Exercício de Laboratório: Implementação do Padrão Factory Method em TypeScript

Objetivo

Desenvolver um sistema de gerenciamento de arquivos que permite a manipulação de diferentes tipos de arquivos (PDF, DOCX, XLSX, TXT), aplicando o padrão de projeto Factory Method. O objetivo é entender como a criação de objetos pode ser delegada para subclasses, garantindo que cada tipo de arquivo seja tratado de forma adequada.

Requisitos

1. **Produto Abstrato**: Crie uma interface `Arquivo` com os seguintes métodos:
   - `abrir(): void`
   - `salvar(): void`
2. **Produtos Concretos**: Implemente as classes `ArquivoPDF`, `ArquivoDOCX`, `ArquivoXLSX` e `ArquivoTXT` que implementam a interface `Arquivo`, com implementações específicas dos métodos `abrir()` e `salvar()`.
3. **Criador Abstrato**: Crie uma classe abstrata `EditorArquivo` com o método abstrato `criarArquivo(): Arquivo` e um método público `gerenciarArquivo()` que utiliza o método `criarArquivo()` para abrir e salvar arquivos.
4. **Criadores Concretos**: Crie as subclasses `EditorPDF`, `EditorDOCX`, `EditorXLSX` e `EditorTXT` que implementam o método `criarArquivo()` para retornar instâncias específicas de `Arquivo`.

Passo a Passo

1. **Crie a interface `Arquivo`**.
2. **Implemente as classes concretas `ArquivoPDF`, `ArquivoDOCX`, `ArquivoXLSX` e `ArquivoTXT`**.
3. **Crie a classe abstrata `EditorArquivo`** com o método abstrato `criarArquivo()`.
4. **Implemente os criadores concretos `EditorPDF`, `EditorDOCX`, `EditorXLSX` e `EditorTXT`**.

Teste a Implementação

Crie instâncias das classes concretas e chame o método `gerenciarArquivo()` para verificar o comportamento.

Resultado Esperado

A saída no console deve refletir a aberturas e salvamento de arquivos de diferentes tipos, conforme cada implementação concreta.