

# Text Editor Project

· Usar Teclas de escape → `Console.ReadKey().Key`  
`ConsoleKey.Escape`

↳ Espera Qualquer Tecla  
esc

· `Environment.NewLine` (em uma linha) → adiciona uma quebra de linha no Terminal.

```
do  
{  
    userText += Console.ReadLine();  
    userText += Environment.NewLine;  
}  
while (Console.ReadKey().Key != ConsoleKey.Escape);
```

→ Armazena tudo que o usuário digita e quebra a linha

· Para abrir um arquivo temos que fechar p/ evitar problemas, p/ facilitar → `using()`

· Ele vai automaticamente abrir e fechar a função

· P/ criar um arquivo → `new StreamWriter(path)`

↳ caminho p/ onde o arquivo irá  
ficar e seu nome

Obs → `StreamWriter` é um struct, então o nome que você definir na DATA para usar  
seus métodos

```
using (StreamWriter file = new StreamWriter(path))  
{  
    file.Write(userText);  
}
```

↳ type  
↳ struct

· Uso do `using` facilita isso → `new StreamWriter()`

↳ adicionar automaticamente `[code]`

Usado p/ tudo.

↳ `close StreamWriter`

· Qualquer arquivo é um Array de Bytes

· Para ler um arquivo → `new StreamReader(path)`

```
using (var file = new StreamReader(path))  
{  
    string textByUser = file.ReadToEnd();  
    Console.WriteLine(textByUser);  
}
```

↳ type  
↳ struct

↳ é o arquivo do começo ao fim.