

**FACULDADE ANHANGUERA BETIM**  
**CIÊNCIA DA COMPUTAÇÃO**

**HIGOR COSTA**  
**SAMUEL WANDERSON M. SILVA**

**A FANTÁSTICA FÁBRICA DE CHOCOLATES**  
Projeto: Java + Mongo DB

**BETIM-MG**  
**2025**

HIGOR COSTA  
SAMUEL WANDERSON M. SILVA

## **NÃO TÃO FANTÁSTICA FÁBRICA DE CHOCOLATES**

Projeto: Java + Mongo DB

Realização de projeto para aprovação a  
disciplina de **POO 2** na Instituição  
Anhanguera

Orientador: Cynthia Barbosa

**BETIM-MG**  
**2025**

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>2</b>
<b>2</b>	<b>PLANEJAMENTO .....</b>	<b>3</b>
	2.1 Organização .....	3
	2.2 Ferramentas .....	4
<b>3</b>	<b>DESENVOLVIMENTO TÉCNICO .....</b>	<b>5</b>
	3.1 Funcionamento.....	5
	3.2 Principais Métodos e Lógica:.....	5
	3.3 Estrutura e integração com o banco de dados. ....	6
<b>4</b>	<b>CONCLUSÃO .....</b>	<b>7</b>
	4.1 Considerações Finais.....	7
	4.2 Próximas melhorias .....	7
	4.3 Aprendizados .....	8

## 1 INTRODUÇÃO

A grande ideia do nosso projeto é permitir que uma aplicação se integre a um banco de dados, podendo realizar operações de leitura, inserção, atualização e exclusão de dados em um contexto centrado em uma “Loja de Chocolates”.

Para bater esta meta, foi optado a utilização da Linguagem Java, que permite a criação de uma interface gráfica simples, com muitos recursos disponíveis (tabelas, labels etc) e portátil a vários sistemas. Para armazenar os dados de maneira intuitiva e otimizada o Mongo DB foi muito bem-vindo, já que ao invés de guardar esquemas complexos que estrutura de dados, ele permite que os dados sejam organizados puramente em texto, com a estrutura de um arquivo JSON.

O desenvolvimento do software é só uma parte do projeto, pois conhecer e aplicar boas práticas, **padrões de projetos** e utilização de meios que tornam o trabalho em equipe claro, organizado, e bem distribuído em relação as tarefas fizeram também parte do escopo do projeto

## 2 PLANEJAMENTO

### 2.1 Organização

Tivemos fortes influências do Kanban e do desenvolvimento colaborativo.

Reuniões: Tivemos pequenas reuniões periódicas, para que fosse mantido a comunicação frequente e regular, essencial para a adaptação e o alinhamento contínuo, que são pilares das metodologias ágeis. Nossa colaboração próxima foi essencial para o andamento do projeto.

Kanban (implícito pelo uso do Trello): O Trello é uma ferramenta que facilita a implementação de um sistema Kanban.

Visualizaram o fluxo de trabalho: As tarefas (funcionalidades do CRUD, bugs, melhorias) eram representadas como cartões que se moviam por diferentes estágios de “A fazer”, “Fazendo”, “Feito”.

Desenvolvimento Iterativo e Incremental: A organização das tarefas no Trello e as reuniões frequentes permitiram que o trabalho fosse dividido em partes menores e gerenciáveis.

Divisão de Tarefas: Baseada em Funcionalidades ou Componentes: Com o Trello, as tarefas podiam ser divididas por funcionalidades específicas do CRUD (Criar, Ler, Atualizar, Deletar para cada entidade do sistema), por componentes (ex: interface do usuário, lógica de negócios, acesso a dados) ou por histórias de usuário.

Trabalhar em conjunto: em trechos mais complexos e desafiadores foi realizado pair programming (pair programming).

Transparência e Sincronização: O Trello fornecia uma visão transparente do status de cada tarefa, e as reuniões serviam para sincronizar o trabalho, discutir impedimentos, planejar os próximos passos e refinar as tarefas no quadro.

## 2.2 Ferramentas

- Organizacionais
  - Trello
  - Microsoft 365
- Desenvolvimento
  - Java + Maven
  - Netbeans
  - Mongo DB + Compass
  - Git
  - Git Hub

### 3 DESENVOLVIMENTO TÉCNICO

#### 3.1 Funcionamento

O sistema apresenta uma janela principal com os seguintes componentes:

- **Cabeçalho:** Exibe uma imagem de logo da "fábrica de chocolates".
- **Formulário de Entrada:** Campos para inserir ou editar dados dos chocolates: Nome, Valor, Marca e Código de Barras.
- **Botões de ação**
  - **Adicionar:** Insere um novo produto no banco de dados com as informações dos campos de entrada.
  - **Atualizar:** Modifica um produto existente no banco de dados. O produto a ser atualizado pode ser selecionado na tabela ou identificado pelo código de barras informado.
  - **Excluir:** Remove um produto do banco de dados, selecionado pela tabela ou identificado pelo código de barras.
  - **Mostrar:** Atualiza a tabela, exibindo todos os produtos cadastrados no banco de dados.
- **Tabela de Produtos:** Lista os chocolates cadastrados, exibindo Nome, Valor, Marca e Código de Barras. Clicar em uma linha da tabela preenche automaticamente os campos do formulário com os dados do produto selecionado, facilitando a atualização ou exclusão.

#### 3.2 Principais Métodos e Lógica:

- **NemTaoFantasticaFabricaDeChocolates()** **(Construtor):** Inicializa a conexão com o banco de dados MongoDB, acessando a coleção "chocolates" no banco "loja".
- **ImageHeaderPanel()** **(Construtor da Classe Interna):** Carregar e preparar uma imagem para ser exibida como cabeçalho.
- **paintComponent()** **(em ImageHeaderPanel):** Desenhar a imagem do cabeçalho no painel, redimensionando-a e centralizando-a, ou exibir uma mensagem de erro.
- **adicionarProduto():** Validar dados e inserir um novo produto no banco de dados.

- **atualizarProduto()**: Validar dados e modificar um produto existente no banco de dados.
- **excluirProduto()**: Remover um produto do banco de dados após confirmação.
- **mostrarProdutos()**: Carregar e exibir todos os produtos do banco de dados na tabela da interface.
- **limparCampos()**: Limpar os campos de entrada de dados e a seleção da tabela.
- **main(String[] args)**: Iniciar a execução da aplicação gráfica.

### 3.3 Estrutura e integração com o banco de dados.

A conexão é estabelecida via MongoClient, que aponta para uma instância local do MongoDB. Os dados dos produtos são manipulados como **documentos BSON** (representados pela classe org.bson.Document no Java), que são equivalentes a objetos JSON.

Em vez de tabelas e linhas, o MongoDB organiza os dados em **coleções** (como "chocolates" no seu caso). As operações de **CRUD** (criar, ler, atualizar, excluir) são realizadas através de métodos do driver MongoDB (insertOne, find, replaceOne, deleteOne), que operam diretamente nesses documentos. A interface Swing captura as informações dos campos de texto, as converte em documentos e as envia para o banco, e vice-versa ao exibir os dados na JTable. Essa abordagem aproveita a flexibilidade do MongoDB, sem a necessidade de um esquema rígido ou mapeamento complexo, facilitando a interação entre a aplicação Java e o banco de dados NoSQL.



## 4 CONCLUSÃO

### 4.1 Considerações Finais

Este trabalho apresentou o desenvolvimento de um sistema CRUD (Create, Read, Update, Delete) para gerenciamento de produtos de chocolate, utilizando **Java Swing** para a interface gráfica e **MongoDB** como banco de dados. A aplicação demonstra a integração eficiente entre uma interface desktop intuitiva e a flexibilidade de um banco de dados NoSQL, permitindo o cadastro, visualização, atualização e exclusão de informações de produtos. O projeto alcançou seus objetivos, validando a viabilidade dessa arquitetura para soluções de gerenciamento de dados, e serve como base sólida para futuras expansões e aprimoramentos.

### 4.2 Próximas melhorias

#### RESUMO

- Criar outro programa, que se comunique com este. O programa existente organiza, gerencia e faz operações com os dados, a ideia é um novo app dedicado exclusivamente a realizar uma venda
- Utilizar-se dos recursos de imagem que o Mongo guarda
- Utilizar um Leitor de Código de Barras para montar a lista de compra de um cliente no novo app a ser criado

O software até então se limita a um sistema que consegue ser um repositório de informações para produtos baseados em chocolate, conseguimos ver, adicionar, atualizar e excluir. Seria mais interessante se a partir dessas informações pudéssemos simular uma compra de uma lista de produtos.

A ideia é que este repositório sirva a um outro aplicativo (que deve ser criado), que funciona de forma que uma operadora de caixa poderia exercer. Por exemplo, escaneando com um leitor de barras o código de um produto na loja com o aplicativo de pagamentos aberto, ele deve realizar uma requisição ao nosso software já criado, que lhe retornaria as informações de nome, preço etc. Assim, o propósito então iria de além de gerenciar as informações dos produtos, mas também servir a um app de consumo final.

### 4.3 Aprendizados

- Realizar integração de Java com Mongo DB
- Funcionamento do Maven (para Java)
- Resolução de erros e complicações frustrantes em Java/Netbeans
- Trabalho em equipe baseado em padrões
- Utilização de ferramentas organizacionais
- Desenvolvimento de aplicação embasada em Design Patterns
- Recursos de GUI Java Swing
- Recurso de documentação Java DOC do Netbeans
- Importação e Exportação de dados por meio de JSON (ou BSON)