

UNIVERSIDADE FEDERAL DE SÃO CARLOS - CAMPUS SOROCABA
CIÊNCIA DA COMPUTAÇÃO

BRUNA SCARPELLI
GUSTAVO SANCHES MARTINS KIS
RICARDO YUGO SUZUKI

PROJETO PRÁTICO DE PROCESSAMENTO MASSIVO DE DADOS:

Tema — Harmonização de vinhos e pratos

Grupo 14

Sistemas de Banco de Dados
Profa. Dra. Sahudy Montenegro González
Fase Intermediária I – 17/06/2025

Sorocaba - SP
2025

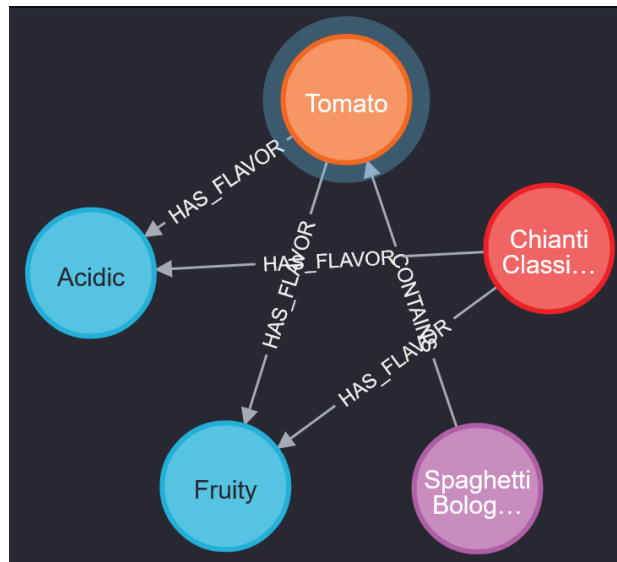
SUMÁRIO

1. DESCRIÇÃO DO PROBLEMA	3
2. IMPLEMENTAÇÃO	3
2.1. Tecnologias escolhidas	3
2.2. Fontes de dados	4
2.3. Etapas	4

1. DESCRIÇÃO DO PROBLEMA

Escolher o vinho certo para acompanhar uma refeição nem sempre é tão simples. Além das preferências pessoais, é preciso levar em conta fatores como os ingredientes do prato, o método de preparo e as características do vinho (acidez, corpo, taninos e notas aromáticas). Diante da enorme variedade de receitas e rótulos disponíveis, a tarefa de encontrar harmonizações que funcionem pode ser complexa.

Para tornar esse processo mais acessível e baseado em dados, nosso projeto propõe a construção de um banco de dados, alimentado por informações de receitas e vinhos, com todos os seus atributos e conexões de harmonização. Utilizando Neo4j para modelar as relações entre ingredientes e perfis de sabor dos vinhos, e Apache Spark para processar esses dados.



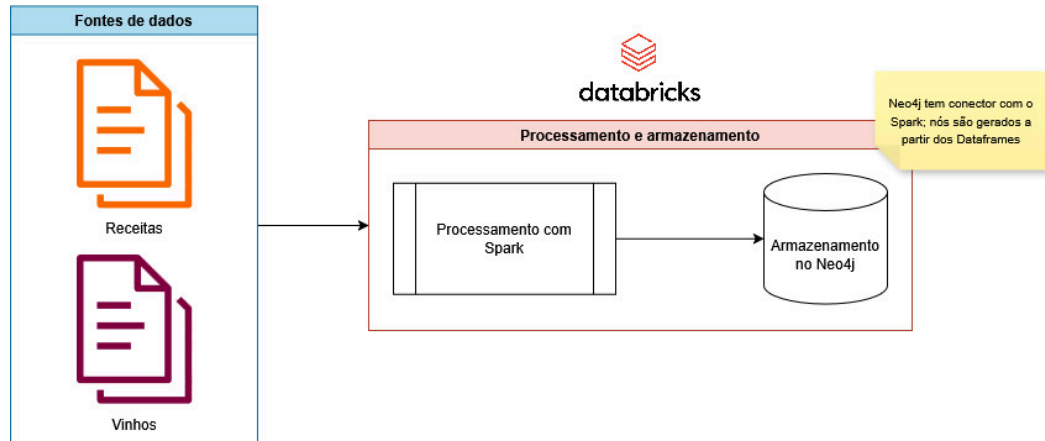
2. IMPLEMENTAÇÃO

2.1. Tecnologias escolhidas

Para o desenvolvimento deste projeto, foram escolhidas duas tecnologias principais: Apache Spark e Neo4j.

O Apache Spark foi selecionado pelo grupo para o processamento dos dados, pois é um framework amplamente adotado no mercado e oferece integração simplificada com outras ferramentas, incluindo o Neo4j.

Já o Neo4j foi escolhido para lidar com relacionamentos, já que o projeto busca explorar conexões entre perfis de vinhos e pratos. Uma solução baseada em grafos facilita a visualização dessas interações e permite identificar, de forma mais ágil, as combinações ideais.



2.2. Fontes de dados

- [Better Recipes for a Better Life](#) (recipes.csv): id, name, prep_time, cook_time, total_time, listas de ingredients e steps, valores nutricionais (calorias, gorduras totais, açúcares, proteínas, sódio, carboidratos etc.) e eventuais tags de dieta, tipo de refeição e nível de dificuldade.
- [Wine Dataset](#) (wines.csv): wine_id, name, description, price, capacity (ml ou cl), grape, secondary_grape_varieties, closure, country, region e, quando disponíveis, colunas de avaliação ou notas de degustação.
- Será criado um flavor_profiles.csv, que deve ter quatro campos: entity_type (“Ingredient” ou “Grape”), entity_name (nome conforme nos outros CSVs), flavor (ácido, doce, amadeirado, herbáceo, tânico, frutado, cítrico, terroso, gorduroso, picante) e intensity (low, medium ou high).

2.3. Etapas

1. Configuração do ambiente
 - a. Configuração do Databricks Community Edition.
 - b. Instalação de Neo4j Community Edition para modelagem de grafos.
2. Ingestão com Spark
 - a. Conectar Spark às fontes CSV (recipes.csv, wines.csv, flavor_profiles.csv).
 - b. Carregamento inicial em DataFrames.

- c. Limpeza (remoção de linhas vazias, padronização de nomes de ingredientes, etc).
- 3. Enriquecimento de receitas
 - a. Explosão da lista de ingredientes em linhas individuais dentro do DataFrame de receitas.
 - b. Associação de cada ingrediente ao seu perfil de sabor usando join com o DataFrame de perfis.
 - c. Agregação dos sabores por receita, resultando em um vetor de atributos de sabor para cada receita.
- 4. Enriquecimento de vinhos
 - a. Explosão das colunas de uvas principais e secundárias em linhas no DataFrame de vinhos.
 - b. Associação de cada uva ao seu perfil de sabor via join com perfis de sabor.
 - c. Agregação dos sabores por vinho, gerando vetor de atributos de sabor por vinho.
- 5. Modelagem no Neo4j
 - a. Definição do esquema de grafo:
 - i. `(:Recipe {id, name, total_time, ...})`
 - ii. `(:Wine {wine_id, name, price, ...})`
 - iii. `(:Ingredient {name})`
 - iv. `(:Grape {name})`
 - v. Relações `(:Recipe)-[:CONTAINS]->(:Ingredient)` e `(:Wine)-[:MADE_FROM]->(:Grape)`
 - vi. Relações de perfil de sabor `(:Ingredient)-[:HAS_FLAVOR {intensity, flavor}]->(:Flavor)` e similar para uvas.
 - b. Utilização do conector Spark–Neo4j.
- 6. Consultas
 - a. Elaboração de queries Cypher para identificar combinações de receita e vinho com perfis de sabor complementares.
- 7. Entrega do resultado
 - a. Geração de um relatório de recomendações extraído via Cypher
 - i. Apresentação dos principais achados: top-N harmonizações para diferentes perfis de prato e faixas de preço de vinho.