

## Estrutura de Produtos e Atributos no Sistema de Gestão de Lojas

Este documento detalha a estrutura de dados para produtos e seus atributos no Firebase Firestore. É crucial para entender como os itens são cadastrados, categorizados e gerenciados dentro do sistema de gestão de lojas.

### 1. Coleção de Atributos de Produtos (product\_attributes)

Esta é uma coleção de nível raiz, o que significa que não está aninhada sob outras coleções como stores. Ela serve como um repositório global para listas de valores pré-definidos (atributos) que podem ser usados por qualquer loja no sistema. O objetivo é manter a consistência e facilitar a gestão centralizada desses atributos.

Cada documento dentro da coleção product\_attributes armazena um tipo específico de atributo.

#### a) Documento categories

- Propósito: Armazena uma lista global de categorias de produtos disponíveis.
- DocumentId: categories (documento único)
- Campos Típicos:
  - list: Array de string
    - Exemplo: ['Camisetas', 'Calças', 'Vestidos', 'Acessórios', 'Calçados']

#### b) Documento colors

- Propósito: Armazena uma lista global de cores de produtos.
- DocumentId: colors (documento único)
- Campos Típicos:
  - list: Array de string
    - Exemplo: ['Preto', 'Branco', 'Vermelho', 'Azul', 'Verde', 'Amarelo']

#### c) Documento sizeGrades

- Propósito: Armazena diferentes "grades" ou grupos de tamanhos, permitindo flexibilidade para diversos tipos de produtos (ex: vestuário, calçados, infantil).
- DocumentId: sizeGrades (documento único)

- Campos Típicos:
  - Padrão: Object (para tamanhos de vestuário comuns)
    - Exemplo: { 'PP': 1, 'P': 2, 'M': 3, 'G': 4, 'GG': 5, 'XG': 6, 'XXG': 7, 'Único': 100 } (O valor numérico pode ser usado para ordenação no UI).
  - Infantil: Object
    - Exemplo: { 'RN': 1, 'P': 2, 'M': 3, 'G': 4 }
  - Calçados: Object
    - Exemplo: { '34': 1, '35': 2, '36': 3, '37': 4, '38': 5, '39': 6, '40': 7 }
  - *(Pode ter outros objetos para diferentes tipos de grade de tamanho)*

#### **d) Documento suppliers**

- Propósito: Armazena uma lista global de fornecedores de produtos.
- DocumentId: suppliers (documento único)
- Campos Típicos:
  - list: Array de string
    - Exemplo: ['Fornecedor Alpha', 'Fornecedor Beta', 'Distribuidora Gama']

## **2. Coleção de Produtos (products)**

Esta é uma coleção aninhada e está ligada a uma loja específica. Isso significa que cada loja tem seu próprio catálogo de produtos.

- Caminho Completo: artifacts/{apId}/stores/{storeId}/products/{productId}
- Propósito: Armazenar detalhes de cada produto disponível em uma loja, incluindo suas variações de estoque.
- Documentos: Cada documento representa um produto individual. O documentId pode ser gerado automaticamente ou ser um código de produto (SKU).

#### **Campos Típicos de um Documento de Produto:**

- code: string (Código SKU ou identificador único do produto).
- name: string (Nome do produto, ex: "Camiseta Algodão")
- description: string (Descrição detalhada do produto).

- category: string (Categoria do produto, deve ser um valor de product\_attributes/categories).
- supplier: string (Fornecedor do produto, deve ser um valor de product\_attributes/suppliers).
- costPrice: number (Preço de custo unitário do produto).
- salePrice: number (Preço de venda unitário sugerido do produto).
- totalQuantity: number (Quantidade total em estoque de todas as variações combinadas. Mantido atualizado pelo aplicativo).
- hasVariations: boolean (Indica se este produto possui variações de cor e/ou tamanho).
- variations: Array de Object (Este é o campo mais importante para estoque por variação).
  - Cada objeto dentro do array variations representa uma combinação específica de cor e tamanho e seu estoque:
    - color: string (Cor da variação, deve ser um valor de product\_attributes/colors).
    - size: string (Tamanho da variação, deve ser um valor de product\_attributes/sizeGrades).
    - quantity: number (Quantidade em estoque para esta variação específica).
    - barcode: string (Opcional, código de barras específico para esta variação).
    - isActive: boolean (Indica se esta variação está ativa para venda).
- imageUrl: string (Opcional, URL da imagem principal do produto).
- createdAt: Timestamp (Data e hora de criação do produto).
- updatedAt: Timestamp (Última atualização do produto).
- updatedBy: string (E-mail do usuário que fez a última atualização).

### 3. Relacionamento e Fluxo de Dados

1. Criação/Edição de Produtos: Ao cadastrar ou editar um produto, os campos category, color, size e supplier são preenchidos usando os valores

disponíveis nas coleções de `product_attributes`. Isso garante dados padronizados.

2. Gerenciamento de Estoque: A `totalQuantity` de um produto é a soma das `quantity` de todas as suas `variations`. As operações de venda (`SalesPage`) e entrada/saída de estoque (`StockMovements` - futuro) atualizam essas quantidades.
3. Filtragem por Loja: A estrutura aninhada `artifacts/{appld}/stores/{storeld}/products` é fundamental para garantir que cada loja acesse apenas seus próprios produtos, otimizando consultas e aplicando regras de segurança por `storeld`.

Esta estrutura permite uma gestão flexível de produtos com variações e um controle de estoque preciso por combinação de atributos, além de facilitar a escalabilidade e a segurança multi-loja no Firestore.