

Aulas 25,26

- Directórios e Ficheiros em C#
- Exercícios Relacionados





Introdução aos Directórios e Ficheiros

Um **Directório** é considerado como a hierarquia de pastas no sistema Operativo. No **C#** é possível manipular não só as pastas, mas a hierarquia de pastas do sistema operativo.

Fazemos isto usando a Classe **Directory**. E os Contéudos que estes directórios possuem são chamados de Ficheiros.

Os Ficheiros em **C#** são manipulados através da Classe **File** e **Stream** do Namespace **System.IO**;



Verificação e Criação de Directórios

Para verificar se um directório existe, usamos o método *Exists* da Classe *Directory*.

Para criar um directório, usamos o método *CreateDirectory* da Classe *Directory*.

Eliminação de Directórios

Para eliminar um directório usamos o método **Delete** da Classe **Directory**

Este método pode ter dois parâmetros:

- O Nome do Directório a ser apagado
- Um valor **true** opcional indicando se desejamos que todos o seu conteúdo (ficheiros e directórios devem ser apagados)



Deslocação de Directórios

Para Mover um directório da Localização do sistema de ficheiros recorre-se ao método **Move** da classe **Directory**. Devem ser especificados os directórios de Origem e Destino



Manipulação de Ficheiros

À semelhança dos directórios podemos Operar com o conteúdo dos Ficheiros, usando a Classe **File**

Exists : Verifica se um Ficheiro Existe

CreateText : Cria um Ficheiro de Texto

Delete : Elimina o Ficheiro especificado



Manipulação de Ficheiros

Move: Move um ficheiro de um lugar para outro. Deve ser especificado a origem e destino

Copy: Copia ficheiros. Deve ser especificado a origem e Destino



Leitura e Escrita de Ficheiros

O C# utiliza o Stream para ler e escrever em arquivos.

A classe Stream está disponível na biblioteca IO, por isso não esqueça de incluir o comando:

using System.IO;

Para criar um objeto do tipo writer utilize o comando.

```
StreamWriter wr = new StreamWriter(@"c:\pasta\arquivo.txt", true);
```


Leitura e Escrita de Ficheiros

Colocar o **@** antes do path do arquivo faz com que o compilador não interprete a barra **"\"** como sendo uma mudança de linha ou tabulação (**\n** ou **\t**), e sim como uma string. Se o parâmetro **true** for omitido (ou for usado **false**), o arquivo será criado a cada execução deste comando.



Leitura e Escrita de Ficheiros

Para escrever no arquivo utilize o comando:

```
wr.WriteLine("Este é o texto a escrever no arquivo");
```

Não esqueça de fechar o arquivo no final da escrita para evitar perda de dados.

```
wr.Close();
```

Para a leitura de dados do arquivo utiliza-se o `StreamReader`:

```
StreamReader rd = new  
StreamReader(@"c:\pasta\arquivo.txt");
```



Leitura e Escrita de Ficheiros

Observe o exemplo abaixo. Duas linhas serão gravadas no arquivo e após, estas linhas são recuperadas e inseridas em uma ListBox.

```
StreamWriter wr = new StreamWriter(@"c:\pasta\arquivo.txt", true);  
wr.WriteLine("Primeira linha");  
wr.WriteLine("Segunda linha");  
wr.Close();  
  
StreamReader rd = new StreamReader(@"c:\pasta\arquivo.txt");  
while(!rd.EndOfStream){  
    string linha = rd.ReadLine();  
    lst1.Items.Add(linha);  
}  
rd.Close();
```



Leitura e Escrita de Ficheiros

Faça um programa que permita definir num ficheiro o número de vezes que o usuário deve abrir um programa. Quando o Número for excedido o programa é bloqueado. Avise ao usuário o número de vezes que ele usou e quantas vezes faltam para bloquear o programa.

Trabalhos em Grupo sobre Files

| Grupo | Tema |
|-------|-----------------------------------------------|
| 01 | Criação de Uma Agenda de Contactos |
| 02 | Controle de uma Roullot |
| 03 | Sistema de Vendas para Cantinas |
| 04 | Sistema de Gestão Bancária |
| 05 | Sistema de Controle de Empréstimos |
| 06 | Jogo de Perguntas e Respostas com Ranking |
| 07 | Sistema de Gestão de uma Congregação das Tj's |

Trabalhos em Grupo sobre Files

| Grupo | Tema |
|-------|--------------------------------------------|
| 08 | Criação do Jogo "Não te Irrites" |
| 09 | Criação do Jogo SUDOKU |
| 10 | Minipautas Inteligentes para a Coordenação |
| 11 | |
| 12 | |
| 13 | |
| 14 | |