

# Atividades: Serialization e Deserialization: XML e JSON em C#

## Atividade 1: Criando um Projeto para Serialização e Desserialização em JSON

**Objetivo:** Entender e praticar a serialização e desserialização (que palavra complicada) de objetos em JSON usando `System.Text.Json`.

Durante esse desafio o mais complicado é falar desserialização 3x rápido.

### Instruções:

#### 1. Crie um novo projeto:

- Abra o Visual Studio ou Visual Studio Code.
- Crie um novo projeto de console chamado `SerializacaoJsonProjeto`.
- Adicione a biblioteca `System.Text.Json` caso não esteja disponível (geralmente já está presente no .NET 5+).

#### 2. Tarefa de Serialização:

- Crie uma classe chamada `Produto` com as propriedades `Id`, `Nome`, e `Preco`.
- No `Main()`, crie um objeto da classe `Produto` e preencha seus campos.
- Serialize o objeto em JSON e exiba o resultado no console.

#### 3. Tarefa de Desserialização:

- Use a string JSON gerada para desserializar de volta em um objeto `Produto`.

- Exiba as propriedades do objeto desserializado no console.

### Desafio Extra:

- Adicione mais propriedades à classe `Produto`, como `Descricao` e `Categoria`. Veja como isso afeta a serialização e desserialização.
- 

## Atividade 2: Explorando a Biblioteca Newtonsoft.Json

**Objetivo:** Familiarizar-se com a biblioteca **Newtonsoft.Json** e suas capacidades de serialização e desserialização.

### Instruções:

#### 1. Crie um novo projeto:

- Crie um novo projeto de console chamado `SerializacaoNewtonsoftProjeto`.
- Instale o pacote NuGet `Newtonsoft.Json` no projeto.

#### 2. Tarefa de Serialização com Formatação:

- Crie uma classe `Cliente` com as propriedades `Id`, `Nome`, e `DataCadastro`.
- Serialize um objeto `Cliente` usando `Newtonsoft.Json` com o formato indentado.
- Exiba o JSON formatado no console.

#### 3. Tarefa de Desserialização:

- Crie uma string JSON representando um cliente e desserialize-a de volta para um objeto `Cliente`.
- Exiba os dados no console.

### Desafio Extra:

- Utilize atributos como `[JsonProperty]` para personalizar a serialização e desserialização (por exemplo, altere os nomes das propriedades no JSON).
- 

## Atividade 3: Serialização e Desserialização em XML

**Objetivo:** Compreender como serializar e desserializar objetos em XML usando `System.Xml.Serialization`.

### Instruções:

#### 1. Crie um novo projeto:

- Crie um projeto de console chamado `SerializacaoXmlProjeto`.

#### 2. Tarefa de Serialização:

- Crie uma classe `Pedido` com as propriedades `Numero`, `Cliente`, e `ValorTotal`.
- Serialize um objeto `Pedido` em XML e exiba o XML gerado no console.

#### 3. Tarefa de Desserialização:

- Use o XML gerado para desserializar o objeto de volta em um `Pedido`.
- Exiba os dados no console.

### Desafio Extra:

- Modifique a classe `Pedido` para incluir uma lista de itens (`List<Item>`), onde `Item` tem propriedades como `Descricao` e `Preco`. Serialize e desserialize o XML com essa nova estrutura.
- 

## Atividade 4: Comparação entre JSON e XML

**Objetivo:** Entender as diferenças práticas entre serializar objetos em JSON e XML, e discutir os cenários em que cada formato é mais adequado.

## Instruções:

### 1. Crie um novo projeto:

- Crie um projeto de console chamado `ComparacaoJsonXmlProjeto`.

### 2. Tarefa:

- Crie uma classe `Produto` (ou reutilize a anterior) com propriedades como `Id`, `Nome`, e `Preco`.
- Implemente tanto a serialização para JSON quanto para XML no mesmo projeto.
- Exiba ambos os resultados no console, um após o outro.

### 3. Reflexão:

- Faça uma reflexão sobre as diferenças visuais e de estrutura entre JSON e XML.
- Discuta em qual cenário você usaria JSON e em qual usaria XML (exemplo: APIs REST modernas preferem JSON, enquanto sistemas legados podem ainda usar XML).
- Veja exemplos na internet e em documentações oficiais

## Desafio Extra:

- Adicione um atributo à classe `Produto` que deve ser ignorado durante a serialização para ambos os formatos (use `[JsonIgnore]` e `[XmlIgnore]`).

---

## Atividade 5: Manipulando Erros durante a Desserialização

**Objetivo:** Praticar a manipulação de erros e exceções durante a desserialização (essa palavra real existe?).

## Instruções:

### 1. Crie um novo projeto:

- Crie um projeto de console chamado `ErrosDesserializacaoProjeto`.

### 2. Tarefa:

- Use uma string JSON ou XML com dados incorretos (por exemplo, um campo com tipo de dado incorreto, como uma string onde deveria haver um número).
- Tente desserializar a string e capture as exceções que forem lançadas.

### 3. Reflexão:

- Mostre como tratar exceções adequadamente e como fornecer mensagens de erro úteis ao usuário.

## Desafio Extra:

- Implemente um mecanismo que permita o tratamento de dados **inválidos sem interromper completamente a execução do programa**.
-