



UNIVERSIDADE SALVADOR

EVERTON NAZARÉ DO SACRAMENTO  
JOSÉ MIRALDO MASCARENHAS DIAS JÚNIOR  
LEONARDO OLIVERA DA SILVA SANTOS  
RAFAEL MIRANDA LIMA

RELATÓRIO DE DESENVOLVIMENTO DO CRUD DE UMA APLICAÇÃO  
WEB PARA A ESCOLINHA DO PATO

Salvador

2022

EVERTON NAZARÉ DO SACRAMENTO  
JOSÉ MIRALDO MASCARENHAS DIAS JÚNIOR  
LEONARDO OLIVERA DA SILVA SANTOS  
RAFAEL MIRANDA LIMA

## RELATÓRIO DE DESENVOLVIMENTO DO CRUD DE UMA APLICAÇÃO WEB PARA A ESCOLINHA DO PATO

Relatório de requisito de software do sistema de gerenciamento de transporte universitário apresentado à unidade curricular Usabilidade, Web, Mobile como requisito parcial para a aprovação no curso de Ciência da Computação da Universidade Salvador.

Professor(es): Adailton de Jesus Cerqueira Junior;  
Wellington Lacerda Silveira da Silva.

Salvador  
2022

## **Sumário**

<b>1.0 Prefácio</b>	<b>4</b>
<b>2.0 Glossário</b>	<b>5</b>
<b>3.0 Introdução</b>	<b>6</b>
<b>4.0 Modelos do projeto</b>	<b>6</b>
<b>4.1 Descrição do sistema</b>	<b>6</b>
4.1.1 Com relação a Aluno	6
4.1.2 Com relação a Professor	6
4.1.3 Com relação a Disciplina	7
4.1.4 Com relação a Boletim	7
<b>4.2 Requisitos Funcionais</b>	<b>7</b>
<b>4.3 Requisitos não funcionais</b>	<b>7</b>
<b>4.4 Diagrama de Caso de Uso</b>	<b>9</b>
<b>4.6 Diagrama de Atividade</b>	<b>10</b>
<b>4.5 Diagrama de Entidade Relacionamento</b>	<b>10</b>
<b>5.0 Processo de Criação</b>	<b>10</b>
<b>6.0 Resultados</b>	<b>10</b>
<b>7.0 Conclusão</b>	<b>11</b>
<b>8.0 Bibliografia</b>	<b>11</b>

## **1.0 Prefácio**

A finalidade deste documento é mostrar a projeção de uma aplicação web para uma escola, com front-end e back-end. A aplicação não será o site da escola, mas uma componente do mesmo.

Esse documento destina-se aos programadores, projetistas, analistas de sistemas, desenvolvedores, testadores, usuários e demais interessados.

Versão 1.0

## 2.0 Glossário

API: (Application Programming Interface ou Interface de Programação de Aplicativos) é a forma que temos de nos comunicar com as funções pré-programadas definidas pelo fornecedor, no caso, Google;

Framework: São estruturas compostas por um conjunto de códigos genéricos que permite o desenvolvimento de sistemas e aplicações;

Front end: É a interface gráfica do projeto, onde se desenvolve a aplicação com a qual o usuário irá interagir diretamente;

Back end: É o código que conecta a internet com o banco de dados, gerencia as conexões dos usuários e alimenta a aplicação web;

CRUD: Esse termo descreve as quatro operações (Criação, Atualização, exclusão e Visualização) básicas para criar e gerenciar elementos de dados persistentes, especialmente em bancos de dados relacionais ;

Processador CSS: linguagem que tem como objetivo gerar folhas de estilos, sendo usado para customizar documentos;

Node.js: É um conjunto de API's que funciona como um interpretador de JavaScript;

Express.js: É um framework para Node.js que fornece recursos para construção de servidores web.

React.js: É uma biblioteca front-end JavaScript

Requisitos não funcionais: São os requisitos relacionados ao uso da aplicação

Requisito funcional: Representa o que o software faz, em termos de tarefas e serviços.

## 3.0 Introdução

O sistema foi desenvolvido para que seja um componente de uma aplicação web, mais especificamente um CRUD, facilitando as questões de cadastro, exclusão e atualização dos dados. Para a realização do projeto utilizamos: framework front : React, biblioteca front: axios, processador css: sass, biblioteca para navegação: React Router Dom, Back End: Node Js, Biblioteca back: express, banco de dados: PostGres e uma instância no Elephant SQL, biblioteca de interface de comunicação com o banco: PG, API REST.

## 4.0 Modelos do projeto

### 4.1 Descrição do sistema

O sistema apresenta uma homepage onde no rodapé estão disponíveis todas as informações da empresa, por exemplo, CNPJ, endereço e meios de comunicação. Também disponibiliza o menu de opções para o funcionário da escola realizar o cadastramento, alteração, exclusão e consulta de estudante/professor/disciplina. Logo em seguida, centralizado no cabeçalho está localizado a logo, em que, toda a página segue a mesma paleta de cores.

#### 4.1.1 Com relação a Aluno

O sistema deve disponibilizar para o funcionário da escola realizar as operações de cadastramento, alteração, exclusão e consulta do aluno. Os dados necessários para o cadastro deste formulário são:

- Número de matrícula –
- Nome do Aluno –
- CPF –
- Modulo Atual –

O aluno novo sempre entra no módulo 1 e só poderá se matricular no próximo módulo se for aprovado no módulo anterior. Se foi reprovado, só pode se matricular no módulo atual..

#### 4.1.2 Com relação a Professor

O sistema deve disponibilizar para o funcionário da escola realizar as operações de cadastramento, alteração, exclusão e consulta do professor. Os dados necessários para o cadastro deste formulário são:

- Id do Professor–
- Nome do Professor –
- Título acadêmico –

- CPF –
- Disciplina responsável –

#### **4.1.3 Com relação a Disciplina**

O sistema deve disponibilizar para o funcionário da escola realizar as operações de cadastramento, alteração, exclusão e consulta da Disciplina. Os dados necessários para o cadastro deste formulário são:

- Id da Disciplina–
- Desc disciplina
- Módulo –

#### **4.1.4 Com relação a Boletim**

O sistema disponibiliza o boletim individualmente para cada aluno e respectivamente seus módulos. Os dados que constam no boletim são:

- Nome do aluno –
- Turma –
- Nota final –
- Resultado –

### **4.2 Requisitos Funcionais**

1. Manter estudante: O sistema deve permitir realizar o cadastro, alteração e exclusão de um perfil de estudante.
2. Manter Professor: O sistema deve permitir realizar o cadastro, alteração e exclusão de um perfil de professor.
3. Manter Disciplina: O sistema deve permitir realizar o cadastro, alteração e exclusão de um perfil de Disciplina.
4. Manter Módulo: O sistema deve permitir realizar o cadastro, alteração e exclusão de um perfil de Módulo.

### **4.3 Requisitos não funcionais**

1- Interoperabilidade: O sistema deve possuir fácil comunicação com o banco de dados. POSTGRESQL é um framework para desenvolvimento e teste de API 's.;

2- Éticos: O sistema não deve conceder dados privados de outros usuários.

Apenas usuários credenciados podem editar rotas, consultar pessoas cadastradas.;

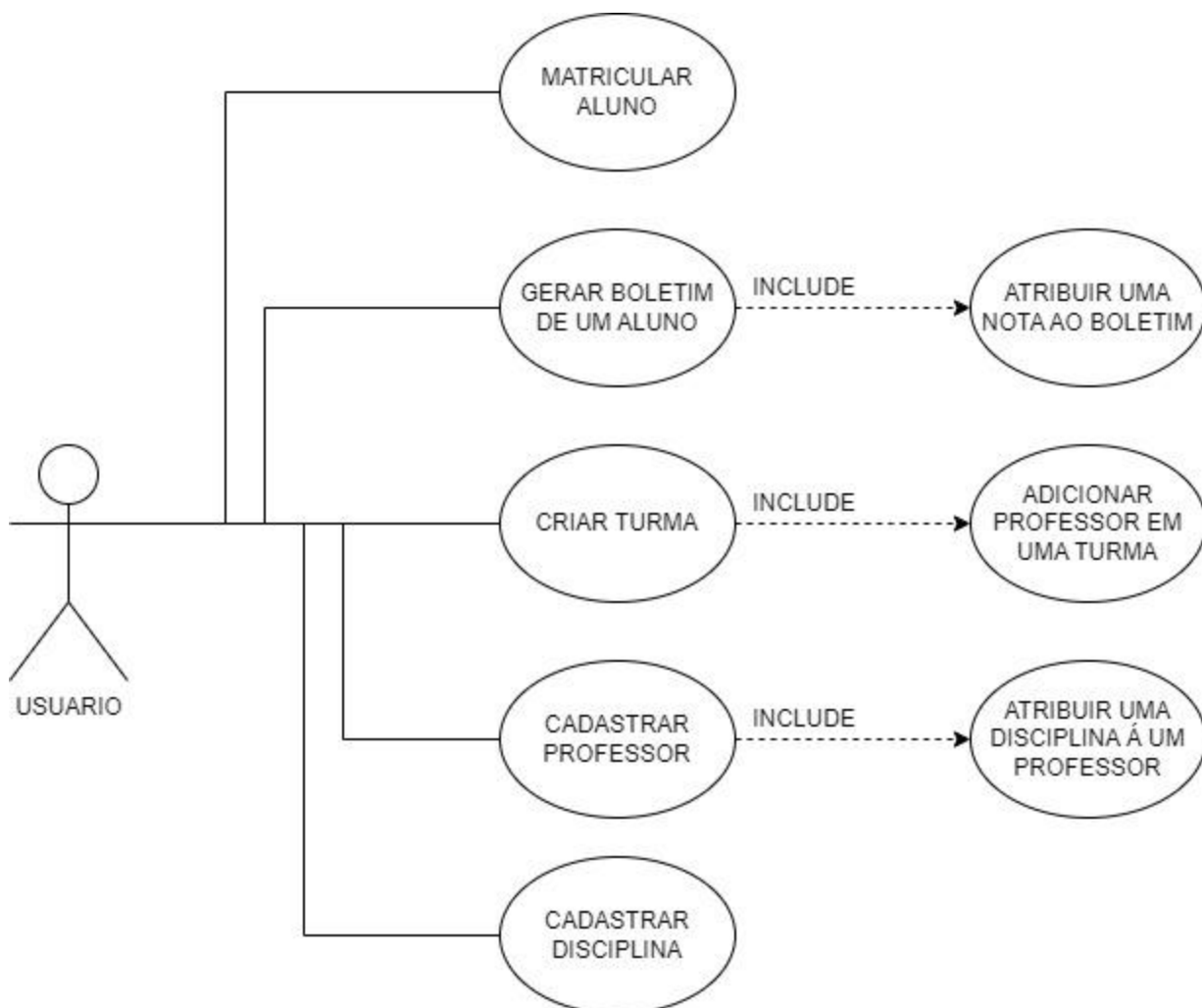
3- Segurança: O sistema deve seguir normas de proteção de dados para seus usuários.

4- Legais: Deve estar de acordo com as normas de LGPD;

5- Facilidade de uso: Operador do sistema deve utilizar o sistema com facilidade após um dia de treinamento;

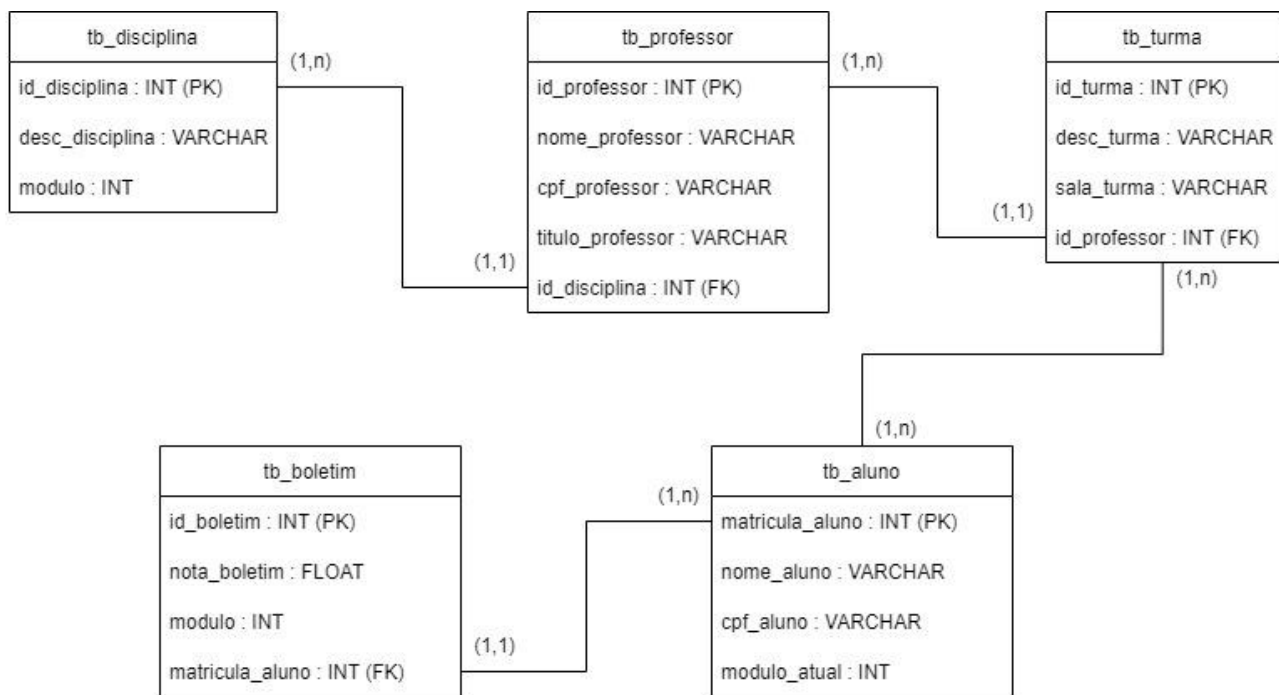
6- Implementação: Utilizar Node.js e a biblioteca Express.js para programação back end, React.js para a programação front end e Postgresql para o banco de dados.

#### 4.4 Diagrama de Caso de Uso





## 4.5 Diagrama de Entidade Relacionamento



(PK) => PRIMARY KEY  
(FK) => FOREIGN KEY

## 5.0 Processo de Criação

O sistema foi desenvolvido utilizando a linguagem de programação JavaScript, com o framework do Front End com React e biblioteca axios, processador CSS foi o SASS, A biblioteca para a navegação nas rotas foi React Router Dom, no Back End foi utilizada a biblioteca Express e arquitetura REST e para o Banco de Dados foi utilizado o PostgreSQL, em uma instância no Site '<https://www.elephantsql.com>', e para a comunicação com o banco a interface PG.

O projeto iniciou-se através da criação do banco de dados, suas tabelas e elementos, após isso o back end começou a ser montado, tendo em mente as diversas funções do sistema, após uma parte do back end estar montada, foi a vez do front end. Enquanto o front end era criado e interligado com o back end, as funções do back end restantes eram criadas, além de funções auxiliares para uma melhor experiência do usuário.

## 6.0 Resultados

Em detrimento do uso das APIs, frameworks e bibliotecas requisitadas, conseguimos obter o resultado esperado acerca do Front end e do back end.

## 7.0 Conclusão

Após a análise dos requisitos que foram pré- estabelecidos na documentação do projeto, Construímos uma aplicação que apresenta alguns crud's, sendo eles: Cadastro de Professor, Cadastro de Aluno, Cadastro de Turma, Cadastro de Disciplina e geração de Boletim). Além disso, mostramos uma tela harmônica utilizando a paleta de cores da logo sem falar que o crud apresenta um conceito visual diferente do padrão. Sendo assim concluímos que o nosso projeto atende a demanda passada, se transformando em um ótimo adicional a homepage da Escolinha do Pato.

## 8.0 Bibliografia

Express(). Express , 2022. Axios. Disponível em URL: <https://expressjs.com/en/5x/api.html> .Acesso em dia: 06/12/2022;

Getting Start. Axios, 2022. Disponível em URL: <https://axios-http.com/docs/intro> .Acesso em dia: 06/12/2022;

Getting Start. Reactjs, 2022. Disponível em URL: <https://reactjs.org/docs/getting-started.html> .Acesso em dia: 06/12/2022;

Getting Start. Nodejs, 2022. Disponível em URL: <https://reactjs.org/docs/getting-started.html> .Acesso em dia: 06/12/2022;

Node.js v19.2.0 documentation. Nodejs, 2022. Disponível em URL: <https://nodejs.org/api/> .Acesso em dia: 06/12/2022;