);

    void Morrer ();
}
```

Agora vamos no Inimigo e vamos implementar essa interface no inicio do script na classe **ControlaInimigo**.

```
public class ControlaInimigo : MonoBehaviour, IMatavel
```

Agora esse script tem que implementar a interface. Isso significa que temos que fazer as mudanças necessárias como os dois métodos da interface.

```
public void TomarDano(int dano)
{
}

public void Morrer()
{
}
```

Agora temos que fazer a lógica dos métodos. Como já temos a parte de **Status**, vamos utilizá-lo para a Vida.

```
public void TomarDano(int _dano)
{
    status.Vida -= dano;
    if(status.Vida <= 0)
    {
        Morrer();
    }
}
```

```
public void Morrer()
{
    Destroy(gameObject);
}
```

Na **Bala** vamos utilizar estes métodos ao invés de destruir o inimigo diretamente, afinal a destruição do zumbi deve pertencer ao código dele.

```
void OnTriggerEnter(Collider objetoDeColisao)
{
    if (objetoDeColisao == "Inimigo" )
    {
        objetoDeColisao.GetComponent<ControlaInimigo>().TomarDano(1);
    }

    Destroy(gameObject);
}
```

Lembre-se de fazer a vida inicial do Inimigo ter valor 1 para ele continuar morrendo com somente um tiro.