

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEC – ENGENHARIA DE COMPUTAÇÃO
BANCO DE DADOS DEC7129

Karen Norberto

Smart Cook
Prof. Alexandre Leopoldo Gonçalves

2025
Araranguá

SUMÁRIO

1. DESCRIÇÃO DO OBJETIVO GERAL	03
2. DESCRIÇÃO DETALHADA	03
3. MODELO CONCEITUAL	04
4. MODELO LÓGICO	04
5. DDL	04
6. CONSULTAS	07
7. APLICAÇÃO	13

1. Descrição do objetivo geral do sistema:

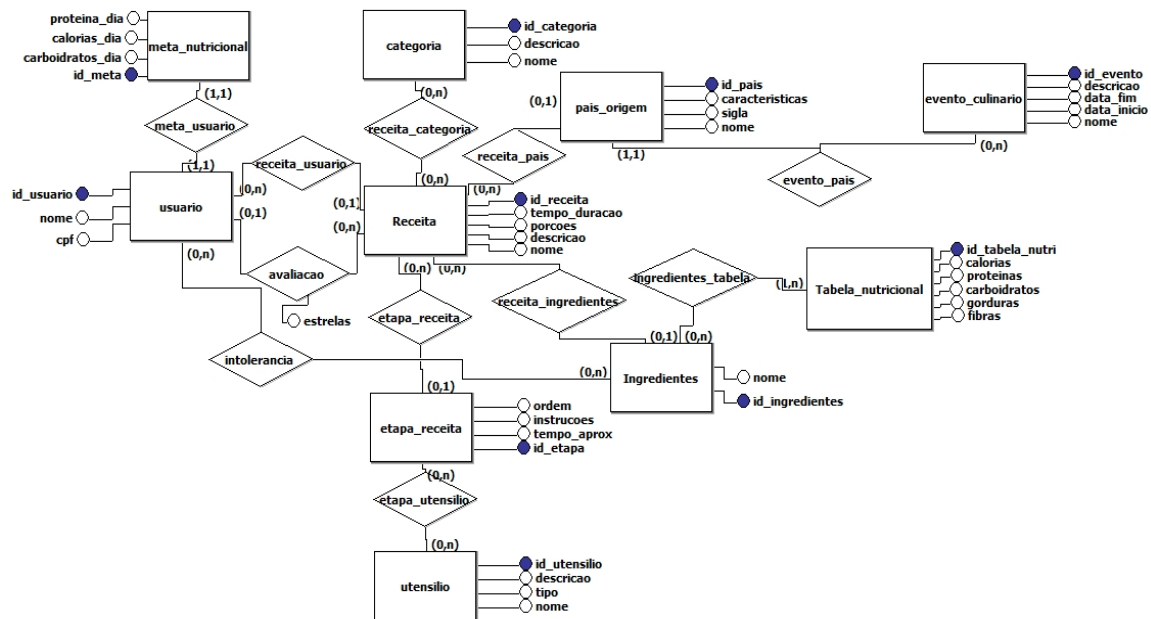
Com a finalidade de auxiliar tanto nutricionalmente quanto para criação de ideias de sabor, foi realizado um projeto cujo objetivo central é obter um assistente nutricional inteligente. O software deve auxiliar os usuários a encontrarem receitas saudáveis e/ou gostosas com base nos ingredientes disponíveis em casa, elencando as calorias e as restrições para cada receita.

2. Descrição detalhada:

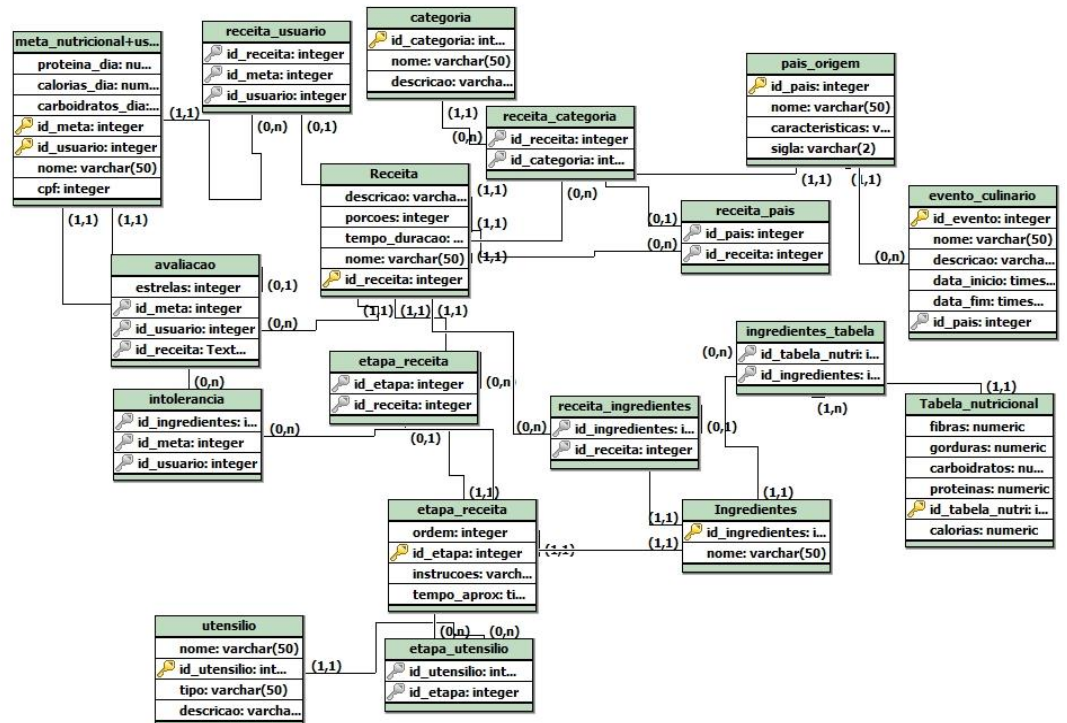
A fim de prover as informações necessárias para o sistema, os seguintes requisitos devem ser levados em consideração:

- 1- O sistema deve permitir o registro de receitas culinárias. Cada registro deve conter um identificador, nome, descrição, tempo de preparo e as porções.
- 2- O sistema deve permitir o registro de ingredientes. Cada registro deve conter um identificador e o nome.
- 3- O sistema deve permitir o registro de categoria, onde deve conter um identificador, nome e descrição.
- 4- O sistema deve permitir o registro do país de origem da receita, contendo um identificador, características do país, nome e a sigla.
- 5- O sistema deve permitir o cadastro de eventos culinários, contendo um identificador, descrição, nome, data de fim, data de início e o nome.
- 6- O sistema deve permitir o cadastro da tabela nutricional das receitas. O cadastro deve conter um identificador, calorias, proteínas, carboidratos, gorduras e fibras.
- 7- O sistema deve permitir o cadastro das etapas das receitas, contendo um identificador, tempo aproximado, a ordem e instruções.
- 8- O sistema deve permitir o cadastro dos utensílios necessários para as receitas. O cadastro deve conter um identificador, descrição, tipo e o nome.
- 9- O sistema deve permitir o registro do usuário. O registro deve conter um identificador, nome e o CPF.
- 10- O sistema deve permitir o registro de metas nutricionais do usuário. O registro deve conter um identificador e as metas de calorias, proteínas e carboidratos do dia.

3. Modelo conceitual:



4. Modelo lógico:



4. DDL:

```
CREATE TABLE RECEITA (  
    id_receita INTEGER PRIMARY KEY,  
    nome VARCHAR(50) NOT NULL,  
    descricao VARCHAR(50) NOT NULL,  
    porcoes INTEGER,  
    tempo_duracao VARCHAR(20)  
);  
""",  
'CATEGORIA': """"  
    CREATE TABLE CATEGORIA (  
        id_categoria INTEGER PRIMARY KEY,  
        nome VARCHAR(50) NOT NULL,  
        descricao VARCHAR(50)  
    );  
""",  
'USUARIO': """"  
    CREATE TABLE USUARIO (  
        id_usuario INTEGER,  
        nome VARCHAR(50) NOT NULL,  
        cpf INTEGER NOT NULL,  
        PRIMARY KEY (id_usuario)  
    );  
""",  
'META_NUTRICIONAL': """"  
    CREATE TABLE META_NUTRICIONAL (  
        id_meta INTEGER,  
        proteina_dia NUMERIC,  
        calorias_dia NUMERIC,  
        carboidratos_dia NUMERIC,  
        id_usuario INTEGER,  
        PRIMARY KEY (id_meta),  
        FOREIGN KEY (id_usuario) REFERENCES USUARIO (id_usuario)  
    );  
""",  
'PAIS_ORIGEM': """"  
    CREATE TABLE PAIS_ORIGEM (  
        id_pais INTEGER PRIMARY KEY,
```

```

    nome VARCHAR(50) NOT NULL,
    características VARCHAR(50),
    sigla VARCHAR(3)
);
""",
'EVENTO_CULINARIO': ""
CREATE TABLE EVENTO_CULINARIO (
    id_evento INTEGER PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    descricao VARCHAR(50),
    data_inicio TIMESTAMP,
    data_fim TIMESTAMP,
    id_pais INTEGER,
    FOREIGN KEY (id_pais) REFERENCES PAIS_ORIGEM (id_pais)
);
""",
'UTENSILIO': ""
CREATE TABLE UTENSILIO (
    id_utensilio INTEGER PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    tipo VARCHAR(50),
    descricao VARCHAR(50)
);
""",
'TABELA_NUTRICIONAL': ""
CREATE TABLE TABELA_NUTRICIONAL (
    id_tabela_nutri INTEGER PRIMARY KEY,
    fibras NUMERIC,
    gorduras NUMERIC,
    carboidratos NUMERIC,
    proteinas NUMERIC,
    calorias NUMERIC,
    id_receita INTEGER,
    FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita)
);
""",
'INGREDIENTES': ""
CREATE TABLE INGREDIENTES (
    id_ingredientes INTEGER PRIMARY KEY,

```

```

    nome VARCHAR(50) NOT NULL,
    id_receita INTEGER,
    FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita)
);
""",
'ETAPA_RECEITA': """"
CREATE TABLE ETAPA_RECEITA (
    id_etapa INTEGER PRIMARY KEY,
    ordem INTEGER,
    instrucoes VARCHAR(100),
    tempo_aprox INTERVAL,
    id_receita INTEGER,
    FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita)
);
""",
'RECEITA_PAIS': """"
CREATE TABLE RECEITA_PAIS (
    id_pais INTEGER,
    id_receita INTEGER,
    FOREIGN KEY (id_pais) REFERENCES PAIS_ORIGEM (id_pais),
    FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita)
);
""",
'ETAPA_DA_RECEITA': """"
CREATE TABLE ETAPA_DA_RECEITA (
    id_etapa INTEGER,
    id_receita INTEGER,
    FOREIGN KEY (id_etapa) REFERENCES ETAPA_RECEITA (id_etapa),
    FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita)
);
""",
'ETAPA_UTENSILIO': """"
CREATE TABLE ETAPA_UTENSILIO (
    id_utensilio INTEGER,
    id_etapa INTEGER,
    FOREIGN KEY (id_utensilio) REFERENCES UTENSILIO (id_utensilio),
    FOREIGN KEY (id_etapa) REFERENCES ETAPA_RECEITA (id_etapa)
);
""",

```

```

'RECEITA_USUARIO': ""
    CREATE TABLE RECEITA_USUARIO (
        id_receita INTEGER,
        id_usuario INTEGER,
        FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita),
        FOREIGN KEY (id_usuario) REFERENCES USUARIO (id_usuario)
    );
"",
'RECEITA_CATEGORIA': ""
    CREATE TABLE RECEITA_CATEGORIA (
        id_receita INTEGER,
        id_categoria INTEGER,
        FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita),
        FOREIGN KEY (id_categoria) REFERENCES CATEGORIA (id_categoria)
    );
"",
'INGREDIENTES_TABELA': ""
    CREATE TABLE INGREDIENTES_TABELA (
        id_tabela_nutri INTEGER,
        id_ingredientes INTEGER,
        FOREIGN KEY (id_tabela_nutri) REFERENCES TABELA_NUTRICIONAL
(id_tabela_nutri),
        FOREIGN KEY (id_ingredientes) REFERENCES INGREDIENTES
(id_ingredientes)
    );
"",
'RECEITA_INGREDIENTES': ""
    CREATE TABLE RECEITA_INGREDIENTES (
        id_ingredientes INTEGER,
        id_receita INTEGER,
        FOREIGN KEY (id_ingredientes) REFERENCES INGREDIENTES
(id_ingredientes),
        FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita)
    );
"",
'USUARIO_META': ""
    CREATE TABLE USUARIO_META (
        id_usuario INTEGER,
        id_meta INTEGER,
        FOREIGN KEY (id_usuario) REFERENCES USUARIO (id_usuario),

```



```

        FOREIGN KEY (id_meta) REFERENCES META_NUTRICIONAL (id_meta)
    );
    """
    'INTOLERANCIA': """
        CREATE TABLE INTOLERANCIA (
            id_ingredientes INTEGER,
            id_usuario INTEGER,
            FOREIGN KEY (id_ingredientes) REFERENCES INGREDIENTES
(id_ingredientes),
            FOREIGN KEY (id_usuario) REFERENCES USUARIO (id_usuario)
        );
    """
    'AVALIACAO': """
        CREATE TABLE AVALIACAO (
            estrelas INTEGER,
            id_usuario INTEGER,
            id_receita INTEGER,
            FOREIGN KEY (id_usuario) REFERENCES USUARIO (id_usuario),
            FOREIGN KEY (id_receita) REFERENCES RECEITA (id_receita)
        );
    """

```

5.Consultas:

5.1 - Consulta 01:

Descrição: Obter o número de avaliações e média por receita.

Consulta:

```

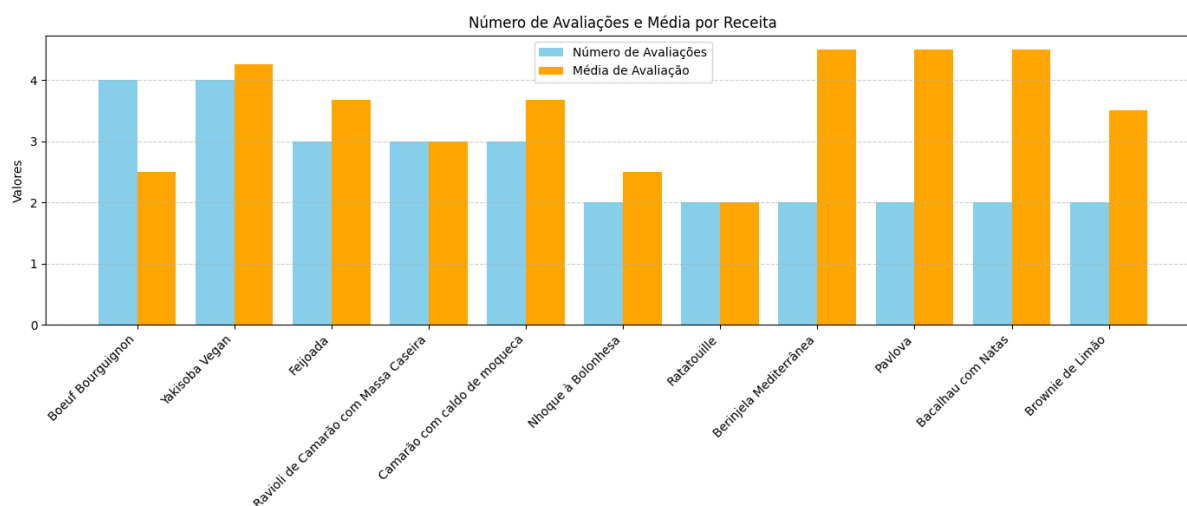
SELECT  R.nome AS receita, COUNT(A.estrelas) AS num_avaliacoes,
ROUND(AVG(A.estrelas), 2) AS media_avaliacao
FROM AVALIACAO A
JOIN RECEITA R ON A.id_receita = R.id_receita
JOIN USUARIO U ON A.id_usuario = U.id_usuario
GROUP BY R.nome
ORDER BY num_avaliacoes DESC;

```

Tabela gerada:

Receita	Número de Avaliações	Média de Avaliação
Boeuf Bourguignon	4	2.50
Yakisoba Vegan	4	4.25
Feijoadada	3	3.67
Ravioli de Camarão com Massa Caseira	3	3.00
Camarão com caldo de moqueca	3	3.67
Nhoque à Bolonhesa	2	2.50
Ratatouille	2	2.00
Berinjela Mediterrânea	2	4.50
Pavlova	2	4.50
Bacalhau com Natas	2	4.50
Brownie de Limão	2	3.50

Gráfico com o resultado:



5.2 - Consulta 02:

Descrição: Obter a quantidade de ingredientes e média de calorias por receita.

Consulta:

```

SELECT
  r.nome AS receita,
  COUNT(DISTINCT i.nome) AS quantidade_ingredientes,
  ROUND(AVG(t.calorias), 2) AS media_calorias
FROM
  RECEITA r
JOIN
  INGREDIENTES i ON r.id_receita = i.id_receita
JOIN
  TABELA_NUTRICIONAL t ON r.id_receita = t.id_receita

```

```

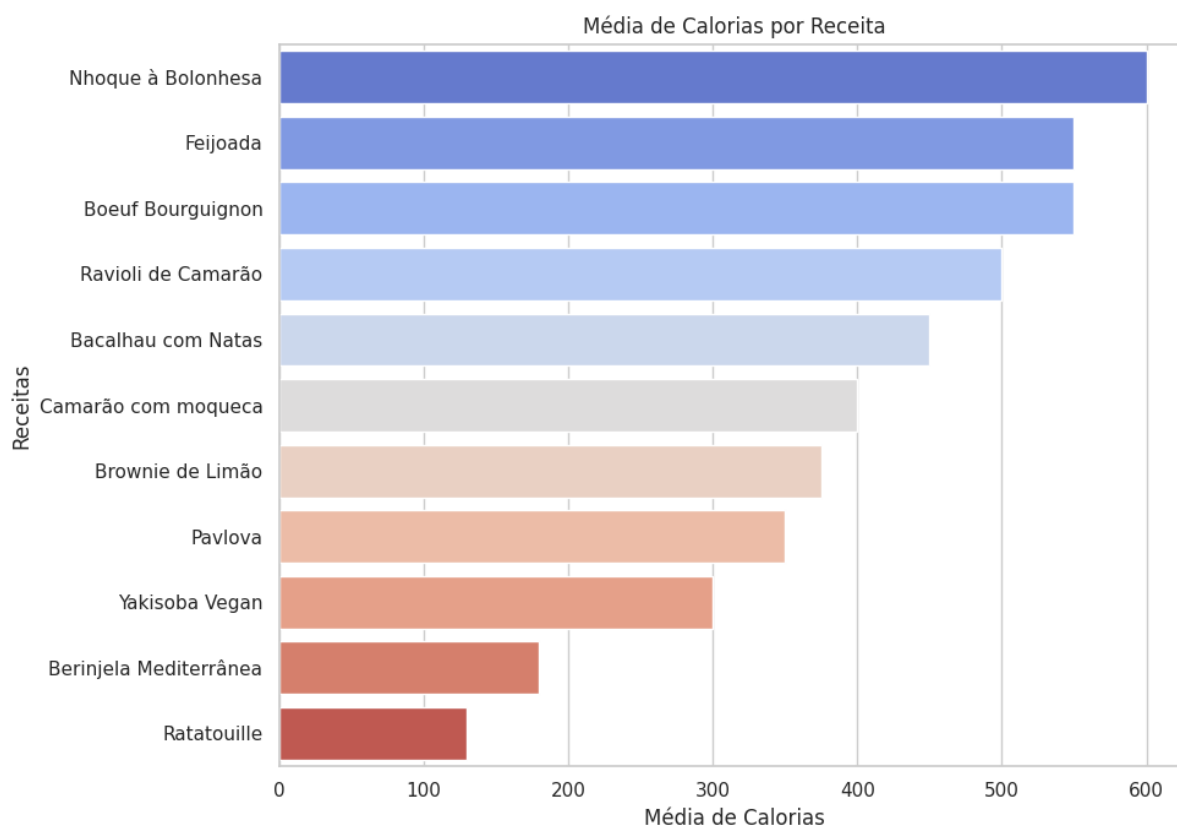
GROUP BY
  r.nome
ORDER BY
  media_calorias DESC;

```

Tabela gerada:

Receita	Quantidade de Ingredientes	Média de Calorias
Nhoque à Bolonhesa	5	600.00
Feijoada	5	550.00
Boeuf Bourguignon	5	550.00
Ravioli de Camarão com Massa Caseira	5	500.00
Bacalhau com Natas	5	450.00
Camarão com caldo de moqueca	5	400.00
Brownie de Limão	5	375.00
Pavlova	5	350.00
Yakisoba Vegan	5	300.00
Berinjela Mediterrânea	5	180.00
Ratatouille	5	130.00

Gráfico com o resultado:



5.3 - Consulta 03:

Descrição: Obter a quantidade de receitas avaliadas e média de estrelas por usuário.

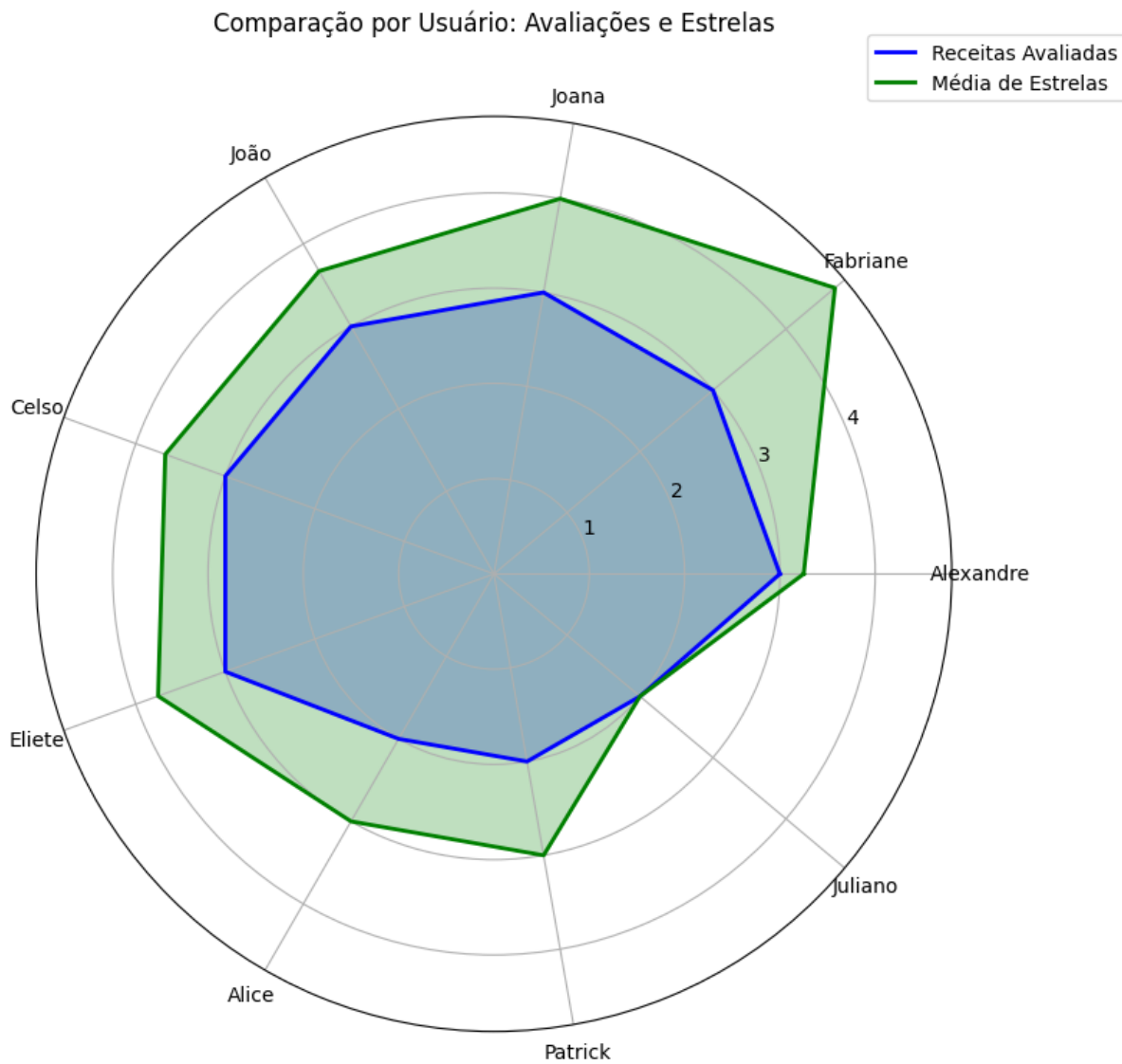
Consulta:

```
SELECT U.nome AS usuario, COUNT(DISTINCT A.id_receita) AS receitas_avaliadas,  
ROUND(AVG(A.estrelas), 2) AS media_estrelas  
FROM USUARIO U  
JOIN AVALIACAO A ON U.id_usuario = A.id_usuario  
JOIN RECEITA R ON A.id_receita = R.id_receita  
GROUP BY U.nome  
ORDER BY receitas_avaliadas DESC;
```

Tabela gerada:

Usuário	Receitas Avaliadas	Média de Estrelas
Alexandre	3	3.25
Fabiane	3	4.67
Joana	3	4.00
João	3	3.67
Celso	3	3.67
Eliete	3	3.75
Alice	2	3.00
Patrick	2	3.00
Juliano	2	2.00

Gráfico com o resultado:



7. Aplicação:

Foi implementado Grafo de Conhecimento a partir de um dado gerado, um csv autoral com os mesmos dados que foram preenchidos no Postgress. A partir do Embeddings gerado, foi realizado perguntas de validação e aprofundamento utilizando o Gemini.