

Tópicos Abordados



- · Arquivos e diretórios
- Drives
- Leitura e escrita de dados
 - FileStream
 - BinaryReader e BinaryWriter
 - StreamReader e StreamWriter

Dados em Arquivos



- Muitas aplicações precisam ser capazes de manipular arquivos
- Arquivos podem ser dos tipos
 - Binário
 - Texto
- Operações em arquivos são operações de I/O (input-output)

Arquivos e Diretórios



- Existem 4 classes que podem ser utilizadas para realizar operações em arquivos e diretórios
 - Arquivos
 - File | FileInfo
 - Diretórios
 - Directory | DirectoryInfo
- As classes possuem basicamente as mesmas funcionalidades
 - File | Directory só possuem métodos estáticos
 - FileInfo | DirectoryInfo precisam ser instanciadas
- Estas classes pertencem ao namespace *System.IO*

Arquivos



• Criando um novo arquivo

FileInfo f = new FileInfo(@"C:\Temp\file.txt");
f.Create();
File.Create(@"C:\Temp\file.txt");

• Copiando um arquivo para outro local

FileInfo f = new FileInfo(@"C:\Temp\file.txt");
f.CopyTo(@"C:\Temp\file2.txt");

File.Copy(@"C:\Temp\file.txt", @"C:\Temp\file2.txt");

• Verificando a existência de um arquivo

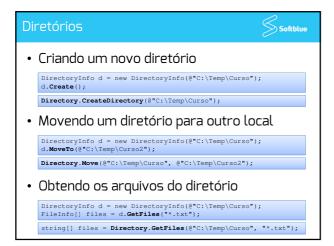
FileInfo f = new FileInfo(@"C:\Temp\file.txt");
bool b = f.Exists;
bool b = File.Exists(@"C:\Temp\file.txt");

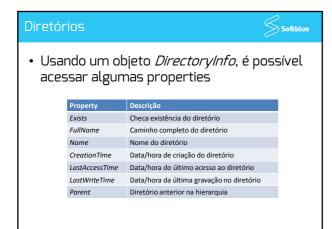
Arauivos

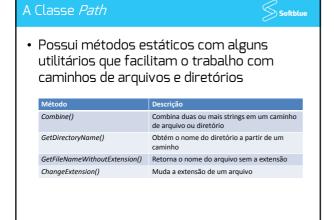


 Usando um objeto FileInfo, é possível acessar algumas properties

Property	Descrição
Extension	Extensão do arquivo
FullName	Caminho completo do arquivo
Name	Nome do arquivo (sem diretórios)
Length	Tamanho do arquivo
CreationTime	Data/hora de criação do arquivo
LastAccessTime	Data/hora do último acesso ao arquivo
LastWriteTime	Data/hora da última gravação no arquivo
Directory	Diretório onde o arquivo está contido



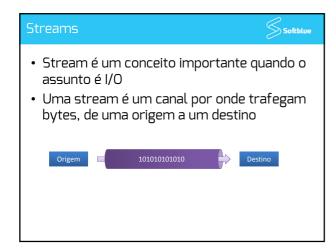


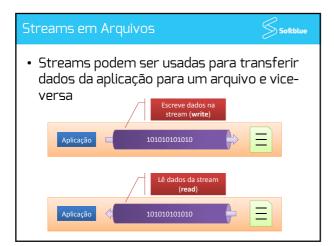


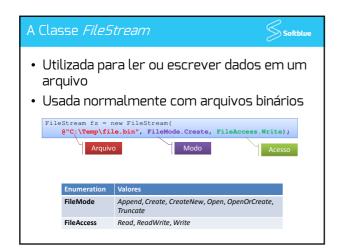
A Classe DriveInfo • Permite obter informações sobre os drives disponíveis DriveInfo drive = new DriveInfo(@"C:\"); DriveInfo[] drives = DriveInfo.GetDrives(); Property Descrição Name Nome do drive AvailableFreeSpace Espaço disponível no drive IsReady Indica se o drive está pronto DriveFormat Formato do sistema de arquivos do drive TotalSize Tamanho total do drive VolumeLabel Label associada ao drive

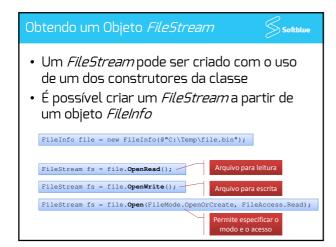
Classe <i>File</i> e a Leitura de Dados	Softblue
 A classe File tem alguns métodos estár que podem ser usados para ler dados o arquivos texto ou binários Ler todos os bytes do arquivo 	
<pre>byte[] bytes = File.ReadAllBytes(@"C:\Temp\file.bin");</pre>	
– Ler todos as linhas do arquivo	
<pre>string[] lines = File.ReadAllLines(@"C:\Temp\file.txt");</pre>	
– Ler o arquivo como uma <i>string</i>	
<pre>string text = File.ReadAllText(@"C:\Temp\file.txt");</pre>	

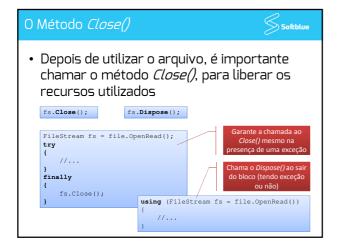
Classe File a Escrita de Dados A classe File tem alguns métodos estáticos que podem ser usados para escrever dados em arquivos texto ou binários Escrever bytes no arquivo byte[] bytes = ...; File.WriteAllBytes(@"C:\Temp\file.bin", bytes); Escrever linhas no arquivo string[] lines = ...; File.WriteAllLines(@"C:\Temp\file.txt", lines); Escrever uma string no arquivo string text = ...; File.WriteAllText(@"C:\Temp\file.txt", text);

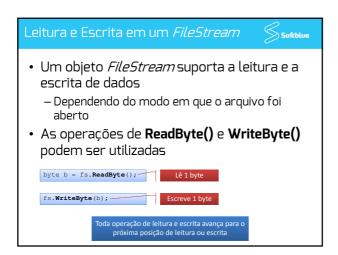


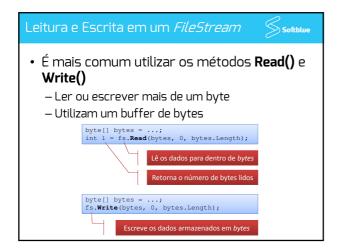














- C# possui algumas classes com os sufixos Reader e Writer
- Estas classes não representam streams, mas podem se integrar a streams e fornecer funcionalidades de mais alto nível
- Classes importantes
 - BinaryReader
 - BinaryWriter
 - StreamReader
 - StreamWriter

Classes BinaryReadere BinaryWriter Softblue



• Trabalham com a leitura e escrita de tipos de dados como *int*, *long*, *double*, etc.

