



# CONTROLE DE VERSÃO - SISTEMA DE DOAÇÕES

## INFORMAÇÕES DO PROJETO

### Identificação

- **Nome do Sistema:** Sistema de Controle de Doações
- **Versão Atual:** 1.1.0 
- **Data de Início:** Agosto/2025
- **Última Atualização:** 01/Setembro/2025
- **Repositório:** <https://github.com/erikcamargo-max/sistema-doacoes-v1>
- **Stack Tecnológica:** Node.js + Express + SQLite + Vanilla JS
- **Ambiente:** Desenvolvimento/Produção Local
- **Status:**  100% OPERACIONAL

### Responsáveis

- **Desenvolvedor Principal:** Erik Camargo
- **Contato:** [definir email/telefone]
- **Cliente/Organização:** [definir nome da organização]

---

## ESTADO ATUAL DO SISTEMA (v1.1.0)

 Funcionalidades Implementadas e Funcionais

### 1. Gestão de Doadores

- ✓ Cadastro de doadores com validação ✓
- ✓ Código único automático (formato: D001-ABC) ✓ FUNCIONANDO
- ✓ Campos pessoais: Nome, CPF, Telefone 1, Telefone 2, Email ✓
- ✓ **Campos de endereço completo:** CEP, Logradouro, Número, Complemento, Bairro, Cidade, Estado ✓
- ✓ **Busca automática de CEP via ViaCEP API** ✓
- ✓ Detecção de duplicatas por CPF/Telefone ✓
- ✓ Listagem com busca e filtros ✓
- ✓ 15+ doadores ativos no sistema ✓

## 2. Gestão de Doações

- ✓ Registro de doações ✓
- ✓ **Tipos de pagamento: DINHEIRO e PIX** ✓
- ✓ Doações únicas e recorrentes ✓
- ✓ Vinculação automática doador-doação ✓
- ✓ **Edição completa de doações** ✓
- ✓ **Histórico de pagamentos funcional** ✓
- ✓ **Adicionar/Excluir pagamentos** ✓
- ✓ Parcelas futuras para recorrentes ✓

## 3. Interface do Usuário

- ✓ Dashboard com cards de resumo ✓
- ✓ Tabela responsiva com ações ✓
- ✓ Modal de cadastro com endereço completo ✓
- ✓ **Modal de edição totalmente funcional** ✓
- ✓ **Modal de histórico de pagamentos** ✓
- ✓ Filtros (tipo, recorrência, busca) ✓

- ✓ Indicadores visuais (badges, cores) ✓
- ✓ **Indicador visual de busca CEP** (amarelo/verde/vermelho) ✓

#### 4. Banco de Dados

- ✓ 4 tabelas principais estruturadas ✓
- ✓ **14 campos na tabela doadores** (incluindo endereço) ✓
- ✓ Índices únicos para CPF e código ✓
- ✓ Relacionamentos com chaves estrangeiras ✓
- ✓ Scripts de inicialização e upgrade ✓
- ✓ **Função checkPossibleDuplicates corrigida** ✓

#### 2. Gestão de Doações

- ✓ Registro de doações (Dinheiro/Produto/Serviço)
- ✓ Doações únicas e recorrentes
- ✓ Vinculação automática doador-doação
- ✓ Histórico de pagamentos
- ✓ Parcelas futuras para recorrentes

#### 3. Interface do Usuário

- ✓ Dashboard com cards de resumo
- ✓ Tabela responsiva com ações
- ✓ Modal de cadastro simplificado
- ✓ Filtros (tipo, recorrência, busca)
- ✓ Indicadores visuais (badges, cores)

#### 4. Banco de Dados

- ✓ 4 tabelas principais (doadores, doacoes, historico\_pagamentos, parcelas\_futuras)

- ✓ Índices únicos para CPF e código
- ✓ Relacionamentos com chaves estrangeiras
- ✓ Scripts de inicialização e upgrade

## ⚠ **Funcionalidades Parciais**

### **1. Gráficos do Dashboard**

- Estado: Placeholder implementado
- Função `createCharts()` vazia
- Arquivo: `app.js`

### **2. Exportação de Dados**

- Estado: Botão existe, função vazia
- Função `exportData()` não implementada
- Arquivo: `app.js`

## ● **Funcionalidades Não Implementadas**

- 1. Geração de Carnê PDF** - `generateCarne()` apenas alert
  - 2. Autenticação/Login** - Sistema sem controle de acesso
  - 3. Backup Automático** - Sem rotina de backup
  - 4. Relatórios Avançados** - Apenas resumo básico
  - 5. Notificações automáticas** - Sem sistema de alertas
-

## BUGS CONHECIDOS

### ✅ Bugs Corrigidos (v1.1.0)

1. **[RESOLVIDO] Função checkPossibleDuplicates**
  - Usava `res` sem ter no escopo
  - Função não fechava corretamente
  - Corrigido no server.js reescrito
2. **[RESOLVIDO] Campos de endereço não carregavam na edição**
  - Query SQL incompleta
  - Corrigido com todos os campos no SELECT
3. **[RESOLVIDO] ViaCEP não funcionava**
  - Função mal implementada
  - Corrigido com nova implementação


### Baixos (Não Críticos)

1. **Console.logs em produção** - Múltiplos logs de debug ativos
  2. **Modal de histórico duplicado no HTML** - Não afeta funcionamento
  3. **Validação de CPF** - Apenas formatação, sem validação de dígitos verificadores
- 








## SEGURANÇA E VALIDAÇÕES

### Implementadas

- ✅ Escape básico de SQL injection (parametrização)
- ✅ Validação de campos obrigatórios

-  Verificação de duplicatas

## Pendentes

-  Autenticação e autorização
  -  Rate limiting
  -  HTTPS
  -  Validação de CPF (algoritmo)
  -  Sanitização de inputs no frontend
  -  Tokens CSRF
  -  Logs de auditoria
- 

## DEPENDÊNCIAS

### Produção

```
json
{
  "express": "^4.18.2",
  "sqlite3": "^5.1.6",
  "cors": "^2.8.5",
  "body-parser": "^1.20.2"
}
```

### Desenvolvimento

```
json
```

```
{  
  "nodemon": "^3.0.1"  
}
```

## Frontend (CDN)

- TailwindCSS 2.2.19
- Feather Icons 4.28.0
- Chart.js 3.9.1

## ESTRUTURA DO BANCO DE DADOS

### Tabela: doadores (14 campos)

Campo	Tipo	Constraints	Descrição
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	ID único
nome	TEXT	NOT NULL	Nome completo
email	TEXT	-	Email opcional
telefone1	TEXT	NOT NULL	Telefone principal
telefone2	TEXT	-	Telefone secundário
cpf	TEXT	UNIQUE (quando não nulo)	CPF sem formatação
codigo_doador	TEXT	UNIQUE	Código visível (D001-ABC)
<b>cep</b>	TEXT	-	CEP (00000-000)
<b>logradouro</b>	TEXT	-	Rua, Avenida, etc
<b>numero</b>	TEXT	-	Número do endereço
<b>complemento</b>	TEXT	-	Apto, Bloco, Sala

Campo	Tipo	Constraints	Descrição
<b>bairro</b>	TEXT	-	Nome do bairro
<b>cidade</b>	TEXT	-	Nome da cidade
<b>estado</b>	TEXT	-	UF (2 caracteres)
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Data de criação

Tabela: doacoes (Tipos: Dinheiro, PIX)

Campo	Tipo	Constraints	Descrição
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	ID único
doador_id	INTEGER	FOREIGN KEY → doadores.id	Referência ao doador
valor	REAL	NOT NULL	Valor da doação
tipo	TEXT	NOT NULL	<b>Dinheiro ou PIX apenas</b>
data_doacao	TEXT	NOT NULL	Data no formato ISO
recorrente	INTEGER	DEFAULT 0	0=única, 1=recorrente
observacoes	TEXT	-	Notas adicionais
parcelas_totais	INTEGER	DEFAULT 1	Total de parcelas
data_proxima_parcela	TEXT	-	Data da próxima parcela
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Data de criação

Tabela: historico\_pagamentos

Campo	Tipo	Constraints	Descrição
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	ID único
doacao_id	INTEGER	FOREIGN KEY → doacoes.id	Referência à doação
data_pagamento	TEXT	NOT NULL	Data do pagamento



Campo	Tipo	Constraints	Descrição
valor	REAL	NOT NULL	Valor pago
status	TEXT	DEFAULT 'Pago'	Status do pagamento
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Data de criação

**Tabela: parcelas\_futuras**

Campo	Tipo	Constraints	Descrição
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	ID único
doacao_id	INTEGER	FOREIGN KEY → doacoes.id	Referência à doação
numero_parcela	INTEGER	-	Número da parcela
data_vencimento	TEXT	NOT NULL	Data de vencimento
valor	REAL	NOT NULL	Valor da parcela
status	TEXT	DEFAULT 'Pendente'	Status da parcela
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Data de criação

## ROADMAP DE DESENVOLVIMENTO

### Fase 1: Correções Urgentes (Sprint 1 - 1 semana)

- ☐ Corrigir variável truncada no app.js
- ☐ Remover duplicação do modal de histórico
- ☐ Implementar interface de verificação de duplicatas
- ☐ Adicionar validação completa de CPF

## **Fase 2: Funcionalidades Essenciais (Sprint 2-3 - 2 semanas)**

- ☐ Implementar edição de doações
- ☐ Completar histórico de pagamentos
- ☐ Implementar exportação (CSV/Excel)
- ☐ Adicionar gráficos funcionais

## **Fase 3: Segurança (Sprint 4 - 1 semana)**

- ☐ Sistema de autenticação
- ☐ Níveis de acesso (admin/operador)
- ☐ Logs de auditoria
- ☐ Backup automático

## **Fase 4: Recursos Avançados (Sprint 5-6 - 2 semanas)**

- ☐ Geração de carnês PDF
- ☐ Relatórios personalizados
- ☐ Dashboard analytics avançado
- ☐ Notificações de vencimento









## **Fase 5: Otimizações (Sprint 7 - 1 semana)**

- ☐ Paginação server-side
  - ☐ Cache de consultas
  - ☐ Compressão de assets
  - ☐ PWA (offline support)
-

## HISTÓRICO DE VERSÕES

**v1.1.0 (01/Setembro/2025)  ATUAL**

**Tipo:** Minor Release - Novas Funcionalidades **Mudanças:**

-  Implementada edição completa de doações
-  Implementado histórico de pagamentos funcional
-  Adicionados campos de endereço (7 novos campos)
-  Integração com API ViaCEP para busca automática
-  Tipos de pagamento simplificados: Dinheiro e PIX
-  Modal de histórico com adicionar/excluir pagamentos
-  Correção da função checkPossibleDuplicates
-  server.js completamente reescrito e otimizado

**Novos Scripts Criados:**

```
bash
```

```
implementar-edicao-historico.js # Implementou edição e histórico
```

```
adicionar-campos-endereco.js   # Adicionou campos de endereço
```

```
corrigir-viacep-edicao.js       # Corrigiu busca CEP
```

```
ajustar-edicao-endereco-tipos.js # Ajustou tipos de pagamento
```

**Estrutura do Banco Atualizada:**







- Tabela doadores: 14 campos (7 novos de endereço)
- Tipos aceitos: Dinheiro e PIX apenas

**Status:**  SISTEMA 100% FUNCIONAL

---

## v1.0.1 (31/Agosto/2025)

**Tipo:** Correção Crítica (Hotfix) **Mudanças:**

-  Corrigido erro SQLITE\_ERROR: coluna codigo\_doador não existia
-  Removida linha truncada no app.js (linha 634)
-  Adicionadas colunas faltantes: codigo\_doador e cpf
-  Gerados códigos automáticos para 15 doadores existentes
-  Criados índices para otimização
-  Criados scripts de reparo: repair.js e fix-codigo-doador.js

**Scripts de Correção Aplicados:**

```
bash  
  
node repair.js           # Correção geral do sistema  
node fix-codigo-doador.js # Correção específica do banco
```

**Status:**  SISTEMA 100% OPERACIONAL

---

## v1.0.0 (Agosto/2025)

**Tipo:** Release Inicial **Mudanças:**

- Sistema base implementado
- CRUD de doadores e doações
- Interface responsiva

- Detecção de duplicatas

### Problemas Conhecidos:

- Modal de duplicatas incompleto
  - Funcionalidades de edição pendentes
  - Sem autenticação
- 



## PROCEDIMENTOS DE DEPLOY

### Desenvolvimento Local

```
bash
```

```
# Instalar dependências
```

```
npm install
```

```
# Inicializar banco (primeira vez)
```

```
npm run init-db
```

```
# Upgrade do banco (adicionar campos)
```

```
npm run upgrade-db
```

```
# Iniciar servidor
```

```
npm start # Produção na porta 3001
```

```
npm run dev # Dev com nodemon
```

### Produção

```
bash
```

*# 1. Fazer backup do banco*

```
cp database/doacoes.db database/backup_$(date +%Y%m%d).db
```

*# 2. Pull das alterações*

```
git pull origin main
```

*# 3. Instalar/atualizar dependências*

```
npm install --production
```

*# 4. Aplicar migrations se houver*

```
npm run upgrade-db
```

*# 5. Reiniciar servidor*

```
pm2 restart sistema-doacoes
```

---

## CONFIGURAÇÕES E VARIÁVEIS

### Servidor

- **Porta:** 3001 (hardcoded)
- **CORS:** Habilitado para todas origens
- **Body Limit:** Padrão Express (~100kb)

### Banco de Dados

- **Tipo:** SQLite3
- **Arquivo:** `./database/doacoes.db`
- **Modo:** Serializado

- **Backup:** Manual necessário

## Frontend

- **API Base:** `/api` (relativo)
  - **Timeout:** Não configurado
  - **Cache:** Desabilitado
- 

## MÉTRICAS E MONITORAMENTO

### KPIs do Sistema

- Total de doadores cadastrados
- Total de doações registradas
- Valor total arrecadado
- Taxa de doações recorrentes
- Média de valor por doação

### Pontos de Monitoramento Sugeridos

#### 1. Performance

- Tempo de resposta das APIs
- Tamanho do banco de dados
- Uso de memória do Node.js

#### 2. Negócio

- Doações por período
- Taxa de inadimplência

- Doadores mais ativos

### 3. Erros

- Falhas de validação
- Tentativas de duplicação
- Erros de banco de dados

---

## SCRIPTS ÚTEIS

### Backup Manual

```
bash

#!/bin/bash
# backup.sh
BACKUP_DIR="/backups"
DB_FILE="/database/doacoes.db"
TIMESTAMP=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR
cp $DB_FILE "$BACKUP_DIR/doacoes_$TIMESTAMP.db"
echo "Backup criado: doacoes_$TIMESTAMP.db"
```

### Limpeza de Logs

```
bash

# Remover console.logs em produção
sed -i '/console\.log/d' public/app.js
```



## Verificação de Integridade

```
sql

-- check_integrity.sql

SELECT 'Doadores sem código' as check_type, COUNT(*) as count
FROM doadores WHERE codigo_doador IS NULL;

SELECT 'Doações órfãs' as check_type, COUNT(*) as count
FROM doacoes WHERE doador_id NOT IN (SELECT id FROM doadores);

SELECT 'Pagamentos órfãos' as check_type, COUNT(*) as count
FROM historico_pagamentos WHERE doacao_id NOT IN (SELECT id FROM doacoes);
```

---

## SUPORTE E CONTATOS

### Desenvolvimento

- **Git Issues:** [Criar issue no GitHub](#)
- **Email:** [definir email de suporte]

### Emergências

- **Servidor caiu:** Verificar logs em `./logs/`
  - **Banco corrompido:** Restaurar último backup
  - **Perda de dados:** Verificar backups automáticos
-

## ✓ CHECKLIST DE MANUTENÇÃO

### Diário

- ☐ Verificar logs de erro
- ☐ Monitorar espaço em disco
- ☐ Conferir backups

### Semanal

- ☐ Backup completo do sistema
- ☐ Análise de performance
- ☐ Revisão de doações pendentes

### Mensal

- ☐ Limpeza de logs antigos
  - ☐ Otimização do banco (VACUUM)
  - ☐ Relatório de métricas
  - ☐ Atualização de dependências
- 

## 🌌 CONSIDERAÇÕES FUTURAS

### Escalabilidade

- Migração para PostgreSQL quando > 10GB
- Implementar cache Redis para > 1000 usuários
- CDN para assets estáticos

## Integrações Possíveis

- Gateway de pagamento (PagSeguro/Stripe)
- WhatsApp API para lembretes
- Google Sheets para relatórios
- Sistema contábil

## Melhorias UX







- Dark mode
- Atalhos de teclado
- Tour guiado para novos usuários
- Personalização de campos

---

**Última Atualização:** 01/Setembro/2025 **Próxima Revisão:** Setembro/2025 **Documento Versão:** 1.1.0

## NOTAS DA VERSÃO 1.1.0

### Principais Melhorias Implementadas:

-  **Edição completa** de doações funcionando
-  **Histórico de pagamentos** com interface visual
-  **Endereços completos** com 7 novos campos
-  **Busca automática de CEP** via ViaCEP API
-  **Tipos simplificados** para Dinheiro e PIX apenas
-  **server.js reescrito** sem erros de sintaxe

## Estatísticas do Sistema:

- 📁 **15+ doadores** cadastrados
- 💳 **2 tipos de pagamento** (Dinheiro/PIX)
- 📌 **14 campos** na tabela doadores
- 🛠️ **10+ scripts** de manutenção criados
- ✅ **100% funcional** em produção

## Scripts de Manutenção Disponíveis:

1. **repair.js** - Reparo geral do sistema
2. **fix-codigo-doador.js** - Correção de códigos
3. **implementar-edicao-historico.js** - Adiciona edição
4. **adicionar-campos-endereco.js** - Adiciona endereços
5. **corrigir-viacep-edicao.js** - Corrige busca CEP
6. **ajustar-edicao-endereco-tipos.js** - Ajusta tipos

## Comando de Backup Recomendado:

```
bash

# Backup completo do sistema v1.1.0
cp database/doacoes.db backups/doacoes_v1.1.0_$(date +%Y%m%d).db
cp server.js backups/server_v1.1.0_$(date +%Y%m%d).js
cp public/app.js backups/app_v1.1.0_$(date +%Y%m%d).js
```

## Próximas Implementações Sugeridas:

1. **Gráficos funcionais** no dashboard

2. **Exportação CSV/Excel** dos dados
3. **Geração de carnês PDF**
4. **Sistema de autenticação**
5. **Backup automático agendado**