

Curso Superior de Desenvolvimento de Software Multiplataforma

Eduardo Brando Almeida, 3011392313038

Felipe Thiago da Silva, 3011392313027

Paulo César Aparecido Quezada e Vasconcelos Victório,  
3011392313032

Thiago Saldanha Diegoli, 3011392313035

Projeto Interdisciplinar  
Gestão Ágil de Projetos de Software  
Desenvolvimento Web III  
Banco de Dados Não-Relacional  
Técnicas de Programação

"Projeto de pedido de Compras FATEC Votorantim"

Orientadores: Prof<sup>a</sup> Maria Janaína da Silva Ferreira

Prof<sup>o</sup> Jones Artur Gonçalves

Prof<sup>o</sup> Ricardo Roberto Leme

Prof<sup>o</sup> Rodrigo de Paula Diver

Votorantim  
Novembro, 2023

## **RESUMO**

O funcionamento de uma instituição de ensino acontece com base em muitas relações, tal como entre discentes e docentes. Um relacionamento essencial ocorre

com o departamento de compras, que garante o funcionamento diário básico da organização envolvida, seja na educação ou no conforto dos professores e alunos.

Este projeto foi pensado como um sistema de pedido compras utilizado pelos responsáveis de compras de materiais da Fatec Votorantim baseado no padrão da BEC (Bolsa Eletrônica de Compras de São Paulo).

**Palavras-Chave:** Compras; NodeJS; FATEC; JavaScript; MongoDB

|                                       |    |
|---------------------------------------|----|
| DESCRIÇÃO DO PROJETO .....            | 4  |
| Proposta do Software (Objetivo) ..... | 4  |
| Justificativa .....                   | 4  |
| Mapa Mental.....                      | 5  |
| Levantamento de Requisitos.....       | 6  |
| 2.2. Requisitos Funcionais.....       | 7  |
| 2.3. Requisitos Não Funcionais .....  | 7  |
| 2.4. Diagrama de Caso de Uso .....    | 8  |
| PROJETO DO SOFTWARE .....             | 8  |
| Arquitetura da Aplicação .....        | 8  |
| Tecnologias Utilizadas.....           | 9  |
| Modelo de Dados .....                 | 9  |
| Modelo em Documentos .....            | 9  |
| Diagrama de Classes .....             | 11 |
| REFERÊNCIAS.....                      | 14 |

## DESCRIÇÃO DO PROJETO

### Proposta do Software (Objetivo)

Baseado nas necessidades dos funcionários do uso constante de novos materiais, o usuário utilizará o sistema para armazenar dados sobre produtos que serão utilizados posteriormente em um novo pedido de compra, de forma que organize opções de compras e otimize tempo de pesquisa.

Com este software professores poderão, por exemplo, realizar um pedido de materiais utilizados em aula, assim como - na necessidade de uma reforma do prédio ou de uma atividade prática - outros funcionários podem requisitar materiais mais robustos diretamente ao comprador, que analisará a requisição e se o pedido será realizado ou não, retornando a decisão ao cliente.

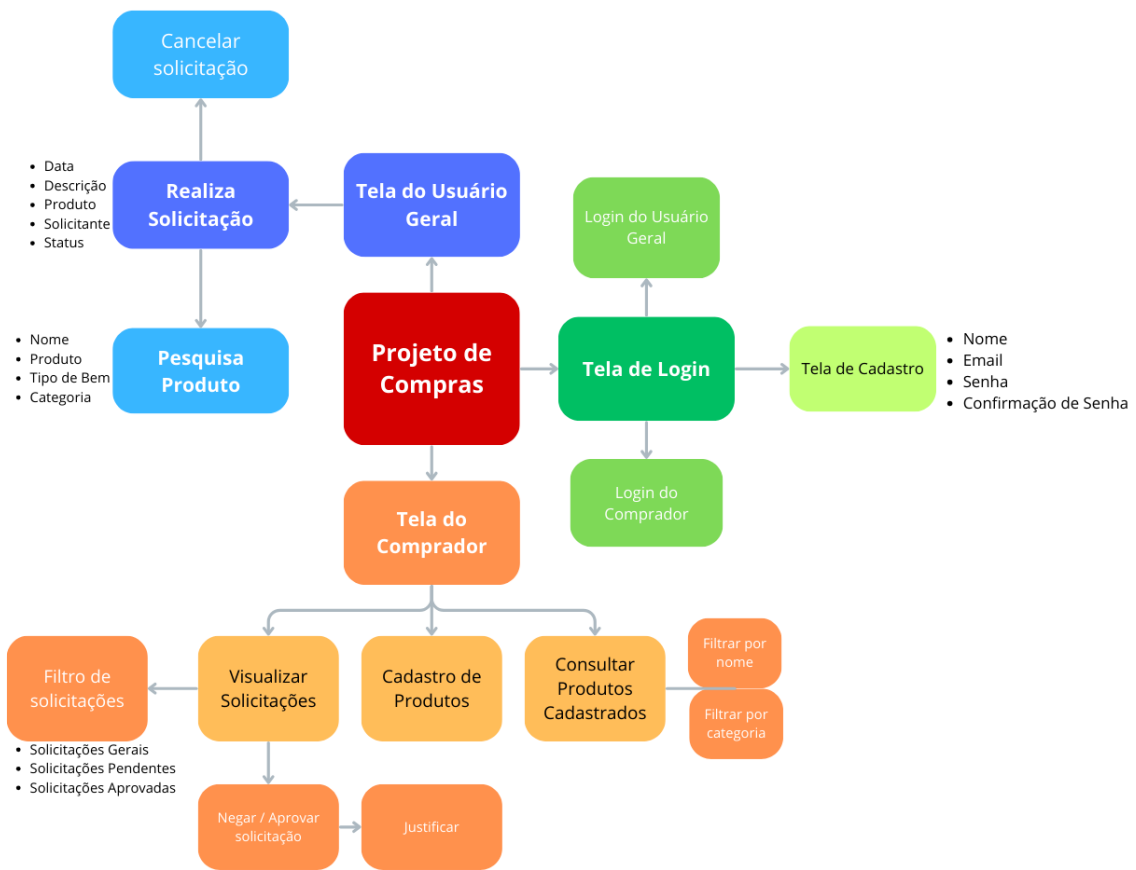
Com foco nesta ideia, o aplicativo utiliza de uma API desenvolvida para cadastrar produtos constantemente envolvidos com o ambiente, sendo permitida a consulta, inclusão, mudança e exclusão de cada um.

### Justificativa

O desenvolvimento deste site se deu com base na necessidade de um sistema que unificasse dados que o comprador coleta entre os funcionários da instituição, deixando tudo de maneira mais organizada e reaproveitável.

Sendo a FATEC Votorantim uma organização recém-formada, a construção deste projeto se justifica na garantia de uma comunicação limpa entre os departamentos, propondo um melhor ambiente de educação para todos.

## Mapa Mental



## REQUISITOS DO PROJETO

### Levantamento de Requisitos

A necessidade de pontos chave da aplicação desenvolvida foi consultada diretamente com os funcionários responsáveis pelos pedidos de compras, sendo a própria compradora da instituição e a coordenadoria. Desta forma, foi realizada uma entrevista em que foram coletadas as informações sobre processos já realizados diariamente, práticas costumeiras, a lógica de negócio envolvida e dificuldades a serem atendidas.

Nesse contexto, foi observado que seria útil um software que reunisse todos os dados que são repassados verbalmente pelos funcionários, pois há muitos desencontros e desentendimentos em relação à aceitação da compra destes materiais pelo governo.

Ao passo que um funcionário se mantém atualizado sobre pedidos correntes, espera-se criar uma cultura de maior compreensão quanto aos critérios para se fazer uma solicitação. Alguns desses critérios envolvem o preço, quantidade, tipo de bem, procedência e repetição de fornecedor e repetição de pedidos de um produto que, com as devidas variáveis, não podem ser realizados ao governo.

Esta conversa serviu de modelo principal para a construção do projeto, mas também foram fornecidos exemplos de serviços já utilizados pelo usuário que guiaram a construção das classes, como o serviço da “Bolsa Eletrônica de Compras de São Paulo” (BECSP), além de planilhas com exemplos de dados a serem cadastrados em nosso sistema.

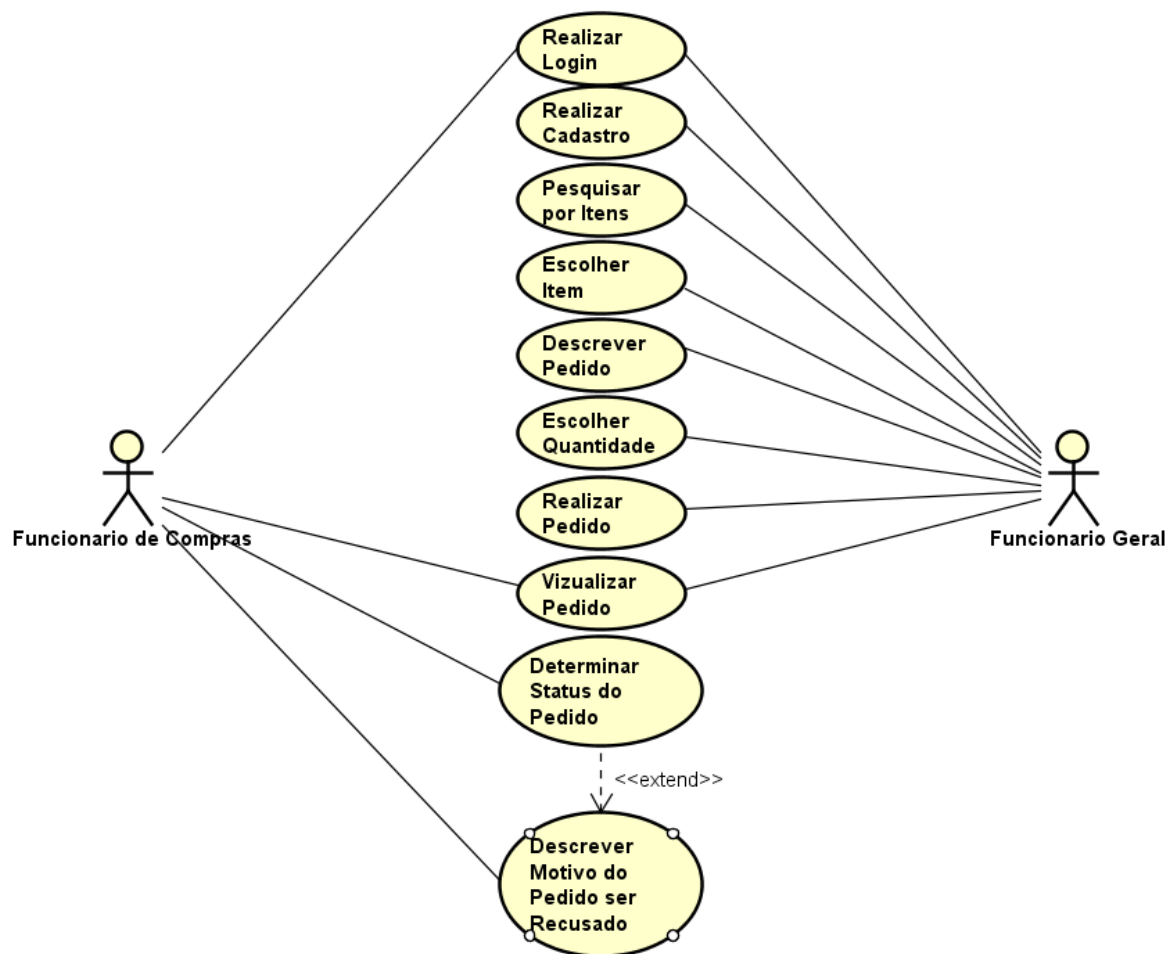
## 2.2. Requisitos Funcionais

| Requisitos Funcionais (RF) |   |
|----------------------------|---|
| Nº Requisito               | Descrição   |
| <b>RF01</b>                | Realizar Login  |
| <b>RF02</b>                | Realizar Cadastro   |
| <b>RF03</b>                | Como funcionário, pesquisar por itens                             |
| <b>RF04</b>                | Como funcionário, escolher um item                                |
| <b>RF05</b>                | Como funcionário, realizar um pedido                              |
| <b>RF06</b>                | Descrever o motivo do pedido                                      |
| <b>RF07</b>                | Ver o pedido realizado  |
| <b>RF08</b>                | Como administrador, visualizar um pedido feito por um funcionário |
| <b>RF09</b>                | Como administrador, determinar o status do pedido                 |
| <b>RF10</b>                | Como administrador, descrever por que o pedido foi recusado       |

## 2.3. Requisitos Não Funcionais

| Requisitos Não Funcionais (RNF) |  |
|---------------------------------|--|
| Nº Requisito                    | Descrição  |
| <b>RNF01</b>                    | Utilizar HTML e CSS  |
| <b>RNF02</b>                    | Utilizar JavaScript e NodeJS   |
| <b>RNF03</b>                    | Criar banco de dados NoSQL (MongoDB)   |
| <b>RNF04</b>                    | Autenticação de usuários via JWT   |
| <b>RNF05</b>                    | Modo de administrador  |
| <b>RNF06</b>                    | Implementar API de pesquisa da BEC   |
| <b>RNF07</b>                    | Pedido é criado com um status que tem como padrão no banco de dados “em aguardo” |
| <b>RNF08</b>                    | O status tem 3 status além do padrão (concluído, em processo e negado)           |

## 2.4. Diagrama de Caso de Uso



## PROJETO DO SOFTWARE

### Arquitetura da Aplicação

A arquitetura utilizada neste projeto é totalmente baseada em REST, consistindo em requisições HTTP (de coleta, inserção, mudança e remoção de dados) que agem de acordo com a API do site. Com base nessa arquitetura foi possível definir uma listagem de dados básica e o funcionamento de formulários, que por sua vez permitem ações como login, registro e filtragens de dados.



## Tecnologias Utilizadas

As tecnologias utilizadas no planejamento e a construção deste software foram as seguintes:

- Visual Studio Code
- JavaScript
- NodeJS e demais microserviços
- MongoDB
- SkillIcons
- Vercel
- GitHub

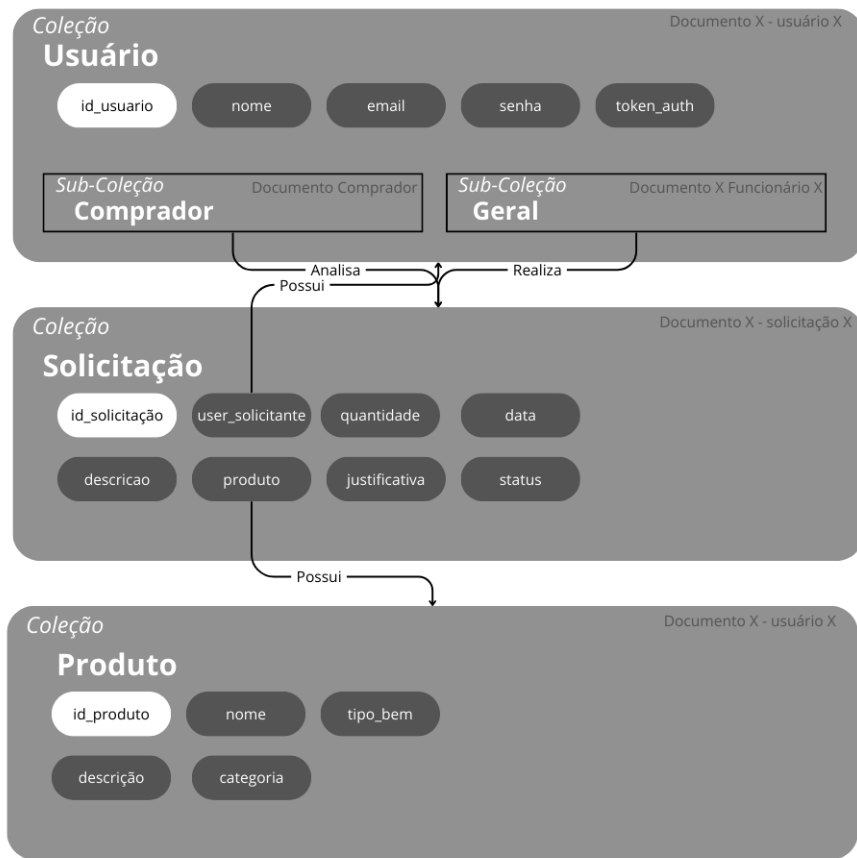
## Modelo de Dados

Por se tratar de um sistema baseado em banco de dados não-relacional, não é possível determiná-lo a partir dos modelos de um banco relacional. Uma vez que não há “tabelas” e relacionamentos exatos para se definir, os documentos se relacionam de maneira muito mais dinâmica e volátil.

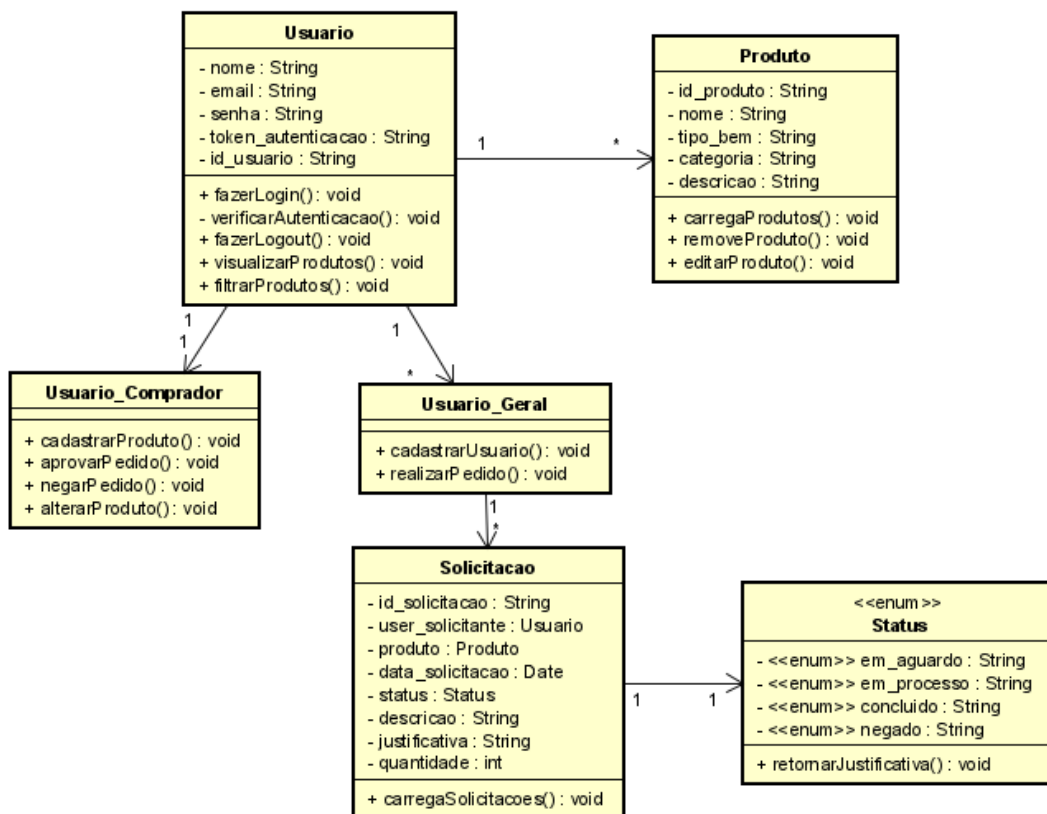
Entretanto, a fins de comparação, ainda podemos definir este banco de dados a partir de uma modelagem baseada em documentos, estes que, na interação entre os dados, acontece por JSON e podem se relacionar a qualquer momento com qualquer instância do projeto.

## Modelo em Documentos

O seguinte modelo demonstra as coleções envolvidas no banco de dados não-relacional como documentos, de que forma elas se relacionam e os dados de cada um.



## Diagrama de Classes



O diagrama de classes dessa aplicação foi montado pensando no paralelo entre dois tipos de usuários: o comprador e o usuário geral (funcionário).

Um usuário, independentemente de sua função, possui informações básicas de nome, e-mail e senha, ao passo que também guarda um token de autenticação que garantirá a segurança do site não permitindo “invasões” a outras páginas sem o login. Desta forma, esse token poderá estar vazio ou contendo uma String gerada automaticamente pela API. O token se torna inválido depois de 24 horas.

Todo usuário pode realizar e desfazer login. Ao acessar qualquer página do site, disparará um método que verificará se seu token é atualmente válido. Qualquer usuário pode visualizar uma lista com os produtos cadastrados para consulta ou posterior uso, tal como filtrá-los quando necessário.

Partindo desta ideia, somente o comprador possui a função de cadastrar os produtos que precisar para futuras solicitações, para isso preenchendo um formulário que chamará o método “cadastrarProduto”, salvando os dados no banco.

Um produto também possui dados básicos de nome, categoria e descrição, mas também é composto pelo tipo de bem (se é material de consumo ou permanente), uma informação crucial para a análise do comprador. Entre seus métodos estão “remover” e “editar”, ações realizadas pelo comprador primariamente pelo método “alterarProduto”. Ao mesmo tempo, possui um método de carregamento chamado de maneira assíncrona em toda lista de consulta de produtos.

Em uma tela exclusiva, também haverá uma lista de todas as solicitações feitas por usuários. O comprador pode selecionar uma solicitação e decidir se a aprovará ou negará, o que fará aparecer um novo campo para a justificativa da decisão.

Por outro lado, somente usuários comuns podem se cadastrar (já que há somente um comprador definido para a Fatec automaticamente cadastrado por seu e-mail) - ato realizado por “cadastrarUsuario” -, ao passo que podem realizar seus próprios pedidos ao comprador, ativando o método “realizarPedido” pelo botão de um formulário, que salvará os dados e os transportarão diretamente à lista do comprador.

Seguindo o fluxo, ao realizar um pedido um usuário geral salvará esses dados em forma de uma classe “Solicitação”, que por sua vez possui dados de identificação do solicitante e do produto; a data de envio, o produto requisitado e uma pequena descrição do uso deste produto. Essa classe também carrega de maneira assíncrona um método de carregamento. Além disso, a classe Solicitação possui um dado de Status, que é definido automaticamente como “em aguardo”, e mudará com base nas decisões do comprador.

Sendo os status pré-definidos, surge então uma nova classe do tipo enum, que possui um método “retornarJustificativa” que age caso a decisão do

comprador seja de negar um produto (isso preencherá o campo “justificativa” da classe de Solicitação e a informação será retornada ao usuário geral em seu histórico de pedidos).

## REFERÊNCIAS

<https://aws.amazon.com/pt/nosql/document/>

<https://periodicos.ifpb.edu.br/index.php/principia/article/viewFile/495/329>

SOMMERVILE, Ian. Engenharia de Software. 8ª ed. Pearson, 2007

VILLAIN, Mateus. Figma: o que é a ferramenta, Design e uso. Disponível em: <https://alura.com.br/artigos/figma>. Acesso em: 03 out. 2023

<https://nocodestartup.io/modelagem-dados-nosql-firebase/>