UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ

DESENVOLVIMENTO DE SOFTWARE PARA PERSISTÊNCIA

Trabalho Final - Bancos de dados não relacional - MongoDB

Discentes:

Docente:
Francisco Victor da Silva Pinheiro

QUIXADÁ - CE

MARÇO DE 2025

Repositório dos códigos: https://github.com/josiasdev/CandidatosAPI

Fontes de Dados: **https://dadosabertos.tse.jus.br/dataset/candidatos-2024**

Entidades

1. Candidato

Descrição: Representa um candidato em uma eleição.

Campos:

- `sq_candidato` (int) - Identificador único do candidato.
- `nm_candidato` (str) - Nome do candidato.
- `dt_nascimento` (datetime) - Data de nascimento do candidato.
- `ds_genero` (str) - Gênero do candidato.
- `ds_grau_instrucao` (str) - Grau de instrução.
- `ds_cor_raca` (str) - Cor/Raça do candidato.
- `ds_ocupacao` (str) - Ocupação do candidato.
- `nr_titulo_eleitoral_candidato` (int) - Número do título eleitoral.

2. Candidatura

Descrição: Representa a candidatura de um candidato a um cargo específico em uma eleição.

Campos:

- `sq_candidato` (int) - Identificador único do candidato.
- `nm_candidato` (str) - Nome do candidato.
- `cd_eleicao` (int) - Código da eleição.
- `sg_uf` (str) - Unidade federativa.
- `ds_cargo` (str) - Cargo ao qual concorre.
- `nr_candidato` (int) - Número do candidato.
- `nr_partido` (int) - Número do partido.
- `sg_partido` (str) - Sigla do partido.

- nm_partido (str) - Nome do partido.
- nr_turno (int) - Número do turno.
- tp_agremiacao (str) - Tipo de agremiação.
- ds_sit_tot_turno (str) - Situação final no turno.
- ds_tp_motivo (str) - Tipo de motivo de cassação (se houver).
- ds_motivo (str) - Motivo da cassação.

3. Bens do Candidato

Descrição: Representa os bens declarados por um candidato.

Campos:

- id (str) - Identificador único do bem.
- nr_titulo_eleitoral_candidato (str) - Número do título eleitoral do candidato.
- sq_candidato (str) - Chave estrangeira para Candidato.
- nr_ordem_bem_candidato (int) - Ordem do bem na declaração.
- ds_tipo_bem_candidato (str) - Tipo do bem.
- ds_bem_candidato (str) - Descrição do bem.
- vr_bem_candidato (float) - Valor do bem.
- dt_ult_atual_bem_candidato (date) - Data da última atualização.
- hh_ult_atual_bem_candidato (time) - Hora da última atualização.

4. Eleição

Descrição: Representa uma eleição específica.

Campos:

- ds_eleicao (str) - Descrição da eleição.
- dt_eleicao (datetime) - Data da eleição.
- ano_eleicao (int) - Ano da eleição.
- cd_tipo_eleicao (int) - Código do tipo de eleição.
- nm_tipo_eleicao (str) - Nome do tipo de eleição.
- tp_abrangencia (str) - Tipo de abrangência da eleição.
- nr_turno (int) - Número do turno.
- cd_eleicao (int) - Código único da eleição.

5. Informações do Candidato

Descrição: Representa informações adicionais sobre um candidato.

Campos:

- `nr_titulo_eleitoral_candidato` (int) - Chave estrangeira para Candidato.
- `ds_nacionalidade` (str) - Nacionalidade do candidato.
- `nm_municipio_nascimento` (str) - Município de nascimento.
- `st_quilombola` (bool) - Se pertence a comunidade quilombola.
- `vr_despesa_max_campanha` (float) - Valor máximo de despesa de campanha.
- `st_reeleicao` (bool) - Indica se está concorrendo à reeleição.
- `st_declarar_bens` (bool) - Indica se declarou bens.
- `st_prest_contas` (bool) - Indica se prestou contas.

Relações Entre Entidades

1. Candidato 1:N Candidaturas - Um candidato pode ter várias candidaturas ao longo do tempo.
2. Candidato 1:N Bens - Um candidato pode ter vários bens declarados.
3. Candidatura N:1 Eleição - Cada candidatura pertence a uma única eleição.
4. Candidato 1:1 InfoCandidato - Cada candidato possui um conjunto único de informações adicionais.
5. Eleição 1:N Candidaturas - Uma eleição pode ter várias candidaturas.

Configuração do Banco de Dados

Banco não relacional utilizado: MongoDB v8.0.4

3. Criar uma API REST com FastAPI

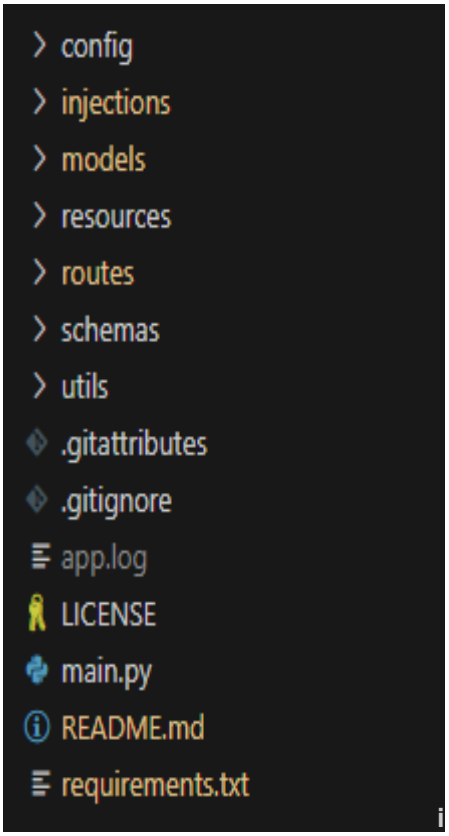
Usando FastAPI, vocês vão criar endpoints para implementar cada funcionalidade solicitada. Cada funcionalidade será implementada em um endpoint específico. Abaixo estão os detalhes de cada funcionalidade que a API deverá oferecer.

3.1 Funcionalidades da API - Candidatura

Funcionalidade	**Responsável**	**Status**
Implementar operações de CRUD		Concluído

Consultas específicas		Concluído
Agregações e cálculos estatísticos simples		Concluído
Paginação e Filtros		Concluído
Logging e Tratamento de Erros		Concluído
Análises e Visualizações		Concluído

Arquitetura do Sistema



main: Este é o ponto de entrada principal da aplicação. Ele orquestra a execução dos diferentes módulos, inicializando a API, configurando as rotas e garantindo que todos os componentes estejam funcionando em conjunto. Aqui, é onde a aplicação é iniciada, e onde as dependências são injetadas, e onde o servidor é iniciado.

injections: Este módulo é responsável pela importação e inserção de dados em massa no banco de dados MongoDB. Ele utiliza os arquivos CSV presentes em "resources" para popular as coleções do banco, automatizando o processo de carga inicial ou atualização de dados.

resources: Aqui, encontram-se arquivos compactados (ZIP) que contêm arquivos CSV. Estes arquivos CSV armazenam os dados brutos, organizados em formato tabular, que serão utilizados pelo módulo "injections" para alimentar o banco de dados MongoDB.

model: Este diretório define a camada de modelo da aplicação. As classes aqui presentes representam as entidades do banco de dados, especificando seus atributos, tipos de dados e validações. Essa camada serve como uma representação orientada a objetos dos dados, facilitando a manipulação e o gerenciamento das informações.

schemas: Este módulo concentra as definições de schemas, que são responsáveis por transformar os dados recuperados do banco de dados MongoDB em formatos específicos para a API. As funções e definições aqui presentes garantem que os dados sejam apresentados de maneira consistente e adequada para consumo externo.

rotas: Este diretório contém a definição das rotas da API REST. Os endpoints aqui definidos utilizam os modelos e schemas para implementar a lógica de negócio, permitindo a interação com os dados por meio de requisições HTTP.

utils:Este módulo agrupa funções utilitárias, como a validação de IDs do MongoDB. Essas funções fornecem funcionalidades de suporte para outras partes da aplicação, promovendo a reutilização de código e a padronização de processos.

app.log: Este arquivo registra os logs da aplicação, fornecendo informações detalhadas sobre o funcionamento do sistema, incluindo erros, avisos e eventos importantes. Esses logs são essenciais para monitorar a saúde da aplicação e diagnosticar problemas.

Dependencias De Instalação

1. annotated-types==0.7.0
2. anyio==4.8.0
3. asttokens==3.0.0
4. attrs==25.1.0
5. backcall==0.2.0
6. beautifulsoup4==4.13.3
7. bleach==6.2.0
8. bson==0.5.10
9. certifi==2025.1.31
10. charset-normalizer==3.4.1
11. click==8.1.8
12. colorama==0.4.6
13. decorator==5.2.1
14. defusedxml==0.7.1
15. dnspython==2.7.0
16. docopt==0.6.2
17. executing==2.2.0
18. fastapi==0.115.8
19. fastjsonschema==2.21.1
20. h11==0.14.0
21. idna==3.10
22. ipython==8.12.3
23. jedi==0.19.2
24. Jinja2==3.1.6
25. jsonschema==4.23.0
26. jsonschema-specifications==2024.10.1
27. jupyter_client==8.6.3
28. jupyter_core==5.7.2
29. jupyterlab_pygments==0.3.0
30. MarkupSafe==3.0.2
31. matplotlib-inline==0.1.7
32. mistune==3.1.2
33. motor==3.7.0
34. nbclient==0.10.2
35. nbconvert==7.16.6
36. nbformat==5.10.4
37. numpy==2.2.3
38. packaging==24.2
39. pandas==2.2.3
40. pandocfilters==1.5.1

41. parso==0.8.4
42. pickleshare==0.7.5
43. pipreqs==0.5.0
44. platformdirs==4.3.6
45. prompt_toolkit==3.0.50
46. pure_eval==0.2.3
47. pydantic==2.10.6
48. pydantic_core==2.27.2
49. Pygments==2.19.1
50. pymongo==4.11
51. python-dateutil==2.9.0.post0
52. python-dotenv==1.0.1
53. python-multipart==0.0.20
54. pytz==2025.1
55. pywin32==308
56. pyzmq==26.2.1
57. referencing==0.36.2
58. requests==2.32.3
59. rpds-py==0.23.1
60. six==1.17.0
61. sniffio==1.3.1
62. soupsieve==2.6
63. stack-data==0.6.3
64. starlette==0.45.3
65. tinycss2==1.4.0
66. tornado==6.4.2
67. traitlets==5.14.3
68. typing_extensions==4.12.2
69. tzdata==2025.1
70. urllib3==2.3.0
71. uvicorn==0.34.0
72. wcwidth==0.2.13
73. webencodings==0.5.1
74. yarg==0.1.9