**Trabalho Final Montanha**

2- Solicitando a IA para me gerar os requisitos funcionais, ou seja, as funcionalidades do sistema proposto, recebi essa resposta:

1. Recepção de Doações

Interface de usuário para doadores registrarem suas doações:

Deve permitir a inserção dos detalhes da doação.

Campos de entrada:

Tipo de doação: Dinheiro, alimentos, roupas, etc.

Quantidade da doação: Valor monetário, quantidade de itens, peso, etc.

Data da doação: Data em que a doação foi feita.

Nome do doador (opcional): Nome da pessoa ou organização que está doando.

Contato do doador (opcional): E-mail ou telefone para contato.

2. Cálculo Total de Doações

Função para somar o total de doações recebidas:

Deve calcular o total acumulado de todas as doações.

Exibição do total acumulado:

Mostrar o total de doações em uma interface amigável.

3. Armazenamento de Informações

Banco de dados para armazenar informações sobre todas as doações:

Alternativa: Armazenamento em arquivos de texto (CSV, JSON, etc.).

Capacidade de recuperação e análise posterior dos dados armazenados:

Função para buscar e exibir dados armazenados.

4. Visualização e Relatórios

Interface para visualizar todas as doações registradas:

Lista ou tabela de todas as doações.

Filtros para visualizar doações:

Por tipo de doação, data, ou doador.

Geração de relatórios detalhados:

Relatórios sobre as doações recebidas com opções de exportação (PDF, Excel, etc.).

5. Administração do Sistema

Função de login para administradores:

Gerenciamento do sistema por administradores.

Funções de CRUD (Create, Read, Update, Delete):

Gerenciamento de registros de doações.

Controle de acesso:

Proteger informações sensíveis e garantir a segurança do sistema.

6. Comunicação com Doadores

Opção para enviar recibos de doações por e-mail:

Geração e envio automático de recibos.

Notificações de agradecimento:

Enviar mensagens de agradecimento para os doadores.

7. Segurança e Privacidade

Implementação de medidas de segurança:

Proteção dos dados dos doadores.

Conformidade com leis de proteção de dados:

Adequação às legislações aplicáveis, como LGPD (Lei Geral de Proteção de Dados).

8. Usabilidade e Acessibilidade

Design responsivo:

Acessível em dispositivos móveis.

Interface intuitiva e fácil de usar:

Facilitar o uso tanto para doadores quanto para administradores.

9. Integração com IA Generativa

Utilização de IA para sugerir melhorias na interface:

Sugestões para aprimoramentos no processo de registro de doações.

Implementação de um chatbot:

Assistência a doadores e administradores na navegação pelo sistema.

10. Backup e Recuperação de Dados

Sistema de backup regular dos dados armazenados:

Procedimentos para backup automático e regular.

Função de recuperação de dados:

Recuperação dos dados em caso de falha do sistema.

3- A resposta que recebi da IA me traz todos os aspectos necessários para um sistema de gestão de doações bem organizado, desde a aceitação de doações até a proteção de dados em backups, oque achei bem interessante. Tendo em vista esse poder de processamento e o enorme banco de dados que as inteligências artificiais tem acesso, acredito que a ia pode transformar a computação aumentando a precisão, fornecendo soluções inteligentes e personalizadas e melhorando e facilitando a experiência do usuário.

Dito isso, a respeito da IA, não tenho muitas críticas, a resposta veio bem completa, trazendo todos os requisitos funcionais que também consegui localizar, suas características e funções de forma bem completa.

4 - Classe Doacao

Propriedades:

id: Identificador único da doação.

tipo: Tipo de doação (dinheiro, alimentos, roupas, etc.).

quantidade: Quantidade da doação.

data: Data da doação.

doador: Referência ao objeto Doador.

Métodos:

\_\_init\_\_(id, tipo, quantidade, data, doador): Inicializa os atributos da doação.

get\_detalhes(): Retorna os detalhes da doação.

Classe Doador

Propriedades:

id: Identificador único do doador.

nome: Nome do doador.

contato: Contato do doador.

Métodos:

\_\_init\_\_(id, nome, contato): Inicializa os atributos do doador.

get\_detalhes(): Retorna os detalhes do doador.

Classe SistemaGerenciamentoDoacoes

Propriedades:

doacoes: Lista de doações.

doadores: Lista de doadores.

Métodos:

\_\_init\_\_(): Inicializa as listas de doações e doadores.

adicionar\_doacao(tipo, quantidade, data, doador): Adiciona uma nova doação.

calcular\_total\_doacoes(): Calcula o total de doações.

listar\_doacoes(): Lista todas as doações.

adicionar\_doador(nome, contato): Adiciona um novo doador.

gerar\_relatorio(): Gera um relatório detalhado das doações.

Relações entre as Classes

A classe Doacao tem uma relação com a classe Doador, indicando que cada doação pode ser associada a um doador.

A classe SistemaGerenciamentoDoacoes gerencia as listas de doações e doadores, permitindo adicionar, listar e calcular informações sobre as doações.

5 - 1. Revisão de Código pela IA

Descrição: Após escrever um trecho de código, o estudante pode solicitar à IA uma revisão. A IA pode fornecer feedback sobre a estrutura, legibilidade, eficiência e correção do código.

Exemplo de Uso:

O estudante escreve uma função para adicionar doações.

Solicita à IA uma revisão para identificar possíveis melhorias ou erros.

A IA sugere refatorações, aponta inconsistências ou oferece alternativas mais eficientes.

2. Estudo Assistido pela IA

Descrição: O estudante pode usar a IA para estudar os temas do trabalho, solicitando explicações detalhadas, exemplos práticos e exercícios para praticar.

Exemplo de Uso:

O estudante pergunta à IA sobre conceitos específicos, como “O que é uma classe em programação orientada a objetos?”.

A IA fornece uma explicação, exemplos de código e exercícios para praticar.

O estudante pratica os exercícios e pode solicitar correções e feedback à IA.

3. Programação por Pares com a IA

Descrição: O estudante programa em conjunto com a IA, onde a IA atua como um parceiro de programação, oferecendo sugestões em tempo real, ajudando a resolver problemas e complementando o conhecimento do estudante.

Exemplo de Uso:

O estudante começa a implementar a interface de usuário.

A IA sugere melhorias na estrutura do código e auxilia na correção de erros.

O estudante e a IA discutem soluções para problemas encontrados durante a codificação.

4. Solicitação de Exemplos e Snippets de Código

Descrição: O estudante pode solicitar à IA exemplos de código ou snippets para tarefas específicas, acelerando o desenvolvimento e aprendizado.

Exemplo de Uso:

O estudante pede um exemplo de como conectar o sistema a um banco de dados.

A IA fornece um snippet de código com uma explicação de como ele funciona.

O estudante adapta o snippet ao contexto do seu projeto.

5. Refatoração Guiada pela IA

Descrição: A IA pode ajudar o estudante a refatorar o código para torná-lo mais eficiente e legível.

Exemplo de Uso:

O estudante tem um código que funciona, mas não está otimizado.

Solicita à IA uma refatoração para melhorar o desempenho e a estrutura.

A IA sugere alterações, explica as melhorias e ajuda na implementação.

6. Brainstorming de Soluções

Descrição: O estudante pode usar a IA para brainstormar diferentes abordagens para resolver um problema específico.

Exemplo de Uso:

O estudante enfrenta um desafio ao calcular o total de doações.

Pede à IA várias abordagens possíveis para solucionar o problema.

A IA apresenta diferentes métodos, e o estudante escolhe a melhor abordagem para implementar.

7. Testes e Debugging com IA

Descrição: A IA pode auxiliar na criação de testes automatizados e no processo de debugging.

Exemplo de Uso:

O estudante implementa uma nova funcionalidade e quer garantir que ela funcione corretamente.

A IA ajuda a criar testes automatizados para essa funcionalidade.

Durante o debugging, a IA identifica possíveis causas de erros e sugere correções.