

**PROJETO DE ENGENHARIA DE SOFTWARE**

**DOCUMENTAÇÃO PD - FALA UFS**

**São Cristóvão - SE**

**2020**

**Projeto de Engenharia de Software**

**Documentação PD Fala UFS**

Documentação do projeto de engenharia do software da plataforma Plano Diretor - Fala UFS, realizado como parte do relatório parcial para os projetos de pesquisa e extensão citados, desenvolvidos em parceria multidepartamental na Universidade Federal de Sergipe.

**ORIENTADOR: Prof. Dr. Rogério Patrício Chagas do Nascimento**

**COORIENTADOR: Prof. Dr. Márcio da Costa Pereira**

**São Cristóvão – SE**

**2020**

**SUMÁRIO**

[**INTRODUÇÃO**](#_j7rrcmsmpla4) **4**

[**DOCUMENTO DE VISÃO DO SISTEMA**](#_yvwjoi8gcibd) **4**

[**REGRAS DE NEGÓCIO**](#_pyy7d8ky1e7r) **5**

[3.1 Usuários](#_cjkkqvo9vr5q) 5

[3.2 Requisitos Funcionais](#_nsm0en3bz0rm) 6

[3.3 Requisitos Não-Funcionais](#_sku8j548lob4) 8

[**DIAGRAMA DE CASOS DE USO**](#_ixtsz4lft4v) **9**

[**DIAGRAMA ENTIDADE RELACIONAMENTO**](#_342nxdy7tsjh) **10**

[**SCRIPT DE CRIAÇÃO DE BASE**](#_hm2i15k66bdd) **10**

[**DESCRIÇÃO DAS TABELAS**](#_vavvr1f9zcmt) **16**

[**DOCUMENTAÇÃO DA API IMPLEMENTADA**](#_wnffgl3dsevh) **20**

### 

### **INTRODUÇÃO**

O presente documento apresenta o PD Fala UFS, que constitui em uma plataforma para realização de audiências públicas assíncronas à nível acadêmico/universitário e para gestão e auxílio na tomada de decisões relativas às informações coletadas nestas audiências públicas. Atualmente as audiências públicas da Universidade Federal de Sergipe, no que concerne à todos os seus campi, se dão de forma síncrona e presencial, dois fatores fortemente limitantes em tempo e espaço para que se alcance uma representatividade justa e democrática. Os maiores interessados nas pautas e decisões tomadas pela universidade não se vêem ou sentem representados, e seus interesses não são sabidos pelos gestores das decisões internas da universidade que impactam diretamente sobre toda a sua comunidade (docentes, discentes, servidores, terceirizados e comunidade externa).

A fim de se contemporizar este processo através de intervenção tecnológica, a proposta é desenvolver uma plataforma web e mobile que se adapte e se incorpore gradualmente aos sistemas já existentes na estrutura jurídica da UFS, mas que venha quebrar paradigmas no que tange ao alcance populacional das comunidades envolvidas direta ou indiretamente nas decisões políticas tomadas pela gestão da Universidade, buscando a democratização da informação e da decisão.

Para facilitar o planejamento da arquitetura do software e a modelagem do seu banco de dados, os requisitos iniciais levantados bem como as propostas de interface são definidos abaixo em tópicos:

### **DOCUMENTO DE VISÃO DO SISTEMA**

|  |  |
| --- | --- |
| **Nome do Produto:** *Fala UFS* | **Público-alvo:** *Comunidade Universitária da UFS.* |
| **Problema/Necessidade:** *Criar um espaço virtual onde seja possível a participação da comunidade em audiências públicas para coleta, discussão e fiscalização de demandas.* | **Características-chave:**   * Permite a participação dos usuários a postar suas opiniões de forma pública e direcionada. * Permite que os gestores acompanhem as interações dos usuários com informações de apoio. * Permite que o processo de audiências públicas possa ser digitalizado e ampliado. Desde a fase inicial de coleta de demanda e propostas, passando para a votação e discussão. Finalizando com a fiscalização dos pontos do plano diretor. * Permitir os registros históricos de participação de audiências públicas ao longo do tempo. |
| **Condições Gerais de Satisfação (Critérios para aceitação):**   * Permitir a participação dos usuários através de texto, mídias e interações nas seções. * Permitir gerar informações para os gestores sobre as interações dos usuários. |
| **Diferenciais:**   * *Permitir criar audiências públicas assíncronas.* * *Permitir gerar informações a partir das participações na aplicação que ajudem os gestores na tomada de decisão.* |
| **Premissas, Riscos e Impedimentos:**   * *Quantidade de pessoas na equipe de desenvolvimento.* * *Problemas orçamentários.* * *Engajamento no projeto.* * *Especificação que não atende as necessidades.* |
| **Expectativas de Prazo:** 06 meses | **Expectativas de Custo:** *Não definido* |

### **REGRAS DE NEGÓCIO**

#### **3.1 Usuários**

|  |  |
| --- | --- |
| **Usuário** | **Descrição** |
| Gestor | Pessoa que irá gerenciar a aplicação e conseguirá extrair informações úteis a partir das interações dos usuários. |
| Cidadão | Aquele que irá interagir e alimentar o sistema de acordo com as configurações de participação definidas pelo gestor. Compreendem esse tipo de usuário os  Alunos, Servidores e Comunidade Externa |

#### 

#### **3.2 Requisitos Funcionais**

|  |  |
| --- | --- |
| **Requisitos Funcionais** | |
| **Código** | **Requisito** |
| RF01 | A aplicação deve possuir uma área com temas pré-definidos para coleta de necessidades gerais. |
| RF02 | A aplicação deve possuir uma área com proposituras relacionadas às audiências públicas. |
| RF03 | A aplicação deve permitir a votação (Sim/Não) no tema na parte das proposituras. |
| RF04 | A aplicação deve possuir uma área com pontos relacionadas ao plano diretor. |
| RF05 | Permitir o registro de participação com a publicação de textos e mídias de forma pública nas áreas temáticas, nas áreas de votação de tópicos e na área de acompanhamento do plano diretor. |
| RF06 | Permitir interagir com aprovação e reprovação nos comentários existentes nas áreas temáticas, nas áreas de votação de tópicos e na área de acompanhamento do plano diretor. |
| RF07 | Permitir a busca de conteúdo das participações de acordo com a área da aplicação. |
| RF08 | Permitir que os usuários se registrem no portal. |
| RF09 | Validação dos usuários com o registro no SIGAA, quando o mesmo apresentar um perfil ligado a UFS. |
| RF10 | Permitir gestores gerenciar usuários (Alunos, Egressos, Servidores e Comunidade Externa). |
| RF11 | Permitir a configuração de tempo de permissão de participação nas áreas da aplicação de acordo com os calendários de audiências públicas. |
| RF12 | Permitir o arquivamento das participações encerradas. |
| RF13 | Permitir a visualização das participações (ativas e arquivadas) |
| RF14 | Permitir o gestor o registro de temas para participações livres |
| RF15 | Permitir o gestor registrar as proposituras da parte de audiências públicas. |
| RF16 | Permitir o gestor transformar uma propositura da área de temas livre para a área de proposituras de audiência pública. |
| RF17 | Permitir o gestor registrar os pontos do plano diretor. |
| RF18 | Permitir o gestor publicar conteúdos informativos. |
| RF19 | Permitir o gestor configurar espaço para transmissão ao vivo de audiências públicas presenciais. |
| RF20 | Permitir o gestor visualizar gráficos que informem a participação dos usuários. |
| RF21 | Permitir o gestor visualização de gráficos com o temas e proposituras que possuem maior participação ou apoio. |
| RF22 | Permitir o gestor listar os usuários que possuem mais interações e apoio de interações. |
| RF23 | Permitir visualizar a quantidade de votos das áreas de votação. |
| RF24 | Efetuar Login na Aplicação. |
| RF25 | Visualizar conteúdos publicados pelos gestores. |
| RF26 | Permitir a visualização de participações encerradas |
| RF27 | Permitir a visualização do espaço de transmissão ao vivo |

#### 

#### **3.3 Requisitos Não-Funcionais**

|  |  |
| --- | --- |
| **Código** | **Requisito** |
| RNF01 | A aplicação deve ter pouco conteúdo textual e mais conteúdo visual. |
| RNF02 | As informações devem estar categorizadas por tópicos, de acordo com a área temática da manifestação, a fim de facilitar compreensão do usuário e sua navegação pelos conteúdos do sistema. |
| RNF03 | Devem haver filtros para exibir manifestações(por tema, mais votadas, menos visualizadas). |
| RNF04 | Aplicação deve ter tempo limite máximo de resposta para ação do usuário de 10 segundos. |
| RNF05 | Aplicação pode reconhecer fala de usuário e transcrever para texto. |
| RNF06 | Seguir as orientações da armazenamento de informações pessoais seguindo a Lei Geral de Proteção de Dados. |
| RNF07 | A aplicação deve ser construída de modo a garantir a acessibilidade. |
| RNF08 | A aplicação deve ser construída de modo a se integrar com outras soluções da UFS. |

#### 

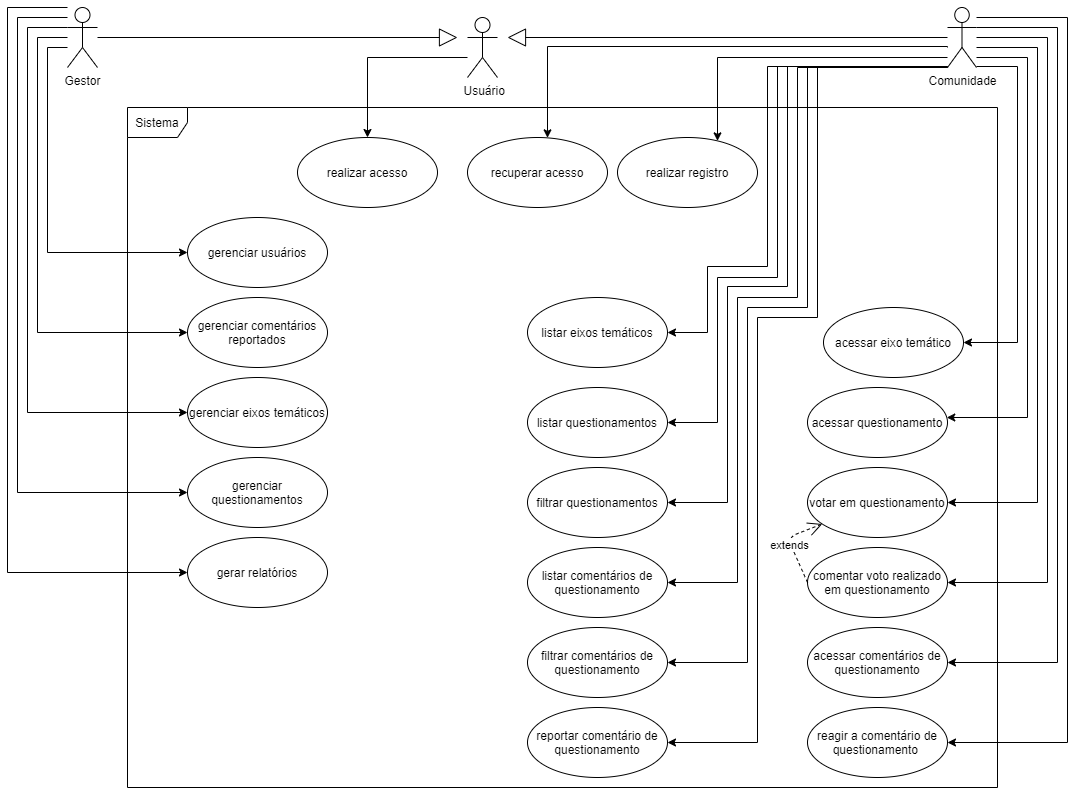
#### 

#### 

#### 

#### 

### **DIAGRAMA DE CASOS DE USO**



### **DIAGRAMA ENTIDADE RELACIONAMENTO**

##### 

### **SCRIPT DE CRIAÇÃO DE BASE**

|  |
| --- |
| --  -- ER/Studio Data Architect SQL Code Generation  -- Project : FalaUFSv3.DM1  --  -- Date Created : Saturday, June 20, 2020 17:30:33  -- Target DBMS : PostgreSQL 9.x  --  --  -- TABLE: "Agrupamento"  --  CREATE TABLE "Agrupamento"(  id\_agrupamento uuid NOT NULL,  titulo varchar(80) NOT NULL,  descricao text,  ativo char(1) NOT NULL,  visivel char(1) NOT NULL,  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL  )  ;  --  -- TABLE: "Eixo"  --  CREATE TABLE "Eixo"(  id\_eixo uuid NOT NULL,  logo varchar(700),  visivel char(1) NOT NULL,  titulo varchar(80) NOT NULL,  descricao text,  ativo char(1) NOT NULL,  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL  )  ;  --  -- TABLE: "Enquete"  --  CREATE TABLE "Enquete"(  id\_enquete uuid NOT NULL,  dt\_venc timestamp NOT NULL,  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL,  descricao text,  ativo char(1) NOT NULL,  visivel char(1) NOT NULL,  id\_eixo uuid,  pergunta varchar(300) NOT NULL,  id\_agrupamento uuid  )  ;  --  -- TABLE: "Moderacao"  --  CREATE TABLE "Moderacao"(  id\_tipo\_moderacao uuid NOT NULL,  id\_opiniao uuid NOT NULL,  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL  )  ;  --  -- TABLE: "Opiniao"  --  CREATE TABLE "Opiniao"(  id\_opiniao uuid NOT NULL,  comentario text,  negativos integer NOT NULL,  positivos integer NOT NULL,  visivel char(1) NOT NULL,  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL  )  ;  --  -- TABLE: "Tipo\_Moderacao"  --  CREATE TABLE "Tipo\_Moderacao"(  id\_tipo\_moderacao uuid NOT NULL,  titulo varchar(30) NOT NULL,  descricao varchar(80),  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL  )  ;  --  -- TABLE: "Usuario"  --  CREATE TABLE "Usuario"(  id\_usuario uuid NOT NULL,  cpf varchar(11) NOT NULL,  ativo char(1) NOT NULL,  tipo char(1) NOT NULL,  email varchar(80) NOT NULL,  nome varchar(80) NOT NULL,  telefone\_principal varchar(20),  senha varchar(20) NOT NULL,  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL  )  ;  --  -- TABLE: "Voto"  --  CREATE TABLE "Voto"(  id\_voto uuid NOT NULL,  dt\_inc timestamp NOT NULL,  id\_enquete uuid NOT NULL,  id\_voto\_valor uuid NOT NULL,  id\_usuario uuid NOT NULL,  id\_opiniao uuid  )  ;  --  -- TABLE: "Voto\_Valor"  --  CREATE TABLE "Voto\_Valor"(  id\_voto\_valor uuid NOT NULL,  descricao varchar(30) NOT NULL,  dt\_inc timestamp NOT NULL,  dt\_alt timestamp NOT NULL  )  ;  --  -- INDEX: "Ref213"  --  CREATE INDEX "Ref213" ON "Enquete"(id\_eixo)  ;  --  -- TABLE: "Agrupamento"  --  ALTER TABLE "Agrupamento" ADD  CONSTRAINT "PK16" PRIMARY KEY (id\_agrupamento)  ;  --  -- TABLE: "Eixo"  --  ALTER TABLE "Eixo" ADD  CONSTRAINT "PK2" PRIMARY KEY (id\_eixo)  ;  --  -- TABLE: "Enquete"  --  ALTER TABLE "Enquete" ADD  CONSTRAINT "PK4" PRIMARY KEY (id\_enquete)  ;  --  -- TABLE: "Moderacao"  --  ALTER TABLE "Moderacao" ADD  CONSTRAINT "PK20" PRIMARY KEY (id\_tipo\_moderacao, id\_opiniao)  ;  --  -- TABLE: "Opiniao"  --  ALTER TABLE "Opiniao" ADD  CONSTRAINT "PK18" PRIMARY KEY (id\_opiniao)  ;  --  -- TABLE: "Tipo\_Moderacao"  --  ALTER TABLE "Tipo\_Moderacao" ADD  CONSTRAINT "PK19" PRIMARY KEY (id\_tipo\_moderacao)  ;  --  -- TABLE: "Usuario"  --  ALTER TABLE "Usuario" ADD  CONSTRAINT "PK1" PRIMARY KEY (id\_usuario)  ;  --  -- TABLE: "Voto"  --  ALTER TABLE "Voto" ADD  CONSTRAINT "PK17" PRIMARY KEY (id\_voto)  ;  --  -- TABLE: "Voto\_Valor"  --  ALTER TABLE "Voto\_Valor" ADD  CONSTRAINT "PK15" PRIMARY KEY (id\_voto\_valor)  ;  --  -- TABLE: "Enquete"  --  ALTER TABLE "Enquete" ADD CONSTRAINT "RefEixo13"  FOREIGN KEY (id\_eixo)  REFERENCES "Eixo"(id\_eixo)  ;  ALTER TABLE "Enquete" ADD CONSTRAINT "RefAgrupamento31"  FOREIGN KEY (id\_agrupamento)  REFERENCES "Agrupamento"(id\_agrupamento)  ;  --  -- TABLE: "Moderacao"  --  ALTER TABLE "Moderacao" ADD CONSTRAINT "RefTipo\_Moderacao40"  FOREIGN KEY (id\_tipo\_moderacao)  REFERENCES "Tipo\_Moderacao"(id\_tipo\_moderacao)  ;  ALTER TABLE "Moderacao" ADD CONSTRAINT "RefOpiniao41"  FOREIGN KEY (id\_opiniao)  REFERENCES "Opiniao"(id\_opiniao)  ;  --  -- TABLE: "Voto"  --  ALTER TABLE "Voto" ADD CONSTRAINT "RefEnquete33"  FOREIGN KEY (id\_enquete)  REFERENCES "Enquete"(id\_enquete)  ;  ALTER TABLE "Voto" ADD CONSTRAINT "RefVoto\_Valor34"  FOREIGN KEY (id\_voto\_valor)  REFERENCES "Voto\_Valor"(id\_voto\_valor)  ;  ALTER TABLE "Voto" ADD CONSTRAINT "RefUsuario35"  FOREIGN KEY (id\_usuario)  REFERENCES "Usuario"(id\_usuario)  ;  ALTER TABLE "Voto" ADD CONSTRAINT "RefOpiniao36"  FOREIGN KEY (id\_opiniao)  REFERENCES "Opiniao"(id\_opiniao)  ; |

### **DESCRIÇÃO DAS TABELAS**

**Tabela Usuário**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_usuario | uuid | x |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| cpf | varchar(11) |  |  | NOT NULL | Número público de identificação de pessoa física, único por pessoa. |
| ativo | char(1) |  |  | NOT NULL | Valida se o usuário pode ou não acessar a aplicação |
| tipo | char(1) |  |  | NOT NULL | Flag de identificação do tipo de usuário  por padrão tipo 2,  (Tipo 1 = Administrador, Tipo 2 = Usuário padrão) |
| email | varchar(80) |  |  | NOT NULL | Email de cadastro do usuário, sendo como identificador de acesso. |
| nome | varchar(80) |  |  | NOT NULL | Nome civil da pessoa registrada |
| telefone\_principal | varchar(20) |  |  | NULL | Telefone de contato da pessoa |
| senha | varchar(20) |  |  | NOT NULL | Senha de acesso do usuário |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação do usuário, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração do usuário, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

**Tabela Enquete**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_enquete | uuid | x |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| dt\_venc | timestamp |  |  | NOT NULL | Data de vencimento da enquete |
| ativo | char(1) |  |  | NOT NULL | Valida se a enquete está ativa ou não para recebimento de votações. |
| descricao | text |  |  | NULL | Recebe a descrição da enquete |
| visível | char(1) |  |  | NOT NULL | Valida se a enquete está acessível para visualização |
| id\_eixo | uuid |  | x | NULL | Associação de enquete a um eixo |
| pergunta | varchar(300) |  |  | NOT NULL | Pergunta da enquete |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

**Tabela Eixo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_eixo | uuid | x |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| logo | varchar(700) |  |  | NULL | Caminho para a imagem da logo |
| ativo | char(1) |  |  | NOT NULL | Valida se o eixo está ativo ou não para recebimento de enquetes |
| descricao | text |  |  | NULL | Recebe a descrição do eixo |
| titulo | char(80) |  |  | NOT NULL | Titulo do eixo |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

**Tabela Agrupamento**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_agrupamento | uuid | x |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| titulo | varchar(80) |  |  | NOT NULL | Título de Agrupamento de eixos |
| descricao | text |  |  | NULL | Descrição dos eixos |
| ativo | char(1) |  |  | NOT NULL | Indica se está permitindo interações |
| visivel | char(1) |  |  | NOT NULL | Indica se está visível em listagens |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

**Tabela Voto\_Valor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_voto\_valor | uuid | x |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| descricao | varchar(30) |  |  | NOT NULL | Descrição da identificação da escala de votos |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

**Tabela Voto**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_voto | uuid | X |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| id\_enquete | uuid |  | X | NOT NULL | Id único gerado e gerenciado pelo banco de dados da tabela de enquete |
| id\_voto\_valor | uuid |  | X | NOT NULL | Id único gerado e gerenciado pelo banco de dados da tabela de voto\_valor |
| id\_usuario | uuid |  | X | NOT NULL | Id único gerado e gerenciado pelo banco de dados da tabela de usuário |
| id\_opiniao | uuid |  |  | NULL | Id único gerado e gerenciado pelo banco de dados de opinião |

**Tabela Opiniao**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_opiniao | uuid | x |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| comentario | text |  |  | NULL | Comentário relacionado a votos |
| negativos | integer |  |  | NOT NULL | Contador de interações negativas |
| positivos | integer |  |  | NOT NULL | Contador de interações positivas |
| visivel | char(1) |  |  | NOT NULL | Visibilidade da opinião |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

**Tabela Moderacao**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_tipo\_moderacao | uuid | x | x | NOT NULL | id único gerado e gerenciado pelo banco de dados de Tipo\_Moderacao |
| id\_opiniao | uuid | x | x | NOT NULL | id único gerado e gerenciado pelo banco de dados de Opiniao |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

**Tabela Tipo\_Moderacao**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Chave | Tipo | PK | FK | NULL | Descrição |
| id\_tipo\_moderacao | uuid | x |  | NOT NULL | Id único gerado e gerenciado pelo banco de dados |
| titulo | varchar(30) |  |  | NOT NULL | Título do tipo de moderação |
| descricao | varchar(80) |  |  | NULL | Descrição explicativa do que é a moderação. |
| dt\_inc | timestamp |  |  | NOT NULL | Data Hora de criação, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |
| dt\_alter | timestamp |  |  | NOT NULL | Data Hora de alteração, onde padrão é o now() <https://www.postgresqltutorial.com/postgresql-now/> |

### **DOCUMENTAÇÃO DA API IMPLEMENTADA**

**Usuários**

|  |  |
| --- | --- |
| GET api/usuarios | -- Puxar todos os usuários |
| GET api/usuarios/:id | -- Puxar dados de um id específico |
| POST api/usuarios | -- Criação de usuário |
| PUT api/usuarios/:id | -- Edição de usuário |
| DELETE api/usuarios/:id | -- Inativação Edição de usuário |

**Enquetes**

|  |  |
| --- | --- |
| GET api/enquetes | -- Puxar todos os enquetes |
| GET api/enquetes/:id | -- Puxar dados de um id específico |
| POST api/enquetes | -- Criação de enquetes |
| PUT api/enquetes/:id | -- Edição de enquetes |
| DELETE api/enquetes/:id | -- Ocultação caso haja interações e deleção caso não haja interação com outras tabelas. |

**Eixos**

|  |  |
| --- | --- |
| GET api/eixos | -- Puxar todos os eixos |
| GET api/eixos/:id | -- Puxar dados de um id específico |
| POST api/eixos | -- Criação de eixos |
| PUT api/eixos/:id | -- Edição de eixos |
| DELETE api/eixos/:id | -- Ocultação caso haja interações e deleção caso não haja interação com outras tabelas. |

**Opinião**

|  |  |
| --- | --- |
| GET api/opiniao | -- Puxa todas as opiniões |
| GET api/opiniao/:id | -- Puxar dados de um id específico |
| POST api/opiniao | -- Criação de opinião |
| PUT api/opiniao/:id | -- Edição de opinião |
| DELETE api/opiniao/:id | -- Inativação Edição de opinião |

**Agrupamento**

|  |  |
| --- | --- |
| GET api/agrupamento | -- Puxa todos os agrupamentos |
| GET api/agrupamento/:id | -- Puxar dados de um id específico |
| POST api/agrupamento | -- Criação de agrupamento |
| PUT api/agrupamento/:id | -- Edição de agrupamento |
| DELETE api/agrupamento/:id | -- Inativação Edição de agrupamento |

**Voto**

|  |  |
| --- | --- |
| GET api/voto | -- Puxa todos os voto |
| GET api/voto/:id | -- Puxar dados de um id específico |
| POST api/voto | -- Criação de voto |
| PUT api/voto/:id | -- Edição de voto |
| DELETE api/voto/:id | -- Inativação Edição de voto |