

CONTRIBUTING.md – StonksView

Bem-vindo(a) ao Guia de Contribuição do StonksView!

Obrigado por dedicar seu tempo para conhecer e colaborar com o **StonksView**, um projeto criado para ajudar estudantes a organizarem melhor sua vida financeira. Este guia foi elaborado para garantir que qualquer pessoa – mesmo sem experiência prévia no projeto – consiga configurar o ambiente, compreender a estrutura e contribuir de forma eficiente e segura.

Ficamos muito felizes por ter você aqui. Vamos começar!

Objetivo deste guia

Este documento explica, de forma clara e objetiva, como montar o ambiente de desenvolvimento do StonksView e como contribuir corretamente com o projeto, mesmo que você nunca tenha trabalhado com ele antes.

O foco é garantir que qualquer pessoa consiga:

1. Instalar dependências
2. Rodar o projeto sem erros
3. Entender a estrutura básica
4. Contribuir seguindo boas práticas

Antes de começar

Para trabalhar no StonksView, você precisa ter instalado:

- Python 3.10 ou superior
- Pip (vem junto com Python)
- Git
- Node.js + npm (caso trabalhe no front-end)
- Virtualenv (opcional, recomendado)

Como verificar sua versão e se estão instalados os aplicativos necessários:

No terminal de seu computador digite:

```
python --version  
git --version  
node -v  
npm -v
```

Caso algum dos comandos acima não funcionarem, instale o software correspondente

Sistemas operacionais suportados:

- Windows 10+
- Ubuntu 20.04+
- macOS 11+

1 - Obtendo o código do projeto

Passo 1 - Clone o repositório usando o comando:

```
git clone https://github.com/SEU-USUARIO/stonksview.git
```

Passo 2 - Entre na pasta correspondente:

```
cd stonksview
```

2 - Montando o ambiente no Python

Passo 1 - Crie o ambiente virtual:

```
python -m venv venv
```

Passo 2 - Ative o ambiente virtual:

Windows: venv\Scripts\activate

Linux/Mac: source venv/bin/activate

Passo 3 - Instalar dependências:

```
pip install -r requirements.txt
```

3 - Configurando o banco de dados

O StonksView usa Django ORM, então você só precisa aplicar migrações:

```
python manage.py migrate
```

Criar usuário administrador (opcional, recomendado):

```
python manage.py createsuperuser
```

4 - Rodando o projeto

Inicie o servidor backend:

```
python manage.py runserver
```

O sistema ficará disponível em:

<http://127.0.0.1:8000/>

Se houver front-end separado:

```
cd frontend  
npm install  
npm start
```

5 - Estrutura do projeto

```
stonksview/  
    ├── stonksview/      # Configurações principais do Django  
    ├── core/            # Lógica de cadastro, login e transações  
    ├── templates/       # HTML do projeto  
    ├── static/          # CSS, JS, imagens  
    ├── manage.py        # Comando principal  
    └── requirements.txt # Dependências
```

6 - Padrão de estilo de código

Siga estes princípios:

Python:

- Estilo PEP8
- Nomes de variáveis e funções em `snake_case`
- Classes em `CamelCase`
- Use 4 espaços para indentação
- Linhas com até 79 caracteres

HTML/CSS:

- HTML semântico
- Classes organizadas
- Evitar CSS inline

Git:

- Commits pequenos
- Mensagens claras
- Evitar “commit gigante de tudo”

7 - Como contribuir

Passo 1 - Criar um branch para a sua atualização:

```
git checkout -b feature/nome-da-sua-feature
```

Observação: use nomes claros como `feature/tela-de-login`

Passo 2 - Fazer commits pequenos e objetivos:

Padrão recomendado:

tipo: descrição curta

Exemplos:

feat: adiciona validação no cadastro

fix: corrige erro ao excluir transação

refactor: organiza funções de autenticação

style: ajusta layout da página inicial

Passo 3 - Enviar atualizações para o GitHub:

```
git add .
```

```
git commit -m "feat: adiciona página de relatório"
```

```
git push origin feature/nome-da-sua-feature
```

Passo 4 - Criar um Pull Request:

Inclua:

- Objetivo da mudança
- Como testar
- Prints (se aplicável)
- Explicação se algo sensível foi alterado

8 - Checklist do contribuidor

Antes de abrir seu PR, verifique:

- O projeto rodou localmente sem erros
- A nova funcionalidade foi testada
- Nenhuma funcionalidade existente foi quebrada
- O código segue o padrão do projeto
- Commits claros e bem organizados
- Branch criada corretamente

9 - Código de conduta

Por favor siga as regras abaixo

- Seja educado
- Mantenha comunicação clara
- Siga os padrões do projeto
- Pergunte antes de grandes mudanças

Obrigado por contribuir com o StonksView!

Se quiser ajuda com outra parte do projeto ou se está com dúvidas abra uma issue
ou envie uma mensagem.