**TRABALHO FINAL**

**Professor: Alexandre de Oliveira(Montanha)**

**1-** Proposta de Problema: descrita pelo professor na atividade postada no github.

**2-** Requisitos Funcionais:

1. Criar **Formulário de Doação** para entrada de tipo, quantidade e data de doação.
2. Criar **Banco de Dados** para Armazenamento seguro e estruturado das doações.
3. Criar **Função** para somar todas as doações registradas.
4. **Interface** clara para mostrar o total acumulado de doações.
5. Exibir **Consulta de Doações** para filtrar e visualizar doações específicas
6. **Geração de Relatórios** das doações entregues e em estoque
7. Criar **Sistema Segurança de Dados** para Medidas de segurança para proteção dos dados.
8. Fazer **Backup de Dados** para maior segurança dos dados e tê-los salvos caso ocorra uma queda de energia.

**3-** Crítica à IA:

A IA pode ser usada como forma de ajuda, consulta e para otimização de tempo, mas é sempre importante que a pessoa que esteja pedindo algo para IA saiba fazer a pergunta da forma correta para melhor interpretação da IA. Nesse caso em que pedi ajuda a IA para citar requisitos funcionais tive uma boa resposta com edição de alguns detalhes.

**4-** Diagrama de Classes:

#### 1. Classe Doacao

* Id (int): Identificador único da doação.
* Tipo (string): Tipo de doação (dinheiro, alimentos, roupas, etc.).
* Quantidade (double): Quantidade da doação.
* Data (DateTime): Data em que a doação foi feita.
* Doacao(int id, string tipo, double quantidade, DateTime data): Construtor para inicializar uma nova doação.

#### 

#### 

#### 

#### 2. Classe DoacaoRepository

* Conexao (SQLiteConnection): Conexão com o banco de dados.
* AddDoacao(Doacao doacao): Adiciona uma nova doação ao banco de dados.
* GetTotalDoacoes(): Calcula o total de todas as doações.
* GetDoacoesPorFiltro(string tipo, DateTime? dataInicio, DateTime? dataFim): Recupera doações com base em critérios de filtro.

#### 3. Classe DoacaoService

* Repository (DoacaoRepository): Instância do repositório de doações.
* RegistrarDoacao(string tipo, double quantidade, DateTime data): Registra uma nova doação no sistema.
* CalcularTotalDoacoes(): Calcula o total de doações utilizando o repositório.
* ConsultarDoacoes(string tipo, DateTime? dataInicio, DateTime? dataFim): Consulta doações com base em critérios fornecidos.

#### 4. Classe MainForm

* Service (DoacaoService): Instância do serviço de doações.
* btnRegistrarDoacao\_Click(object sender, EventArgs e): Manipula o evento de clique do botão para registrar uma doação.
* btnVerTotal\_Click(object sender, EventArgs e): Manipula o evento de clique do botão para visualizar o total de doações.

#### 5. Classe RegistrarDoacaoForm

* MainForm (MainForm): Referência ao formulário principal para atualizar a interface após o registro de uma doação.
* btnSubmit\_Click(object sender, EventArgs e): Manipula o evento de clique do botão para submeter uma nova doação.

### Exemplificação de Uso das Classes

1. Registrar uma Doação:
   * O usuário clica no botão "Registrar Doação" no MainForm.
   * RegistrarDoacaoForm é exibido, e o usuário insere os dados da doação.
   * Ao submeter a doação, RegistrarDoacaoForm chama RegistrarDoacao em DoacaoService.
   * DoacaoService valida os dados e chama AddDoacao em DoacaoRepository.
   * DoacaoRepository armazena os dados no banco de dados.
2. Calcular Total de Doações:
   * O usuário clica no botão "Ver Total de Doações" no MainForm.
   * MainForm chama CalcularTotalDoacoes em DoacaoService.
   * DoacaoService chama GetTotalDoacoes em DoacaoRepository.
   * DoacaoRepository calcula e retorna o total de doações do banco de dados.
   * MainForm exibe o total ao usuário.

**5-** Estratégia de Programação com IA:

1. Usar a IA para criar uma integração e poupar tempo que uma pessoa gastaria estudando a biblioteca de um site ou empresa.
2. Implementar IA em projetos para preenchimento automático de dados ou sugestão baseado em usuários anteriores.
3. Baixar IA no compilador de códigos para auxílio na criação de projetos.
4. Ter IA conectada ao banco de dados da empresa ou ao pc do programador para achar dados mais facilmente.
5. Pedir auxílio da IA para corrigir erro no código que o programador não esteja conseguindo resolver.
6. Ajuda em estudos, com a IA criando exercícios e apresentações sobre o tema.