



# {Engenharia de Software 4.0}

## Curso de Programação em C#

### Aula 03 - Arquivos Programação Paralela Assemblies

20/03/23

**Veridiano Barroso**  
VERIDIANO.FILHO@UFAC.BR

Apoio



O C# nos apresenta algumas classes estáticas que facilitam o trabalho com arquivos, dentre elas:

**File**  
**Directory**

**Classe File para operações típicas, como copiar, mover, renomear, criar, abrir, excluir e acrescentar a um único arquivo por vez**

**Directory classe para operações típicas, como copiar, mover, renomear, criar e excluir diretórios.**



O modelo de programação paralela é um conjunto de tecnologias de software para expressar algoritmos paralelos e criar aplicações compatíveis com sistemas que suportam a computação paralela.

A Computação paralela é uma forma de computação em que vários cálculos são realizados simultaneamente

**Um processador pode executar somente um processo a cada instante e em um Sistema Operacional multitarefa, processos se alternam no uso do processador – cada processo é executado durante um quantum de tempo. Se houver N processadores, N processos podem ser executados simultaneamente.**

**Método `invoke()` executa cada uma das ações fornecidas, possivelmente em paralelo.**

# Aula 3 - Programação Paralela

## Programação Paralela

**Fornecer um conjunto de ações, cada um dos quais envolve um método que você deseja executar. E o .NET Framework cuida do resto e as threads são criadas e gerenciadas automaticamente em seu nome.**

# Aula 3 - Programação Paralela

## Programação Paralela

**Vamos simular uma tarefa demorada**

**Thread.Sleep para retardar o andamento do pedido**

# Aula 3 - Assinatura e implantação de assemblies; Assemblies;



**Um assembly é o arquivo que é gerado pela sua aplicação .NET**

**Esse arquivo pode ser do tipo dll ou exe.**

**Dentro dele, está o código que foi gerado pelo compilador do C#.**

# Aula 3 - Assinatura e implantação de assemblies; Assemblies;



Para criar um assembly com uma extensão .exe

**Ildasm.exe (IL Disassembler)**

# Referências



## System.IO Namespace

Contém tipos que permitem ler e gravar em arquivos e fluxos de dados, e tipos que fornecem suporte básico de diretório e arquivo.

docsmstf /



## System.Threading.Tasks Namespace

Fornece tipos que simplificam a tarefa de escrever código simultâneo e assíncrono. Os principais tipos são Task, que representa uma operação assíncrona que pode ser aguardada e cancelada e Task, que é uma tarefa que pode retornar um valor.

docsmstf /



## Como exibir o conteúdo de um assembly

Você pode usar o Desmontador IL para exibir os atributos e referências de um assembly para outros módulos e assemblies.

docsmstf /

# Vamos lá!

{ Engenharia  
de Software  
4.0 }

