

Projeto de Bancos de Dados CandiDados

Ana Sofia Schweizer Silvestre, 200014382
Luiz Carlos da Silva Néto Vartuli, 200023314
Victor Hugo França Lisboa, 200028341

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)
CIC0097 - Bancos de Dados

1. Introdução

O projeto CandiDados tem como intuito implementar uma plataforma de consulta de informações de políticos do poder legislativo. Nela são apresentados todos os dados públicos de cada político que atua no Poder Legislativo, como: nome e outros dados pessoais, partido, benefícios, processos judiciais que respondem, órgãos que atuam e projetos de lei que escreveram e votaram. Para cada político, também, é possível checar o seu histórico de partidos, mandatos e órgãos que atuaram.

O sistema foi implementado utilizando o SGBD MySQL, *Python* para implementação da camada de persistência e negócio e *html*, *css* e *JavaScript* para implementação da interface com o usuário.

O repositório do Github deste projeto está disponível neste link.

2. Modelo de Entidade Relacionamento

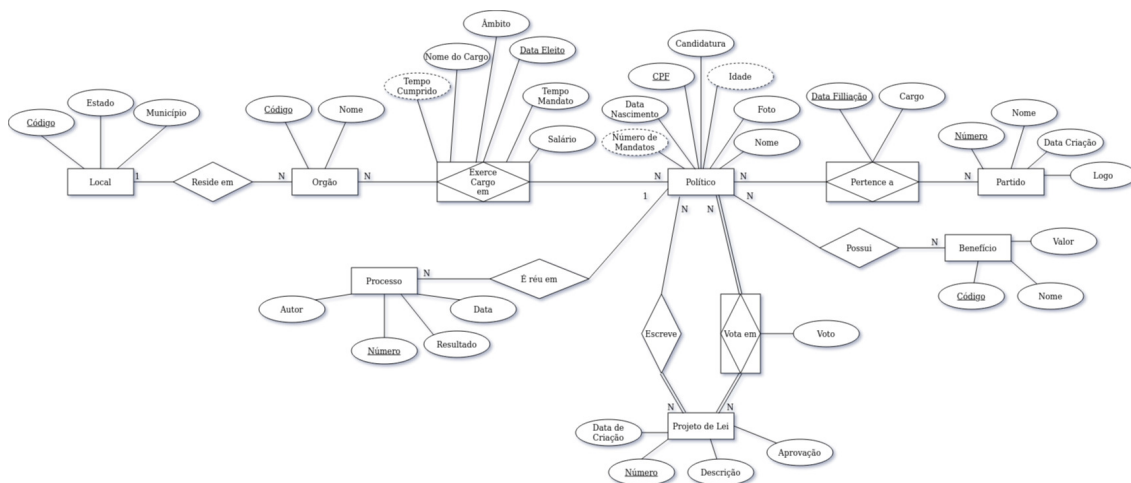


Figura 1. Modelo Entidade Relacionamento do Sistema CandiDados.

3. Modelo Relacional

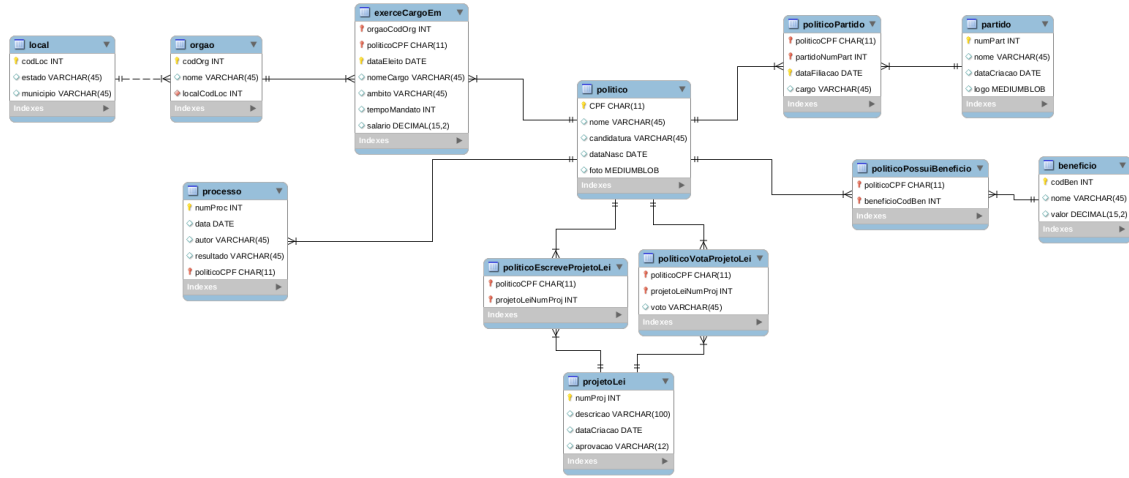


Figura 2. Modelo Relacional do Sistema Candidatos.

4. Álgebra Relacional

Consultas em álgebra relacional das tabelas do banco.

1. Projeção do nome dos políticos, nome do cargo que trabalham e nome do órgão em que trabalham.

$$RELACAO \leftarrow (politico \bowtie_{(CPF=politicoCPF)} exerceCargoEm)$$

$$RELACAO \leftarrow (RELACAO \bowtie_{(codOrg=orgaoCodOrg)} orgao)$$

$$\pi_{\rho_{nome}(politico.nome), nomeCargo, orgao.nome}(RELACAO)$$

2. Projeção do nome dos políticos, do âmbito de seu cargo tal como cidade e estado em que trabalham.

$$RELACAO \leftarrow (politico \bowtie_{(CPF=politicoCPF)} exerceCargoEm)$$

$$RELACAO \leftarrow (RELACAO \bowtie_{(codOrg=orgaoCodOrg)} orgao)$$

$$RELACAO \leftarrow (RELACAO \bowtie_{(codLoc=localCodLoc)} local)$$

$$\pi_{politico.nome, ambito, municipio, estado}(RELACAO)$$

3. Projeção do nome dos políticos, a descrição do projeto de lei em que votaram e o seu voto em si.

$$RELACAO \leftarrow (politico \bowtie_{(CPF=politicoCPF)} politicoVotaProjetoLei)$$

$$RELACAO \leftarrow (RELACAO \bowtie_{(numProj=projetoLeiNumProj)} projetoLei)$$

$$\pi_{nome, descricao, voto}(RELACAO)$$

4. Projeção do nome dos políticos, nome dos benefícios e os valores desses benefícios. Só são projetados os registros dos benefícios com valor superior a 5000.

$$RELACAO \leftarrow (politico \bowtie_{(CPF=politicoCPF)} politicoPossuiBeneficio)$$

$$RELACAO \leftarrow (RELACAO \bowtie_{(codBen=beneficioCodBen)} beneficio)$$

$$SELECAO \leftarrow \sigma_{(valor>5000)}(RELACAO)$$

$$\pi_{politico.nome, beneficio.nome, valor}(SELECAO)$$

5. Projeção do nome dos políticos, partidos em que estão filiados e suas respectivas datas de filiação. Apenas são projetados os políticos que se filiaram a partidos a partir do ano de 2005.

$$\begin{aligned} RELACAO &\leftarrow (politico \bowtie_{(CPF=politicoCPF)} politicoPartido) \\ RELACAO &\leftarrow (RELACAO \bowtie_{(numPart=partidoNumPart)} partido) \\ SELECAO &\leftarrow \sigma_{(dataFiliacao > "2005-01-01")}(RELACAO) \\ \pi_{politico.nome, partido.nome, dataFiliacao} SELECAO \end{aligned}$$

5. Formas Normais

```
CREATE TABLE politico(  
    CPF CHAR(11) PRIMARY KEY,  
    nome VARCHAR(45) NOT NULL,  
    candidatura VARCHAR(45),  
    dataNasc DATE NOT NULL,  
    foto MEDIUMBLOB  
);
```

Figura 3. Tabela politico

1FN: A tabela politico está de acordo com a 1ª forma normal (1FN) porque cada célula possui apenas um valor. O SGBD MySQL não permite que seja inserido mais de um valor para uma coluna de um registro.

2FN: A tabela politico está de acordo com a 2ª forma normal (2FN) porque está na 1FN e nenhum atributo é funcionalmente dependente de nenhum outro atributo além do CPF, que é a chave da tabela.

3FN: A tabela politico está de acordo com a 3ª forma normal (3FN) pois está na 2FN e todos os atributos são determinados única e exclusivamente pelo CPF do político.

```
CREATE TABLE beneficio(  
    codBen INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(45) NOT NULL,  
    valor DECIMAL(15,2) NOT NULL  
);
```

Figura 4. Tabela Benefício

1FN: A tabela beneficio está na 1FN porque cada célula possui apenas um valor.

2FN: A tabela beneficio está na 2FN porque todos os atributos complemento da chave são totalmente funcionalmente dependentes da chave codBen.

3FN: A tabela beneficio não está na 3FN pois, apesar de estar na 2FN, tem o atributo valor transitivamente dependente do atributo nome.

```
CREATE TABLE partido(  
    numPart INT PRIMARY KEY,  
    nome VARCHAR(45) NOT NULL,  
    dataCriacao DATE NOT NULL,  
    logo MEDIUMBLOB  
);
```

Figura 5. Tabela Partido

1FN: A tabela partido está na 1FN porque cada célula possui apenas um valor.

2FN: A tabela partido está na 2FN porque nenhum atributo é funcionalmente dependente de nenhum outro atributo além da chave numPart.

3FN: A tabela partido está na 3FN pois está na 2FN e os atributos nome, dataCriacao e logo dependem exclusivamente da chave numPart.

```
CREATE TABLE processo(  
    numProc INT PRIMARY KEY,  
    dataInicio DATE NOT NULL,  
    autor VARCHAR(45) NOT NULL,  
    resultado ENUM('CULPADO', 'INOCENTADO'),  
    politicoCPF CHAR(11),  
    FOREIGN KEY (politicoCPF) REFERENCES politico(CPF) ON DELETE CASCADE  
);
```

Figura 6. Tabela Processo

1FN: A tabela processo está na 1FN porque cada célula possui apenas um valor.

2FN: A tabela processo está na 2FN porque está na 1FN e nenhum atributo é funcionalmente dependente de nenhum outro atributo além da chave numProc.

3FN: A tabela processo está na 3FN pois está na 2FN e todos os outros atributos dependem somente da chave numProc.

```
CREATE TABLE politicoPartido(  
    politicoCPF CHAR(11),  
    partidoNumPart INT,  
    dataFiliacao DATE,  
    cargo VARCHAR(45),  
    FOREIGN KEY (politicoCPF) REFERENCES politico(CPF) ON DELETE CASCADE,  
    FOREIGN KEY (partidoNumPart) REFERENCES partido(numPart) ON DELETE CASCADE,  
    PRIMARY KEY (politicoCPF, partidoNumPart, dataFiliacao)  
);
```

Figura 7. Tabela politicoPartido

1FN: A tabela politicoPartido está na 1FN porque cada célula possui apenas um valor.

2FN: A tabela politicoPartido está na 2FN porque está na 1FN e todos os atributos são totalmente funcionalmente dependentes da chave (politicoCPF, partidoNumPart, dataFiliacao).

3FN: A tabela politicoPartido está na 3FN pois está na 2FN e o atributo cargo, o único complemento, não é determinado por nenhum atributo além da chave.

6. Camada de Mapeamento

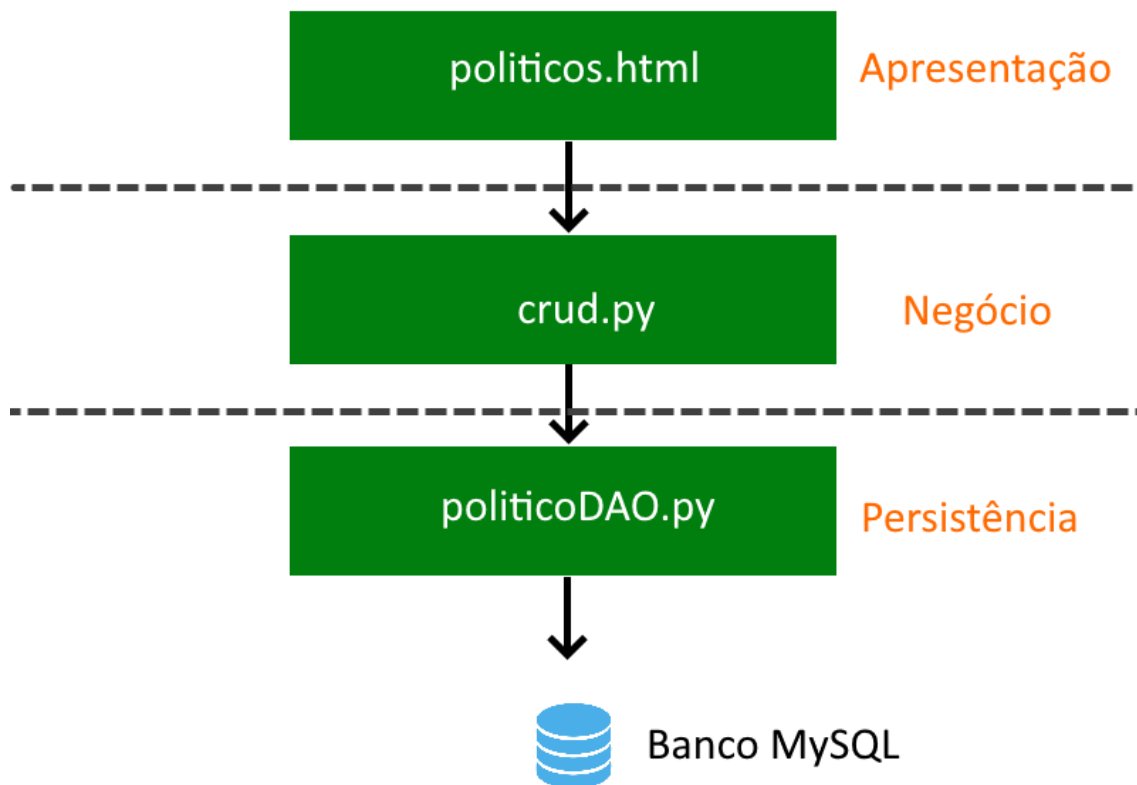


Figura 8. Diagrama da Camada de Mapeamento do Sistema Candidados.