

# Análise de Gastos dos Senadores do Brasil

Thiago de O. Silva<sup>1</sup>, Arthur M. de Albuquerque<sup>1</sup>, João R. C. Maia<sup>1</sup>, Miguel S. de A. do Nascimento<sup>1</sup>, François A. Boéchat<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação

Universidade Federal do Rio de Janeiro (UFRJ) – Rio de Janeiro, RJ – Brazil

{thiagoprg91, arthur.albuquerque.1995, miguelsantosufrj}@gmail.com,  
joaomaia@ufrj.br, fran\_5298@Hotmail.com

**Abstract.** *This article describes the steps of designing and implementing a relational database based on factual data analysis. Queries will be presented using open data available on the Federal Senate website regarding the expenses of Senators in Brazil. The work also has a web graphical interface for better visualization and understanding of the results presented.*

**Resumo.** *Este artigo descreve as etapas de projeção e implementação de um banco de dados relacional com base em análise de dados factuais. Serão apresentadas consultas utilizando dados abertos disponibilizados no site do Senado Federal referentes aos gastos dos senadores do Brasil. O trabalho conta ainda com uma interface gráfica web para melhor visualização e entendimento dos resultados apresentados.*

## 1. Introdução

No presente relatório, encontram-se a descrição do projeto, os modelos ER e físico, bem como as consultas realizadas. Em seguida temos o desenvolvimento da aplicação web onde serão realizadas as consultas em SQL no banco de dados. Os dados utilizados se referem aos gastos dos senadores brasileiros que estão disponíveis publicamente.

## 2. Modelagem do banco de dados

A seguir, os modelos conceitual, lógico e físico criados e utilizados na implementação do banco de dados.

### 2.1 Modelo conceitual

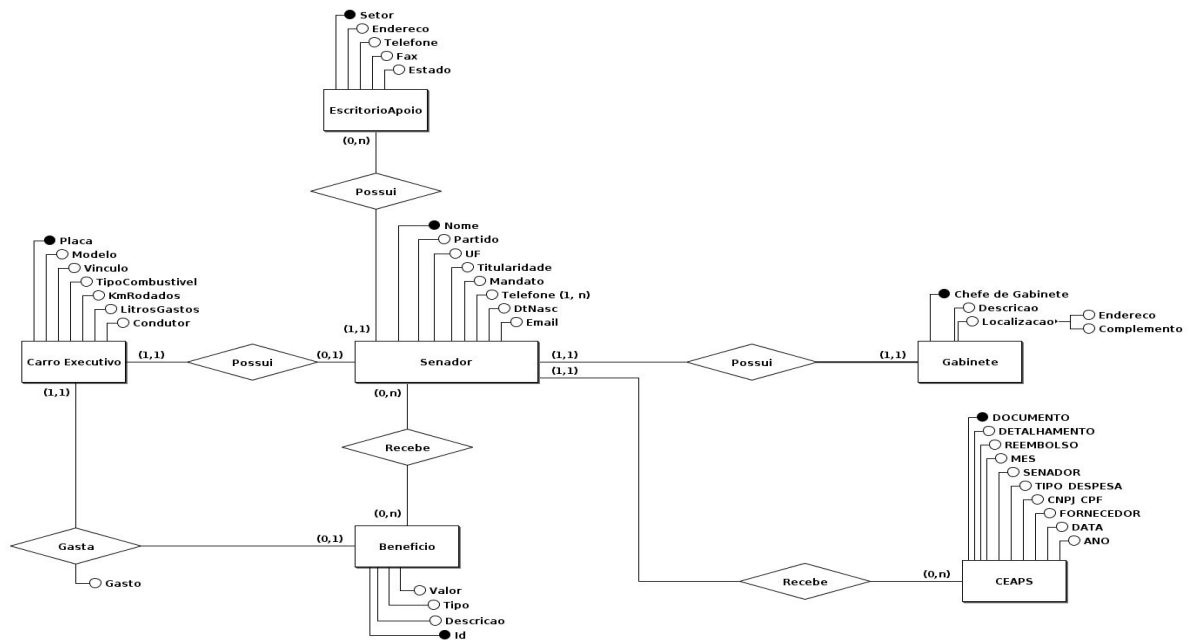


Figura 1. Modelo conceitual

## 2.2 Modelo lógico

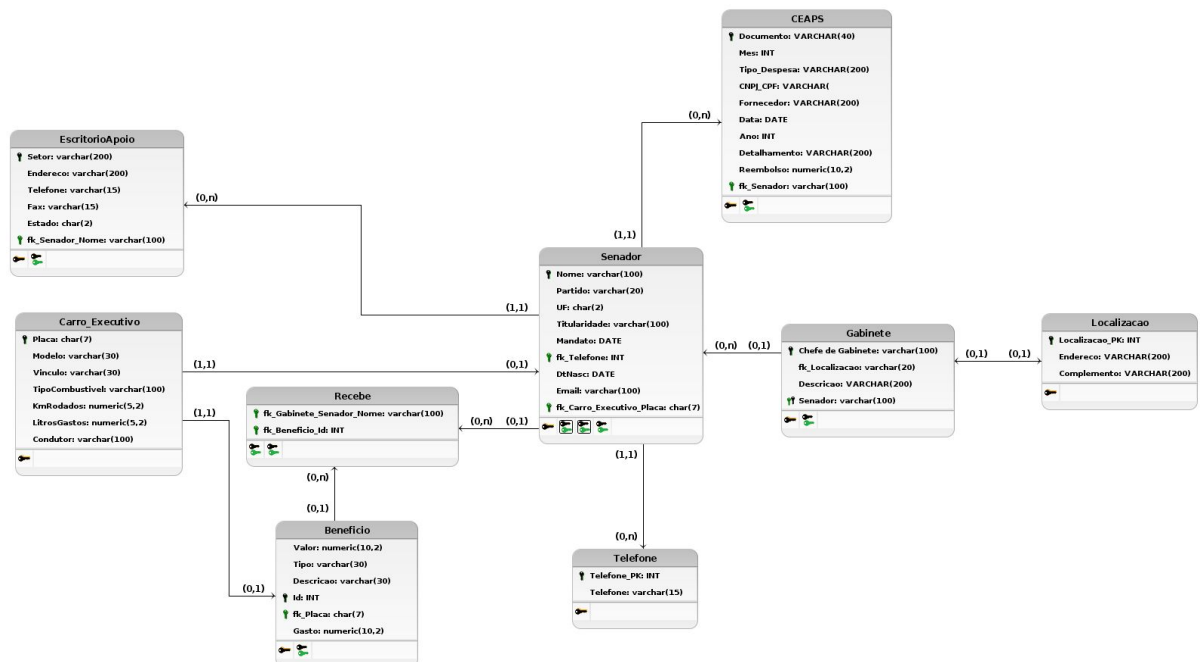


Figura 2. Modelo Lógico

## 2.3 Modelo físico

```
CREATE TABLE Senador (
  ID INT AUTO INCREMENT PRIMARY KEY,
  Nome VARCHAR(100),
  Partido VARCHAR(20),
  UF CHAR(2),
  Titularidade VARCHAR(100),
  Mandato VARCHAR(20),
  DtNasc CHAR(10),
  Email VARCHAR(100),
  fk_Placa CHAR(7)
);

CREATE TABLE Beneficio (
  Valor NUMERIC(10,2),
  Tipo VARCHAR(30),
  Descricao VARCHAR(500),
  Id INT AUTO INCREMENT PRIMARY KEY
);

CREATE TABLE Carro_Executivo (
  Placa CHAR(7) PRIMARY KEY,
  Modelo VARCHAR(30),
  Vinculo VARCHAR(10),
  TipoCombustivel VARCHAR(100),
  KmRodados INT,
  LitrosGastos NUMERIC(5,2),
  Condutor VARCHAR(100)
);

CREATE TABLE Localizacao (
  Localizacao_PK INT NOT NULL PRIMARY KEY AUTO INCREMENT,
  Endereco VARCHAR(200),
  Complemento VARCHAR(200)
);

CREATE TABLE Sen_Telefone (
  Sen INT,
  Telefone VARCHAR(15)
);

CREATE TABLE CEAPS (
  ID INT AUTO INCREMENT PRIMARY KEY,
  Documento VARCHAR(40),
  Mes INT,
  Tipo_Despesa VARCHAR(200),
  CNPJ_CPF VARCHAR(20),
  Fornecedor VARCHAR(200),
  Data CHAR(10),
  Ano INT,
  Detalhamento VARCHAR(1000),
  Reembolso NUMERIC(10,2),
  Sen_ID INT
);

CREATE TABLE Gabinete (
  Chefe_de_Gabinete VARCHAR(100),
  fk_Localizacao INT,
  Descricao VARCHAR(200),
  Sen_ID INT,
  PRIMARY KEY (Chefe_de_Gabinete, Sen_ID)
);

CREATE TABLE Recebe (
  fk_Sen_ID INT,
  fk_Beneficio_Id INT
);
```

Figura 3. Modelo Físico. Parte 1

```
ALTER TABLE Gabinete ADD CONSTRAINT FK_Gabinete_2
FOREIGN KEY (fk_Localizacao)
REFERENCES Localizacao (Localizacao_PK)
ON DELETE SET NULL;

ALTER TABLE Sen_Telefone ADD CONSTRAINT FK_Senador_3
FOREIGN KEY (Sen)
REFERENCES Senador (ID);

ALTER TABLE Senador ADD CONSTRAINT FK_Senador_4
FOREIGN KEY (fk_Placa)
REFERENCES Carro_Executivo (Placa)
ON DELETE CASCADE;

ALTER TABLE EscritorioApoio ADD CONSTRAINT FK_EscritorioApoio_2
FOREIGN KEY (fk_Sen_ID)
REFERENCES Senador (ID)
ON DELETE CASCADE;

ALTER TABLE Recebe ADD CONSTRAINT FK_Recebe_1
FOREIGN KEY (fk_Sen_ID)
REFERENCES Senador (ID)
ON DELETE SET NULL;

ALTER TABLE Recebe ADD CONSTRAINT FK_Recebe_2
FOREIGN KEY (fk_Beneficio_Id)
REFERENCES Beneficio (Id)
ON DELETE SET NULL;

ALTER TABLE Gabinete ADD CONSTRAINT FK_Gabinete_3
FOREIGN KEY (Sen_ID)
REFERENCES Senador (ID);

ALTER TABLE CEAPS ADD CONSTRAINT FK_CEAPS_2
FOREIGN KEY (Sen_ID)
REFERENCES Senador (ID);
```

Figura 4. Modelo Físico. Parte 2

### 3. Consultas

Conforme foi enunciado no trabalho, foram realizadas 10 consultas em SQL, sob as seguintes condições:

- 1 consulta envolvendo apenas as operações de seleção e projeção.
- 2 consultas envolvendo a junção de apenas duas relações, sendo que uma deve conter junção externa.
- 2 consultas envolvendo a junção de três ou mais relações.
- 1 consulta envolvendo uma das operações sobre conjuntos (união, diferença ou intersecção).
- 3 consultas envolvendo funções de agregação.
- 1 consulta envolvendo subconsultas aninhadas.

#### 3.1. Explicação textual

Para a primeira consulta, foram projetados, pela cláusula SQL SELECT, todos os atributos da relação Escritórios de Apoio, e então foram selecionados, pela cláusula SQL WHERE, aqueles que se localizam no estado do Rio de Janeiro.

Para a segunda consulta, foi consultada a quantidade de quilômetros rodados por cada senador, onde foi feita uma junção externa à esquerda entre as relações Senador e CarroExecutivo, respectivamente. Os nomes dos senadores foram ordenados pela quantidade de quilômetros rodados, de forma descendente.

Para a terceira consulta, foram selecionados os senadores que possuem apenas 1 telefone de contato, através de uma junção interna entre as relações Senador e Sen\_Telefone.

Para a quarta consulta, foram retornados os valores que cada senador recebeu de Auxílio-Combustível, fazendo-se uma junção interna entre as relações Senador, Recebe e Benefício, onde o atributo Tipo da relação Auxilio era 'Auxilio-Combustível'. Os senadores foram ordenados pelo valor recebido pelo auxílio de forma descendente.

Para a quinta consulta, foram retornados os valores totais de benefícios recebidos para cada partido, ou seja, a soma do Auxílio Combustível e do Auxílio Moradia. Assim como na consulta anterior, foi feita uma junção interna entre as relações Senador, Recebe e Benefício.

Para a sexta consulta, foi selecionado o reembolso total recebido por 3 partidos arbitrários - PT, PSL e PSDB - pelo CEAPS. Foi utilizada a operação de conjunto UNION do SQL para realizar a união entre (SELECT ID FROM Senador WHERE Partido = 'PT'), (SELECT ID FROM Senador WHERE Partido = 'PSL') e (SELECT ID FROM Senador WHERE Partido = 'PSDB').

Para a sétima consulta, foi retornado o reembolso total do CEAPS que cada fornecedor recebeu, com o uso da função de agregação SUM sobre o atributo Reembolso da relação CEAPS.

Para a oitava consulta, foi retornado o valor total do reembolso do CEAPS recebido por cada senador, com seus nomes ordenados pelo valor do reembolso, de maneira descendente. Assim como na consulta anterior, foi utilizado SUM(Reembolso)

Para a nona consulta, foi retornado a quantidade de reembolsos do CEAPS cada senador recebeu, bem como o valor médio dos reembolsos, através das funções de agregação COUNT e AVG, usadas no atributo Reembolso da relação CEAPS.

Para a décima consulta, foram selecionados os senadores que abdicaram do Auxílio-Moradia, através da cláusula SQL NOT IN seguida da subconsulta (SELECT fk\_Sen\_ID FROM Recebe WHERE fk\_Beneficio\_Id = 77 OR fk\_Beneficio\_Id = 78) aninhada dentro da consulta principal, onde foi retornado o nome, partido e a Unidade Federal do senador (SELECT Nome, Partido, UF FROM Senador WHERE Senador.ID).

### 3.2. Comandos SQL

- 1 consulta envolvendo apenas as operações de seleção e projeção:  

```
SELECT * FROM EscritorioApoio  
WHERE Estado = 'RJ';
```
- 2 consultas envolvendo a junção de apenas duas relações, sendo que uma deve conter junção externa:  

```
SELECT Senador.Nome, KmRodados  
FROM Senador  
LEFT JOIN Carro_Executivo ON Senador.fk_Placa = Carro_Executivo.Placa  
ORDER BY KmRodados DESC;
```

```
SELECT Senador.Nome, COUNT(Sen_Telefone.Sen)  
FROM Senador INNER JOIN Sen_Telefone ON Senador.ID =  
Sen_Telefone.Sen  
GROUP BY Senador.Nome HAVING COUNT(Sen_Telefone.Sen) < 2;
```
- 2 consultas envolvendo a junção de três ou mais relações:  

```
SELECT Nome, Partido, UF, Valor  
FROM Senador  
INNER JOIN Recebe ON Recebe.fk_Sen_ID = Senador.ID  
INNER JOIN Beneficio ON Recebe.fk_Beneficio_Id = Beneficio.ID  
WHERE Tipo = "Auxilio-Combustivel" ORDER BY Valor DESC;
```

```

SELECT Partido, SUM(Valor)
FROM Senador
INNER JOIN Recebe ON Recebe.fk_Sen_ID = Senador.ID
INNER JOIN Beneficio ON Recebe.fk_Beneficio_Id = Beneficio.ID
GROUP BY Partido;

```

- 1 consulta envolvendo uma das operações sobre conjuntos (união, diferença ou intersecção):

```

SELECT Partido, SUM(Reembolso)
FROM CEAPS
INNER JOIN Senador ON Senador.ID = CEAPS.Sen_ID
WHERE Senador.ID IN
    ( (SELECT ID FROM Senador WHERE Partido = 'PT')
      UNION
      (SELECT ID FROM Senador WHERE Partido = 'PSL')
      UNION
      (SELECT ID FROM Senador WHERE Partido = 'PSDB') )
GROUP BY Partido;

```

- 3 consultas envolvendo funções de agregação:

```

SELECT Fornecedor, SUM(Reembolso)
FROM CEAPS
INNER JOIN Senador ON Sen_ID=Senador.ID
GROUP BY Fornecedor;

```

```

SELECT Senador.Nome, SUM(Reembolso)
FROM CEAPS INNER JOIN Senador ON Senador.ID = Sen_ID
GROUP BY Sen_ID ORDER BY SUM(Reembolso) DESC;

```

```

SELECT COUNT(Reembolso), Nome
FROM CEAPS
INNER JOIN Senador ON Sen_ID = Senador.ID
GROUP BY Sen_ID ORDER BY COUNT(Reembolso) DESC;

```

- 1 consulta envolvendo subconsultas aninhadas:

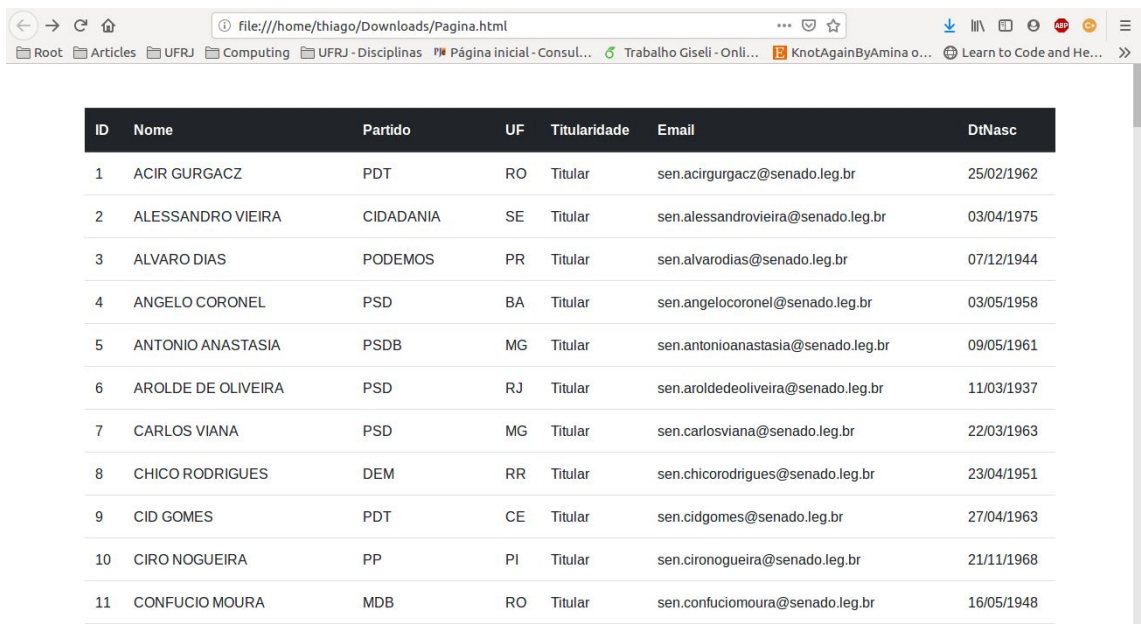
```

SELECT Nome, Partido, UF
FROM Senador
WHERE Senador.ID NOT IN (
    SELECT fk_Sen_ID
    FROM Recebe
    WHERE fk_Beneficio_Id = 77 OR fk_Beneficio_Id = 78);

```

#### 4. Aplicação Web

Para a aplicação web foi utilizada a ferramenta NodeJs



A screenshot of a web browser displaying a table of senators. The browser's address bar shows the file path 'file:///home/thiago/Downloads/Pagina.html'. The table has seven columns: ID, Nome, Partido, UF, Titularidade, Email, and DtNasc. It contains 11 rows of data, each representing a senator with their respective details.

ID	Nome	Partido	UF	Titularidade	Email	DtNasc
1	ACIR GURGACZ	PDT	RO	Titular	sen.acirgurgacz@senado.leg.br	25/02/1962
2	ALESSANDRO VIEIRA	CIDADANIA	SE	Titular	sen.alessandrovieira@senado.leg.br	03/04/1975
3	ALVARO DIAS	PODEMOS	PR	Titular	sen.alvarodias@senado.leg.br	07/12/1944
4	ANGELO CORONEL	PSD	BA	Titular	sen.angelocoronel@senado.leg.br	03/05/1958
5	ANTONIO ANASTASIA	PSDB	MG	Titular	sen.antonioanastasia@senado.leg.br	09/05/1961
6	AROLDE DE OLIVEIRA	PSD	RJ	Titular	sen.aroldeoliveira@senado.leg.br	11/03/1937
7	CARLOS VIANA	PSD	MG	Titular	sen.carlosviana@senado.leg.br	22/03/1963
8	CHICO RODRIGUES	DEM	RR	Titular	sen.chicorodrigues@senado.leg.br	23/04/1951
9	CID GOMES	PDT	CE	Titular	sen.cidgomes@senado.leg.br	27/04/1963
10	CIRO NOGUEIRA	PP	PI	Titular	sen.cironogueira@senado.leg.br	21/11/1968
11	CONFUCIO MOURA	MDB	RO	Titular	sen.confuciomoura@senado.leg.br	16/05/1948

Figura 5. Página html da aplicação web

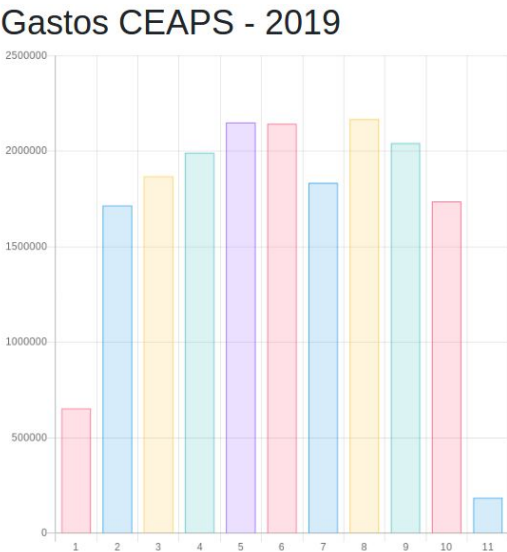


Figura 6. Exemplo de gráfico gerado na aplicação web

## **5. Distribuição de tarefas**

Arthur: Relatório, Modelagem ER, Apresentação

François: Modelagem ER, Consultas SQL, modificação dos datasets, Criação dos Scripts de Insert

Miguel: Relatório, Modelagem ER, Consultas SQL, Modelo físico, Apresentação

Thiago: Relatório, Modelagem ER, Apresentação, Aplicação web

João: Seleção de Datasets, Modelagem ER, Modelo Lógico, Modelo Físico, Modificação dos Datasets para estar de acordo com o banco, Criação dos Scripts de Insert, Consultas SQL, Aplicação web

## **6. Referências**

SILBERSCHATZ, A; KORTH, H.F.; SUDARSHAN S. Database Systems Concepts. 6a. ed. McGraw Hill-Companies, 2010.

HEUSER, C.A. Projeto de Banco de Dados. 6a ed. Bookman, Porto Alegre, 2009.

ELMASRI, R.; NAVATHE, S.B. Sistemas de Banco de Dados. 6a. ed. Pearson Education, 2011.