

# Capa

Título do Projeto: Cidadão Ativo

Nome do Estudante: Guilherme Melato

Curso: Engenharia de Software.

Data de Entrega: ?

---

## Resumo

Esse documento contém informações sobre o projeto Cidadão Ativo, que tem como objetivo principal aproximar o cidadão dos assuntos que surgem na câmara de vereadores, fazendo-o participar mais ativamente das pautas propostas e ter a noção do que está acontecendo na cidade. Aqui será levantada a melhor maneira de realizar isso, para que seja uma aplicação eficiente, objetiva e de fácil acesso, para todas as idades.

---

## 1. Introdução

**Contexto:** A necessidade por um meio onde os cidadãos pudessem estar a par do que os vereadores criam e propõe para a cidade foi a principal causa do início desse projeto. Com tal ferramenta, as pessoas poderiam não somente contribuir para decisões, mas estarem cientes do que está sendo feito, ou o que não está.

**Justificativa:** Hoje não há nada relacionado a isso nos aplicativos de apps que seja eficiente e objetivo para aproximar as pessoas que moram nas cidades dos projetos que são propostos pelos vereadores. Com este software, uma possibilidade será aberta para inovar cada vez mais nesse tipo de área e trazer mais consciência em quem votar e no que o dinheiro de impostos é gasto.

**Objetivos:** O objetivo principal é levar informação sobre as pautas que surgem na câmara de vereadores da cidade onde as pessoas moram de

forma eficiente, objetiva e clara, e não somente isso, mas torná-las participantes das decisões, de forma indireta.

---

## 2. Descrição do Projeto

**Tema do Projeto:** Cidadão Ativo (?) será uma aplicação que haverá dois tipos de usuários principais, além dos administradores: os vereadores, que poderão adicionar projetos que realizarão na cidade, como asfaltamento ou limpeza de uma rua, criação de emendas e projetos de leis; e os cidadãos, que poderão acessar a aplicação, filtrar por bairro ou tipo de projeto, e votar com um *upvote* ou *downvote* e definir as postagens mais votadas.

**Problemas a Resolver:** Essa aplicação visa resolver a questão da falta de acesso aos projetos que são aprovados pelos vereadores, mantendo turva a visão crítica que temos daqueles em quem votamos e dificultando a hora de clicar os números nas urnas. Isso fará o povo participar, mesmo que de forma indireta, nas decisões de projetos de leis, emendas e demais realizações nos mais diversos bairros.

### Limitações:

- Escalabilidade: O projeto inicial será feito para apenas uma cidade, podendo ter dificuldade se pessoas de outras cidade acessem e se cadastrem.
  - Inclusão e acessibilidade: Nem todos as pessoas têm acesso à internet ou são habilitadas fisicamente para o uso apropriado do software.
  - Abrangência de funcionalidades: Para ser algo objetivo e disponível para todas as idades, a implementação de funções complexas pode ser complexa.
  - Dependência da Adesão: O sucesso da aplicação depende da adesão de vereadores e da população de forma ativa.
  - Moderação dos dados: Tornar os números de votos e projetos criados autênticos para a confiança dos usuários.
- 

## 3. Especificação Técnica

## 3.1 Requisitos de Software

### Requisitos Funcionais:

- O sistema deve permitir que o usuário acesse sua conta utilizando CPF e senha.
- O sistema deve permitir que os usuários filtrem publicações por bairro desejado.
- O sistema deve permitir que os usuários votem em publicações utilizando upvote para apoiar e downvote para discordar.
- O sistema deve permitir que o **Vereador** e o **Cidadão** enviem propostas de melhorias para os Administradores.
- O sistema deve permitir que o **Administrador** possa editar publicações, incluindo exclusão das mesmas e edição de texto.
- O sistema deve permitir que o **Administrador** possa criar e editar dashboards.
- O sistema deve permitir que o **Vereador** e o **Cidadão** tenham acesso aos dashboards.
- O sistema deve permitir que o **Vereador** possa criar novos projetos, mas sem editar ou excluir projetos já existentes.
- O sistema deve permitir que o **Vereador** tenha acesso aos resultados do survey.
- O sistema deve permitir que o **Cidadão** tenha acesso ao survey de Interesse da Comunidade.
- O sistema deve permitir que os **Cidadãos** e **Vereadores** denunciem publicações inadequadas para revisão dos Administradores.
- O sistema deve permitir que os **Administradores** gerenciem os perfis de usuários, podendo bloquear ou restringir acessos quando necessário.
- O sistema deve exibir um ranking das publicações mais votadas dentro de um período específico.
- O sistema deve permitir que os **Vereadores** justifiquem seus projetos com descrições detalhadas e anexos.
- O sistema deve gerar relatórios sobre o número de interações, votos e propostas enviadas, acessíveis para **Administradores** e **Vereadores**.

### Requisitos Não-Funcionais:

- O sistema deve ser capaz de aguentar uma quantidade considerável de usuários (Com base na quantidade de uma cidade).
- O sistema deve **processar uma votação** em no máximo 1 segundo

- Todas as senhas dos usuários devem ser armazenadas utilizando **criptografia SHA-256** ou superior.
- O sistema deve estar em conformidade com a **LGPD (Lei Geral de Proteção de Dados)**, garantindo o consentimento do usuário para coleta de dados.
- As ações administrativas (edição, exclusão, bloqueio de usuários) devem ser **registradas em logs de auditoria**.
- As cores e contrastes do sistema devem permitir **leitura confortável** para usuários com baixa visão.
- O backend do sistema deve utilizar **NestJs**, e o banco de dados deve ser **MySQL**

## 3.2 Considerações de Design

### Escolhas de Design:

- Foi selecionada uma aplicação PWA à uma aplicação mobile, pois o custo aumentaria muito para disponibilizar para os diferentes SO mobile existentes, podendo alcançar mais usuários.
- A interface será minimalista e com filtros claros para disponibilizar o conteúdo de forma direta, facilitando o uso para todas as idades, especialmente para usuários de mais idade.

### Visão Inicial de Arquitetura

- frontend: Interface web para os cidadãos (visualizar projetos, votar, filtrar), vereadores (cadastrar projetos) e admins (criar e editar dashboards, editar os projetos, aprovar projetos), além de enviar requisições HTTP para o backend
- backend: API RESTful que gerencia solicitações dos usuários (cadastros, votações e filtros), e irá validar as requisições enviadas pelo front através dos módulos de autenticação e moderação
- Banco de Dados: Banco relacional para guardar as informações de usuários, projetos e dashboards
- Módulo de Autenticação: Gerencia o login e o tempo de sessão dos usuários no geral
- Módulo de Moderação: Permite que os moderadores aprovem projetos, criem e editem dashboards e editem os projetos

### Padrões de Arquitetura

- O método de Cliente-Servidor será utilizado, permitindo uma boa, eficiente e simples comunicação entre o cliente (frontend) e o servidor (backend)
- O padrão MVC será adotado para organizar a lógica de negócios facilitando a manutenção.

### 3.3 Stack Tecnológica

#### **Linguagens de Programação:**

- Como linguagem de programação principal, será utilizado o TypeScript, pois é uma linguagem robusta, confiável por causa da sua tipagem estática opcional e manutenibilidade quando a aplicação é escalada.

#### **Frameworks e bibliotecas:**

- Será usado o NestJS como framework principal para a construção do sistema, além de bibliotecas de autenticação
- Serão utilizadas as bibliotecas *common*, *typeorm*, *jwt*, *swagger* e *config*.

#### **Ferramentas de Desenvolvimento e Gestão de Projeto:**

- As ferramentas a serem utilizadas serão o VSCode, MySql e ClickUp (para gerenciamento do projeto).

### 3.4 Considerações de Segurança

- Acessos não autorizados: Implementar uma boa autenticação, sendo uma das possibilidades entrar usando o GOV.br.
- Manipulação dos votos: Mitigada ao definir somente um voto por usuário.
- Propostas validadas: Integridade das propostas criadas validadas pelos logins dos Administradores cadastrados no banco de dados.
- Uso indevido da Plataforma: Monitorar as postagens através de logs no BD

## 4. Próximos passos

- Desenvolver o mockup da aplicação web

- Iniciar a parte de autenticação no backend

## 5. Referências

- <https://newsletter.pragmaticengineer.com/p/software-engineering-rfc-and-design>
- <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>
- <https://www.simform.com/blog/software-architecture-patterns/>

## 7. Avaliação dos Professores