

REQUISITOS NÃO FUNCIONAIS DE SISTEMA

- RNF1: Performance

A aplicação foi construída de modo que os cálculos são realizados imediatamente após o fornecimento dos dados necessários, sem processamento em segundo plano ou espera significativa, como é visto no método `calculaComb`, que executa a operação de cálculo e exibe o resultado assim que os dados são inseridos.

- RNF2: Usabilidade

Exemplo Prático: O programa utiliza a interface de console para interagir com o usuário, provendo instruções claras e concisas, como as encontradas nas chamadas `Console.WriteLine` que guiam o usuário durante o processo de entrada de dados, como "Digite o valor da Gasolina separando os decimais com vírgula: ".

- RNF3: Confiabilidade

Exemplo Prático: O tratamento de exceções, como `FormatException`, assegura que o usuário não possa inserir dados em um formato errado, evitando assim o término inesperado do programa. Este tratamento é feito através dos blocos `try-catch` encontrados em métodos como `editaDados` e `inicioSist`.

- RNF4: Manutenibilidade

Exemplo Prático: O código segue uma estrutura clara e modular, com métodos distintos para cada funcionalidade como `editaDados`, `calculaComb`, e `fechaSist`. Comentários no código explicam o propósito de variáveis booleanas como `editaDadosStatus`, facilitando a compreensão e manutenção futuras.

- RNF5: Portabilidade

Exemplo Prático: O código não depende de APIs específicas de sistema operacional e usa somente funcionalidades padrão do .NET Framework, que são suportadas em plataformas Linux, Mac e Windows. Isso é exemplificado pelo uso das funções básicas da classe `Console` e pela ausência de chamadas a APIs nativas específicas.