

Link do código:

https://github.com/Anacarolina98/Trabalho_Final_Programa-o/tree/main

Sistema de Gerenciamento de Doações para Vítimas de Enchente no Rio Grande do Sul

Objetivo

Desenvolver um sistema para gerenciar doações destinadas às vítimas de uma enchente no Rio Grande do Sul, facilitando o processo de recebimento, cálculo e armazenamento das doações.

Funcionalidades do Sistema

1. Receber Doações

- **Tela de Entrada de Doações:** Interface onde os doadores podem inserir os detalhes da doação.
 - **Tipo de Doação:** Campo para selecionar o tipo de doação (dinheiro, alimentos, roupas, etc.).
 - **Quantidade:** Campo para inserir a quantidade doada (quantia em dinheiro, peso dos alimentos, número de peças de roupa, etc.).
 - **Dados do Doador:** Campos para inserir informações do doador (nome, contato, etc.).
- **Validação de Dados:** Verificação para garantir que todos os campos obrigatórios estão preenchidos e que os dados inseridos são válidos.

2. Calcular Total de Doações

- **Função de Soma:** Algoritmo que some todas as doações recebidas e calcule o total.
- **Interface de Exibição:** Tela que mostre de maneira clara e compreensível o total acumulado de doações, podendo ser filtrada por tipo de doação.

3. Armazenar Informações de Doações

- **Banco de Dados/Arquivo de Texto:** Mecanismo para armazenar todas as informações sobre as doações de maneira organizada e segura.

Entrevista

Quem serão os principais usuários do sistema (administradores, doadores, voluntários, etc.)?

- "Os principais usuários serão os administradores, que vão gerenciar tudo, os doadores, que vão fazer as doações, e os voluntários, que vão ajudar a organizar e distribuir as doações."

Que tipo de acesso e permissões diferentes usuários devem ter?

- "Os administradores terão acesso completo, podendo ver, editar e excluir todas as doações. Os doadores só poderão registrar suas próprias doações e ver um resumo delas. Os voluntários terão acesso para ver as doações e marcar o status de distribuição."

Quais tipos de doações serão aceitas (dinheiro, alimentos, roupas, etc.)?

- "Vamos aceitar dinheiro, alimentos, roupas, itens de higiene e qualquer outra coisa útil que possa ajudar as vítimas da enchente."

Que informações específicas você deseja coletar sobre cada doação?

- "Para cada doação, precisamos saber o tipo de doação, a quantidade, o nome do doador, o contato do doador, e a data da doação."

Como você gostaria que o formulário de doações fosse estruturado?

- "O formulário deve ser bem simples e direto: uma lista suspensa para o tipo de doação, um campo para a quantidade, campos de texto para nome e contato do doador, e um campo de data que pode ser preenchido automaticamente com a data atual."

Como você gostaria que o total das doações fosse apresentado (por tipo, por período, etc.)?

- "Gostaria de ver o total das doações separado por tipo e também filtrável por período, como semanal, mensal ou anual."

Existem métricas específicas que você deseja acompanhar além do total de doações?

- "Sim, gostaria de acompanhar o número de doadores, o valor médio das doações em dinheiro, e a quantidade média dos outros tipos de doações."

Quais informações precisam ser armazenadas para cada doação e doador?

- "Para cada doação, precisamos armazenar o tipo, a quantidade, o nome do doador, o contato do doador, e a data da doação."

Por quanto tempo você precisa manter os dados armazenados?

- "Precisamos manter os dados armazenados por pelo menos cinco anos para garantir que possamos fazer auditorias e relatórios históricos."

Existem requisitos específicos para a segurança dos dados?

- "Sim, precisamos garantir que os dados pessoais dos doadores estejam protegidos, utilizando criptografia para armazenar essas informações e acessos controlados."

Existe algum design ou layout específico que você prefere para a interface do usuário?

- "Quero algo limpo e fácil de navegar, talvez com cores suaves e seções bem definidas para cada funcionalidade."

Quais são as funcionalidades principais que devem estar facilmente acessíveis na interface?

- "Os usuários devem facilmente encontrar o formulário de doação, um resumo dos totais de doações, e opções de filtro para visualizar as doações por tipo e período."

Que tipos de relatórios você precisa gerar a partir dos dados de doações?

- "Precisamos de relatórios que mostrem o total de doações por tipo, por período, o número de doadores e as doações por região se possível."

Quais filtros e opções de visualização você deseja para os relatórios?

- "Gostaria de filtros por tipo de doação, data, e talvez por região. Além disso, poder visualizar os dados em gráficos seria ótimo."

Quem será responsável pela manutenção e suporte do sistema após a implantação?

- "Idealmente, uma equipe de TI dedicada ou um fornecedor externo que possamos contratar para suporte contínuo."

Qual é o nível de suporte que você espera (tempo de resposta, tipos de problemas a serem resolvidos, etc.)?

- "Espero um suporte que responda em até 24 horas para problemas críticos e que possa resolver problemas técnicos, bugs, e realizar atualizações regulares."

Você espera que o sistema cresça em termos de número de usuários ou volume de dados?

- "Sim, esperamos que o número de doadores e as doações aumentem, especialmente durante crises maiores, então o sistema precisa ser escalável."

Existem funcionalidades adicionais que você pode querer adicionar no futuro?

- "Talvez a integração com redes sociais para facilitar a divulgação, e um módulo de análise para entender melhor os padrões de doação."

Quais são suas preocupações em relação à segurança dos dados doadores e das doações?

- "Minha maior preocupação é proteger as informações pessoais dos doadores contra vazamentos e garantir que as transações financeiras sejam seguras."

Existe alguma política de privacidade que precisamos seguir?

- "Sim, precisamos seguir as normas de proteção de dados, como a LGPD no Brasil, para garantir que os dados dos doadores sejam tratados de forma segura e transparente."

Existem regulamentações específicas que o sistema precisa seguir?

- "Além da LGPD, precisamos estar atentos a qualquer regulamentação específica sobre doações e transações financeiras."

Você tem requisitos específicos para auditorias ou relatórios de conformidade?

- "Sim, precisamos ser capazes de gerar relatórios detalhados para auditorias internas e externas, garantindo transparência nas doações recebidas e distribuídas."

Como você pretende coletar feedback dos usuários do sistema?

- "Podemos ter uma seção no sistema onde os usuários podem deixar feedback, além de enviar pesquisas por e-mail periodicamente."

Existem funcionalidades específicas para feedback e sugestões dentro do sistema?

- "Sim, uma funcionalidade onde os usuários podem facilmente enviar feedback e sugestões diretamente no sistema seria ideal."

Requisitos coletados pelo usuário

- Nós vamos ter três tipos de usuários: os chefes (administradores), os doadores bonzinhos e os voluntários heróis.
- **Chefes:** Eles são os chefões, então vão ter acesso a tudo, tipo o código secreto do cofre.
- **Doadores Bonzinhos:** Esses só querem ajudar, então eles só vão poder mexer nas próprias doações, tipo naquelas caixas de doações que você vê no supermercado.
- **Voluntários Heróis:** Eles que vão estar lá na linha de frente, então eles vão ver as doações e marcar quem ganhou o prêmio de 'melhor doador'.
- Tudo o que puder ajudar, tipo dinheiro, comida, roupas, sabonetes, até mesmo coisas que a gente nem imagina.
- A gente quer saber que tipo de coisa a pessoa doou, quanto ela doou, o nome dela, como entrar em contato e quando foi a doação.
- Vamos fazer isso ser simples: uma lista pra escolher o tipo de doação, um lugar pra colocar a quantidade, e uns campos pra pessoa colocar o nome, contato e a data.

Ah, e a data já vem preenchida automaticamente, a gente não quer dar trabalho pra ninguém.

- A gente vai dividir por tipo de doação e, se quiser, dá pra ver quanto a gente conseguiu em cada semana, mês ou ano.
- Quantas pessoas doaram, quanto foi a média das doações em dinheiro e a quantidade média das outras coisas.
- O tipo de doação, quanto foi, o nome do doador, como falar com ele e quando ele fez a doação.
- Vamos manter esses dados seguros por pelo menos uns cinco anos.
- Vamos trancar tudo isso com criptografia e só quem pode abrir a porta é quem tem a chave certa.
- Simples e intuitiva, sem muita firula.
- Um jeito fácil de fazer a doação, um resuminho das doações e filtros pra ver tudo direitinho.
- Ver o total das doações por tipo, por período e quantas pessoas ajudaram.
- Queremos gráficos e tabelas bonitas, nada de listas chatas.
- Uma galera que manja de tecnologia ou então um time que a gente contrate.
- Que seja rápido e resolva nossos pepinos logo. Tem que aguentar o tranco e aceitar mais gente ajudando.
- Tipo, uma integração com redes sociais pra espalhar o bem e um jeito de entender melhor como a gente pode ajudar mais.
- Criptografia neles!
- E a gente precisa seguir as regras sobre proteção de dados, tipo aquela LGPD que tá na moda.
- Vamos pedir umas dicas pros usuários e também mandar umas pesquisas de vez em quando.
- Vamos deixar um jeito fácil pra galera mandar suas ideias e sugestões direto pra gente.

Requisitos Funcionais

- Os administradores têm acesso total a todas as funcionalidades do sistema, enquanto os doadores e voluntários têm acesso limitado apenas às suas próprias doações e à visualização das doações, respectivamente.
- Os doadores podem registrar suas doações, especificando o tipo de doação, a quantidade, seu nome, contato e a data da doação.
- Os voluntários podem visualizar as doações registradas e marcar o doador que ganhou o prêmio de 'melhor doador', se aplicável.
- O sistema deve calcular e apresentar os totais de doações por tipo e permitir a visualização desses totais por período (semanal, mensal, anual).
- O sistema deve calcular e exibir métricas como o número total de doadores, o valor médio das doações em dinheiro e a quantidade média das outras doações.
- O sistema deve armazenar informações sobre cada doação, incluindo tipo, quantidade, nome do doador, contato e data da doação.
- Os dados das doações e dos doadores devem ser armazenados de forma segura por pelo menos cinco anos, utilizando criptografia para proteção.

- A interface do usuário deve ser simples e intuitiva, facilitando o registro de doações e a visualização dos totais.
- O sistema deve gerar relatórios que mostram o total das doações por tipo, por período e o número de pessoas que contribuíram.
- Os relatórios e totais de doações devem ser apresentados de forma visualmente atraente, utilizando gráficos e tabelas.

Requisitos não Funcionais

- Os dados das doações e dos doadores devem ser protegidos utilizando criptografia e seguindo as regulamentações de proteção de dados, como a LGPD.
- O sistema deve ser capaz de lidar com um aumento no número de doações e usuários sem comprometer o desempenho.
- O sistema deve ser fácil de manter e oferecer suporte rápido para resolver problemas e fornecer atualizações.
- O sistema deve ser capaz de se integrar com redes sociais para facilitar a divulgação e aumentar o alcance das campanhas de doação.
- Deve haver mecanismos para coletar feedback dos usuários e permitir que eles enviem sugestões diretamente para a equipe responsável pelo sistema.

Crítica à IA

Acho que a análise dos requisitos funcionais até que foi boa, mas ainda tem alguns detalhes que poderiam ter sido mais explorados. Tipo, a questão da rastreabilidade das doações e como elas são atribuídas aos voluntários. Acho que isso poderia ter sido mais detalhado, sabe? Também, sobre os relatórios para os administradores, eu não fui muito a fundo nisso. Acho que teria sido legal pensar em quais informações exatamente eles precisam para tomar decisões, porque pode variar bastante, né?

No geral, acho que foi um bom começo, mas acho ela devia ter procurado mais informações sobre os sistemas semelhantes, isso poderia ter ajudado a identificar ainda mais os requisitos importantes.

Em relação ao código preferir montar ele com base o que aprendi, comecei pedindo a IA para me estruturar o código, mas está dando bastantes erros na adicionei um print sobre, além de que ela esta elaborando ele bem mais complexo e eu preferi teste a questão do CRUD que aprendi e queria tirar a prova.

Então pedi ela para validar, pois estava com dificuldade em algumas linhas.

Diagrama de Classes

Classe: Doação

- **Propriedades:**
 - Tipo: Representa o tipo de doação (dinheiro, alimentos, roupas, etc.).
 - Quantidade: Indica a quantidade da doação.

- Data: Registra a data da doação.
- Doador: Armazena as informações do doador, como nome e contato.

Classe: Doador

- **Propriedades:**
 - Nome: Nome completo do doador.
 - Contato: Informações de contato do doador, como e-mail ou telefone.

Classe: Voluntário

- **Propriedades:**
 - Nome: Nome completo do voluntário.
 - Área de Atuação: Indica a área específica em que o voluntário está atuando, como distribuição ou coleta de doações.

Classe: Administrador

- **Propriedades:**
 - Nome: Nome completo do administrador.
 - Cargo: Indica o cargo ou função do administrador dentro do sistema.

Classe: Sistema

- **Métodos:**
 - registrarDoacao(): Responsável por registrar uma nova doação no sistema.
 - calcularTotalDoacoes(): Calcula o total de doações recebidas pelo sistema.
 - gerarRelatorio(): Gera relatórios com informações detalhadas sobre as doações.

Relacionamentos:

- A classe Doação está associada com a classe Doador, pois cada doação é feita por um doador específico.
- A classe Doação também está associada com a classe Voluntário, pois os voluntários podem estar envolvidos na coleta ou distribuição das doações.
- A classe Sistema utiliza métodos da classe Doação para registrar e calcular as doações, e também utiliza métodos próprios para gerar relatórios.

Revisão da IA sobre o Código



```
preciso que corrija este código pra mim pois a linha " private String
contato;" não está funcionando
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
```

```
class Doacao {
    private String tipo;
    private int quantidade;
    private Date data;
    private Doador doador;

    public Doacao(String tipo, int quantidade, Date data, Doador
doador) {
        this.tipo = tipo;
        this.quantidade = quantidade;
```



O código fornecido não apresenta um problema claro diretamente na linha `private String contato;`. No entanto, ao observar o código, percebi que o método `getContato` está ausente na classe `Doador`. Esse método seria necessário caso você precise acessar o atributo `contato` fora da classe `Doador`.

Vamos adicionar o método `getContato` na classe `Doador` para garantir que possamos acessar o atributo `contato`. Além disso, vamos assegurar que o código da classe `SistemaGerenciamentoDoacoes` utilize esse método caso seja necessário.

```
class Doador {
    private String nome;
    private String contato;

    public Doador(String nome, String contato) {
        this.nome = nome;
        this.contato = contato;
    }
}
```

Alterações realizadas:

1. Adicionado o método `getContato` na classe `Doador` para possibilitar o acesso ao atributo `contato`.
2. Modificado o método `gerarRelatorio` da classe `SistemaGerenciamentoDoacoes` para exibir o contato do doador no relatório.

Agora, o código deve funcionar corretamente e exibir as informações de contato dos doadores no relatório.





esta linha esta errada identifique o erro

```
public static void
lerArquivo(String caminho) {
    try (BufferedReader reader = new BufferedReader(new
FileReader(caminho))) {
        String linha;
        while ((linha = reader.readLine()) != null) {
            System.out.println(linha);
        }
    } catch (IOException e) {
        System.out.println("Erro ao ler o arquivo: " + e.getMessage());
    }
}
```



Você está absolutamente correto, a chave extra `}` no final do método `lerArquivo` é desnecessária e causa um erro de sintaxe. Aqui está a correção:

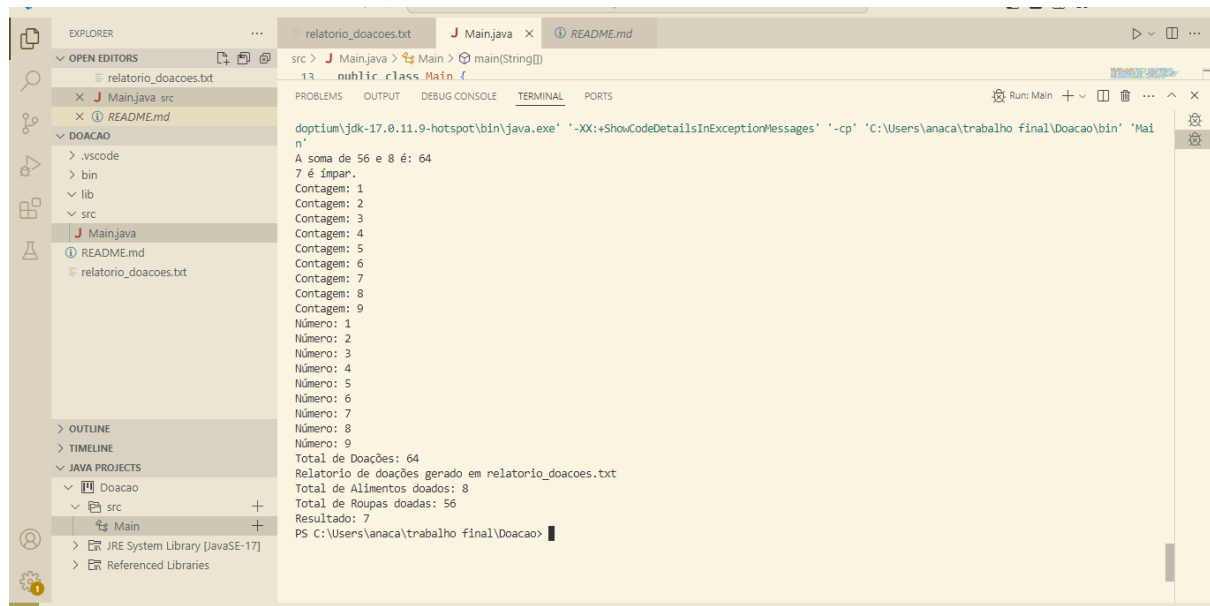
java

Copiar código

```
public static void lerArquivo(String caminho) {
    try (BufferedReader reader = new BufferedReader(new FileReader(caminho))) {
        String linha;
        while ((linha = reader.readLine()) != null) {
            System.out.println(linha);
        }
    } catch (IOException e) {
        System.out.println("Erro ao ler o arquivo: " + e.getMessage());
    }
}
```



Comprovação do código



```
src > J Main.java > Main > main(String[])
12 public class Main {

doptium\jdk-17.0.11.9-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\anaca\trabalho final\Doacao\bin' 'Mai
n'
A soma de 56 e 8 é: 64
7 é ímpar.
Contagem: 1
Contagem: 2
Contagem: 3
Contagem: 4
Contagem: 5
Contagem: 6
Contagem: 7
Contagem: 8
Contagem: 9
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
Número: 6
Número: 7
Número: 8
Número: 9
Total de Doações: 64
Relatorio de doações gerado em relatorio_doacoes.txt
Total de Alimentos doados: 8
Total de Roupas doadas: 56
Resultado: 7
PS C:\Users\anaca\trabalho final\Doacao>
```