

Curso C# Completo

Capítulo: Trabalhando com arquivos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

File, FileInfo, IOException

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

File, FileInfo

- Namespace System.IO
- Realiza operações com arquivos (create, copy, delete, move, open, etc.) e ajuda na criação de objetos FileStream.
- File
 - static members (simples, mas realiza verificação de segurança para cada operação)
 - [https://msdn.microsoft.com/en-us/library/system.io.file\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.file(v=vs.110).aspx)
- FileInfo
 - instance members
 - [https://msdn.microsoft.com/en-us/library/system.io.fileinfo\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.fileinfo(v=vs.110).aspx)

Em programas simples porque há custo de velocidade

Sistemas complexos

IOException

Namespace System.IO

- IOException
 - DirectoryNotFoundException
 - DriveNotFoundException
 - EndOfStreamException
 - FileLoadException
 - FileNotFoundException
 - PathTooLongException
 - PipeException

Demo File

```
using System;
using System.IO;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string sourcePath = @"c:\temp\file1.txt";
            string targetPath = @"c:\temp\file2.txt";

            try {
                File.Copy(sourcePath, targetPath);
                string[] lines = File.ReadAllLines(sourcePath);
                foreach (string line in lines) {
                    Console.WriteLine(line);
                }
            }
            catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
        }
    }
}
```

Demo FileInfo

```
using System;
using System.IO;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string sourcePath = @"c:\temp\file1.txt";
            string targetPath = @"c:\temp\file2.txt";

            try {
                FileInfo fileInfo = new FileInfo(sourcePath);
                fileInfo.CopyTo(targetPath);
                string[] lines = File.ReadAllLines(sourcePath);
                foreach (string line in lines) {
                    Console.WriteLine(line);
                }
            }
            catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
        }
    }
}
```

FileStream, StreamReader

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

FileStream

[https://msdn.microsoft.com/en-us/library/system.io.filestream\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/system.io.filestream(v=vs.90).aspx)

Disponibiliza uma stream associada a um arquivo, permitindo operações de leitura e escrita.

Suporte a dados binários.

Instanciação:

- Vários construtores
- File / FileInfo



StreamReader

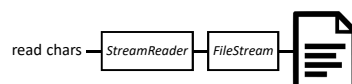
[https://msdn.microsoft.com/en-us/library/system.io.streamreader\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/system.io.streamreader(v=vs.90).aspx)

É uma stream capaz de ler caracteres a partir de uma stream binária (ex: FileStream).

Suporte a dados no formato de texto.

Instanciação:

- Vários construtores
- File / FileInfo



Demo 1

```
using System;
using System.IO;
namespace Course {
    class Program {
        static void Main(string[] args) {
            string path = @"C:\temp\file1.txt";
            FileStream fs = null;
            StreamReader sr = null;
            try {
                fs = new FileStream(path, FileMode.Open); // File.OpenRead(path);
                sr = new StreamReader(fs);
                string line = sr.ReadLine();
                Console.WriteLine(line);
            }
            catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
            finally {
                if (sr != null) sr.Close();
                if (fs != null) fs.Close();
            }
        }
    }
}
```

Demo 2

```
using System;
using System.IO;
namespace Course {
    class Program {
        static void Main(string[] args) {
            string path = @"C:\temp\file1.txt";
            StreamReader sr = null;
            try {
                sr = File.OpenText(path);
                while (!sr.EndOfStream) {
                    string line = sr.ReadLine();
                    Console.WriteLine(line);
                }
            }
            catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
            finally {
                if (sr != null) sr.Close();
            }
        }
    }
}
```

using block

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

using block

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/using-statement>

Sintaxe simplificada que garante que os objetos IDisposable serão fechados.

Objetos IDisposable NÃO são gerenciados pelo CLR. Eles precisam ser manualmente fechados.

Exemplos: Font, FileStream, StreamReader, StreamWriter

Demo 1

```
using System;
using System.IO;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string path = @"c:\temp\file1.txt";

            try {
                using (FileStream fs = new FileStream(path, FileMode.Open)) {
                    using (StreamReader sr = new StreamReader(fs)) {
                        string line = sr.ReadLine();
                        Console.WriteLine(line);
                    }
                }
            } catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
        }
    }
}
```

Demo 2

```
using System;
using System.IO;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string path = @"c:\temp\file1.txt";

            try {
                using (StreamReader sr = File.OpenText(path)) {
                    while (!sr.EndOfStream) {
                        string line = sr.ReadLine();
                        Console.WriteLine(line);
                    }
                }
            } catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
        }
    }
}
```

StreamWriter

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

StreamWriter

[https://msdn.microsoft.com/en-us/library/system.io.streamwriter\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.streamwriter(v=vs.110).aspx)

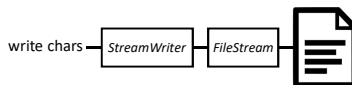
É uma stream capaz de escrever caracteres a partir de uma stream binária (ex: FileStream).

Suporte a dados no formato de texto.

Instantiation:

- Several constructors
- File / FileInfo
 - CreateText(path)
 - AppendText(String)

write chars



Demo

```

using System;
using System.IO;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string sourcePath = @"c:\temp\file1.txt";
            string targetPath = @"c:\temp\file2.txt";

            try {
                string[] lines = File.ReadAllLines(sourcePath);
                using (StreamWriter sw = File.AppendText(targetPath)) {
                    foreach (string line in lines) {
                        sw.WriteLine(line.ToUpper());
                    }
                }
            }
            catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
        }
    }
}
  
```

Directory, DirectoryInfo

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Directory, DirectoryInfo

- Namespace System.IO
- Operações com pastas (create, enumerate, get files, etc.).
- Directory
 - static members (simple, but performs security check for each operation)
 - [https://msdn.microsoft.com/en-us/library/system.io.directory\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.directory(v=vs.110).aspx)
- DirectoryInfo
 - instance members
 - [https://msdn.microsoft.com/en-us/library/system.io.directoryinfo\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.directoryinfo(v=vs.110).aspx)

Demo

- Vamos listar as pastas a partir de uma pasta informada
- Vamos listar os arquivos a partir de uma pasta informada
- Vamos criar uma pasta

```

using System;
using System.IO;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string path = @"c:\temp\myfolder";

            try {
                var folders = Directory.EnumerateDirectories(path, "*", SearchOption.AllDirectories);
                Console.WriteLine("FOLDERS:");
                foreach (string s in folders) {
                    Console.WriteLine(s);
                }

                var files = Directory.EnumerateFiles(path, "*", SearchOption.AllDirectories);
                Console.WriteLine("FILES:");
                foreach (string s in files) {
                    Console.WriteLine(s);
                }

                Directory.CreateDirectory(@"c:\temp\myfolder\newfolder");
            }
            catch (IOException e) {
                Console.WriteLine("An error occurred");
                Console.WriteLine(e.Message);
            }
        }
    }
}

```

Path

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Path

- Namespace System.IO
- Realiza operações com strings que contém informações de arquivos ou pastas.
- [https://msdn.microsoft.com/en-us/library/system.io.path\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.path(v=vs.110).aspx)

Demo

```
using System;
using System.IO;

namespace Course {
    class Program {
        static void Main(string[] args) {
            string path = @"c:\temp\myfolder\file1.txt";

            Console.WriteLine("DirectorySeparatorChar: " + Path.DirectorySeparatorChar);
            Console.WriteLine("PathSeparator: " + Path.PathSeparator);
            Console.WriteLine("GetDirectoryName: " + Path.GetDirectoryName(path));
            Console.WriteLine("GetFileName: " + Path.GetFileName(path));
            Console.WriteLine("GetExtension: " + Path.GetExtension(path));
            Console.WriteLine("GetFileNameWithoutExtension: " + Path.GetFileNameWithoutExtension(path));
            Console.WriteLine("GetFullPath: " + Path.GetFullPath(path));
            Console.WriteLine("GetTempPath: " + Path.GetTempPath());
        }
    }
}
```

Exercício de fixação

<http://educandoweb.com.br>

Prof. Dr. Nélio Alves

Fazer um programa para ler o caminho de um arquivo .csv contendo os dados de itens vendidos. Cada item possui um nome, preço unitário e quantidade, separados por vírgula. Você deve gerar um novo arquivo chamado "summary.csv", localizado em uma subpasta chamada "out" a partir da pasta original do arquivo de origem, contendo apenas o nome e o valor total para aquele item (preço unitário multiplicado pela quantidade), conforme exemplo.

Example:

Source file:

```
TV LED,1290.99,1
Video Game Chair,350.50,3
Iphone X,900.00,2
Samsung Galaxy 9,850.00,2
```

Output file (out/summary.csv):

```
TV LED,1290.99
Video Game Chair,1051.50
Iphone X,1800.00
Samsung Galaxy 9,1700.00
```

<https://github.com/acenelio/files1-csharp>
