## TRABALHO A3

**Estrutura do Trabalho**

**1. Proposta de Problema**

**Título:** Sistema de Gerenciamento de Doações para Vítimas de Enchentes no Rio Grande do Sul

**Descrição do Problema:** O objetivo deste trabalho é desenvolver um sistema de gerenciamento de doações para ajudar as vítimas de uma enchente no Rio Grande do Sul. O sistema deve permitir o registro, cálculo e armazenamento das doações recebidas. As principais funcionalidades do sistema são:

1. **Receber Doações:** O sistema deve permitir que os doadores informem os detalhes da doação, incluindo tipo (dinheiro, alimentos, roupas, etc.), quantidade e data.
2. **Calcular Total de Doações:** O sistema deve somar todas as doações e apresentar o total de maneira clara e compreensível.
3. **Armazenar Informações de Doações:** Todas as doações devem ser armazenadas em um banco de dados ou arquivo texto para futura análise e recuperação.

**2. Requisitos Funcionais**

**Utilização da IA Generativa:** Submeteremos a descrição do problema a uma IA Generativa para identificar os requisitos funcionais. A seguir, as funcionalidades identificadas pela IA:

**Funcionalidades do Sistema:**

1. Interface para entrada de doações (tipo, quantidade, data).
2. Validação dos dados de entrada.
3. Cálculo automático do total de doações.
4. Armazenamento das doações em um banco de dados ou arquivo.
5. Interface para exibir o total de doações.

**3. Crítica à IA**

**Análise Crítica das Sugestões da IA:** A IA foi eficaz em identificar as funcionalidades essenciais para o sistema. No entanto, é importante considerar algumas melhorias:

* **Interface do Usuário:** A IA sugeriu uma interface para entrada de doações, mas não detalhou a necessidade de uma interface amigável e intuitiva.
* **Validação de Dados:** A IA mencionou a validação de dados, mas poderia especificar tipos de validação, como verificar se a quantidade é um número positivo.
* **Relatórios e Análises:** A IA não mencionou funcionalidades adicionais como geração de relatórios ou análise de dados, que podem ser úteis.

**4. Diagrama de Classes**

**Descrição Textual do Diagrama de Classes:**

* **Classe Doacao:** Representa uma doação individual.
  + **Propriedades:** tipo (String), quantidade (int), data (String)
  + **Métodos:** construtor, getters e setters
* **Classe GerenciadorDoacoes:** Gerencia as doações.
  + **Propriedades:** listaDeDoacoes (ArrayList<Doacao>)
  + **Métodos:** adicionarDoacao(Doacao), calcularTotalDoacoes(), salvarDoacoes(), carregarDoacoes()

**Relações:**

* **Doacao** é utilizada por **GerenciadorDoacoes** para armazenar e manipular as doações.

**5. Estratégia de Programação com IA**

**Estratégias Sugeridas:**

1. **Revisão de Código com IA:** Escrever o código e solicitar à IA revisões para detectar erros ou sugerir melhorias.
2. **Exercícios de Prática:** Solicitar à IA exercícios relacionados aos tópicos do trabalho para reforçar o aprendizado.
3. **Programação em Pares com IA:** Desenvolver o código em colaboração com a IA, discutindo abordagens e soluções.

**Análise Crítica do Código:**

* **Positivo:** O código é modular e utiliza classes para representar entidades do problema. Métodos são usados para encapsular funcionalidades.
* **Melhoria:** Poderíamos adicionar tratamento de exceções mais robusto, interfaces gráficas para entrada e exibição de dados, e persistência mais avançada utilizando bancos de dados.