

# Projeto: Evernote Clone B

versão: 20211215\_1008

## Descrição

O Software Evernote é uma ferramenta destinada à organização de informação pessoal baseada em notas (anotações). A proposta aqui apresentada possui como objetivo a criação de uma solução parecida porém de escopo menor que o Evernote. A ferramenta deve permitir aos usuários salvar, editar e excluir anotações; Como se fosse um bloco de notas online, acessível por navegador de internet (Browser).

## Requisitos

**RF\_001:** O sistema deve permitir que seja mantido um cadastro de notas, que serão criadas e gerenciadas pelos usuários. **Prioridade:** Essencial.

**RF\_002:** O sistema deve permitir que seja atribuído nas notas, uma categoria. Cada nota poderá ter uma categoria apenas. **Prioridade:** Importante.

**RF\_003:** O sistema deve permitir que seja possível a atribuição de etiquetas nas notas. Cada nota poderá ter diversas etiquetas associadas. **Prioridade:** Importante.

**RF\_004:** O sistema deve permitir exibir todas as notas, notas de uma categoria escolhida pelo usuário e/ou notas que possuam alguma etiqueta escolhida pelo usuário. **Prioridade:** Importante.

**RF\_005:** O sistema deve persistir os dados, para que seja possível aos usuários salvar uma nota hoje e lê-la novamente nos dias seguintes. **Prioridade:** Essencial.

# Modelagem de Dados

Abaixo segue uma possível modelagem de dados.

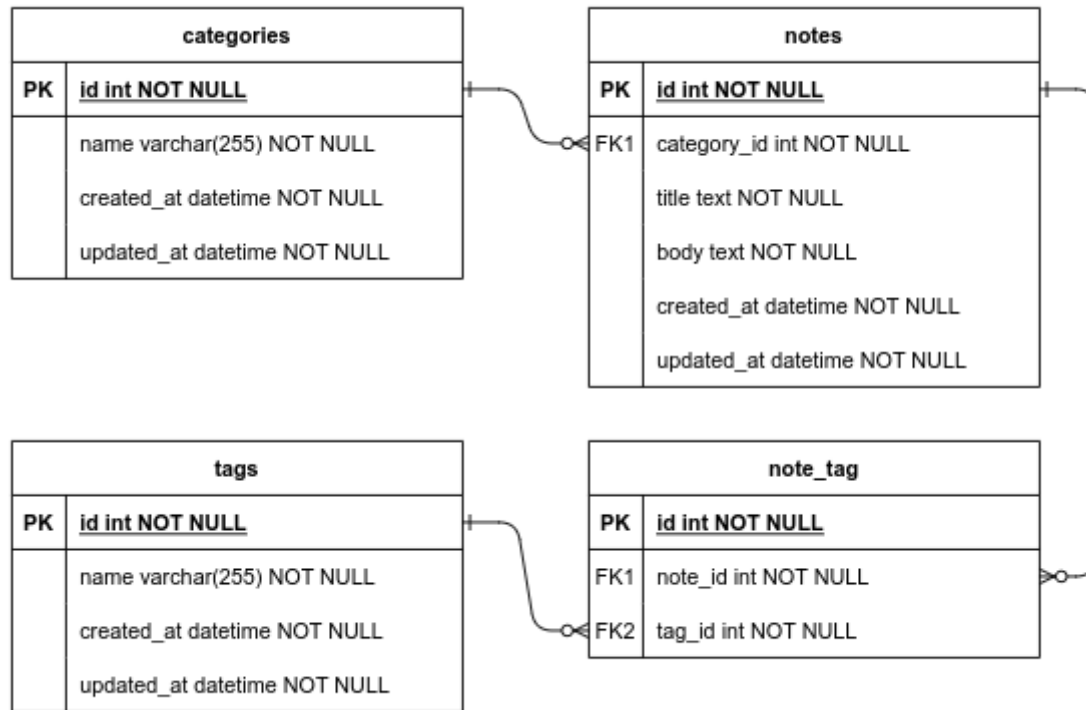


Figura 1: Modelagem de dados Evernote Clone B

# Wireframes e Sitemap

Abaixo segue uma sugestão de *Wireframes* e o *Sitemap* esperado para o projeto.

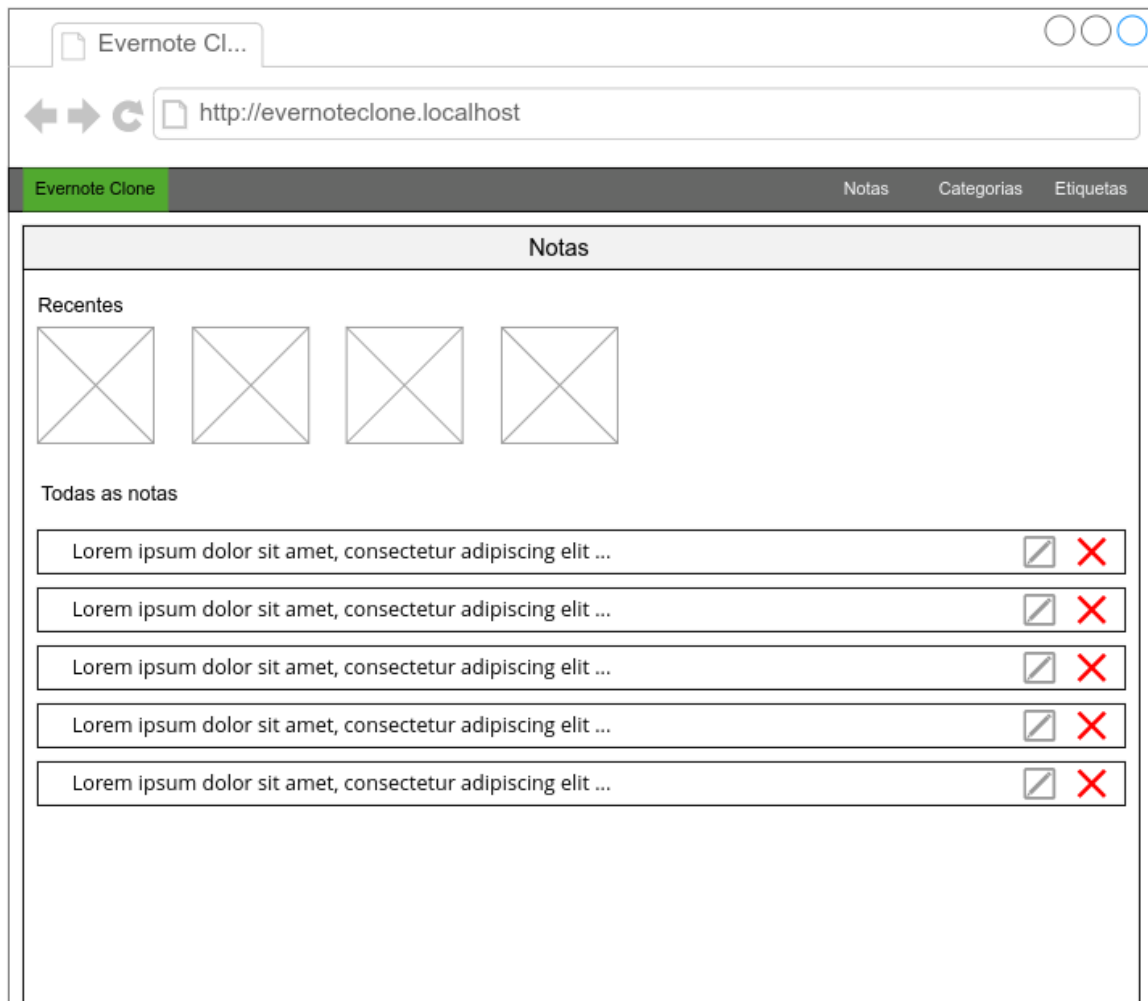


Figura 2: *Wireframe* da tela inicial, exibindo as anotações (notas), sugestão.

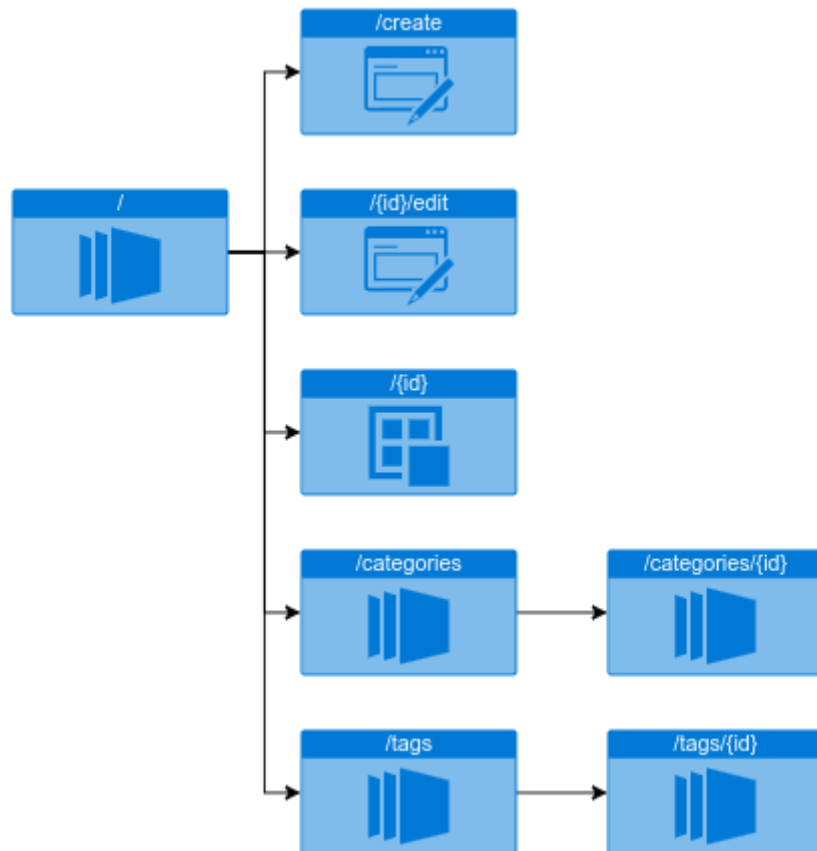


Figura 3: *Sitemap* esperado da ferramenta / solução.

Sobre o sitemap: é comum em aplicações web mapearmos as telas, os possíveis caminhos, em rotas da aplicação. No caso do sitemap proposto, há 8 rotas que são:

- **/** = página inicial da aplicação, a raiz da aplicação web.
- **/create** = página para a criação de anotações (notas).
- **/ {id} /edit** = página para edição de anotações (notas), aqui o símbolo {id} se refere a um parâmetro de *Uniform Resource Locator* (URL), este parâmetro é o valor da coluna id na tabela de notas, tabela onde ficam armazenadas as notas.
- **/ {id}** = página de exibição do conteúdo da nota.
- **/categories** = página de exibição das categorias presentes no sistema.
- **/categories/{id}** = página de exibição das notas relacionadas com uma categoria escolhida. Aqui o {id} refere-se ao valor da coluna id da tabela de categorias.
- **/tags** = página de exibição das etiquetas presentes no sistema.

- **/tags/{id}** = página de exibição das notas de uma determinada etiqueta. aqui o {id} indica qual é a etiqueta que foi solicitada a listagem das notas que possuem referência a mesma.

## Dicas, Materiais de apoio e tutoriais

Abaixo segue uma lista de alguns materiais que acreditamos que possam ajudar na construção da solução.

1. Irias, A. **MVC conceito e exemplo em Node.js**. Disponível em <  
<https://irias.com.br/blog/mvc-conceito-e-exemplo-em-node-js/> >.
2. Michel C. **CRUD application with Express and PostgreSQL in 10 steps**. Disponível em <  
<https://blog.pagesd.info/2019/10/15/crud-with-express-postgresql-10-steps/> >.
3. Okoye, C. **Creating a To-do App with Node.js + Express and PostgreSQL database**. Disponível em <  
<https://codeburst.io/creating-a-todo-app-with-node-js-express-and-postgresql-data-base-6dbf80c3fe9c> >.
4. Didin J. **Node, Express, Sequelize, and PostgreSQL Association Example**. Disponível em <  
<https://www.djamware.com/post/5bb1f05280aca74669894417/node-express-sequelize-and-postgresql-association-example> >.
5. Sequelize. **Associations**. Disponível em <  
<https://sequelize.org/v4/manual/tutorial/associations.html> >.
6. Rocketseat. **CRUD com Node.JS, Express, TypeORM e PostgreSQL - #code/drops 113**. Disponível em <  
<https://www.youtube.com/watch?v=9AO2hZJsHrs> >.
7. Duarte. L. **Tutorial de CRUD com Node.js, Sequelize e PostgreSQL**. Disponível em <  
<https://www.luiztools.com.br/post/tutorial-de-crud-com-node-js-sequelize-e-postgresql/> >.
8. Chege, J. **Building a RESTful Web API in Node.js using PostgreSQL and Express**. Disponível em <  
<https://www.section.io/engineering-education/restful-web-api-using-nodejs-postgresql-and-express/> >