



**UNIVERSIDADE NOVE DE JULHO
DIRETORIA DOS CURSOS DE INFORMÁTICA**

**AMANDA CARVALHO GOES - RA 2222101399
ANDRÉ LUIZ FERREIRA NASCIMENTO - RA 2222104390
GUILHERME DA SILVA RODRIGUES AMORIM - RA 2222103622
HUGO DA SILVA DUARTE - RA 2222101723
IGOR CAVALCANTE OLIVEIRA - RA 2222102619
JOÃO PEDRO MENDES GOMES - RA 92012790
KAROLINE DE OLIVEIRA CARVALHO - RA 920204856
LUCIANO BRAGA MONTEIRO - RA 2222106869
MARCOS VINICIUS DANTAS DE OLIVEIRA - RA 2222101758
RAFAEL SILVA E SILVA - RA 2222102004
WANDERSON FERREIRA DE MATOS - RA 2222102750**

**PROJETO PRÁTICO EM PROGRAMAÇÃO
POINT G8T**

**SÃO PAULO
2022**

AMANDA CARVALHO GOES - RA 2222101399
ANDRÉ LUIZ FERREIRA NASCIMENTO - RA 2222104390
GUILHERME DA SILVA RODRIGUES AMORIM - RA 2222103622
HUGO DA SILVA DUARTE - RA 2222101723
IGOR CAVALCANTE OLIVEIRA - RA 2222102619
JOÃO PEDRO MENDES GOMES - RA 92012790
KAROLINE DE OLIVEIRA CARVALHO - RA 920204856
LUCIANO BRAGA MONTEIRO - RA 2222106869
MARCOS VINICIUS DANTAS DE OLIVEIRA - RA 2222101758
RAFAEL SILVA E SILVA - RA 2222102004
WANDERSON FERREIRA DE MATOS - RA 2222102750

**PROJETO PRÁTICO EM PROGRAMAÇÃO
POINT G8T**

Trabalho apresentado à Universidade Nove de Julho,
UNINOVE, em cumprimento parcial às exigências da
disciplina de Planejamento em Informática, sob orientação do
Prof. **Márcio Klein**.

**SÃO PAULO
2022**

RESUMO

O documento a seguir apresenta o formato de criação de uma API baseado no modelo PDV, documentando etapas cruciais para o desenvolvimento de *software*, como: criação de classes e lógica de programação por trás do sistema, modelagem de dados e descrição de entidades e seus atributos, métodos de verificação de segurança para autenticação e autorização, dentre outros.

Palavras-chave: GERENCIAMENTO, CRUD, PDV, MVC.

SUMÁRIO

OBJETIVOS	1
METODOLOGIA	2
CRONOGRAMA	3
PLANEJAMENTO DO SITE	4
TELAS DO WEBSITE	11
MODELAGEM DE DADOS	13
LÓGICA DE PROGRAMAÇÃO	16
SEGURANÇA DA INFORMAÇÃO	17
REFERÊNCIAS	18

1. OBJETIVOS

Projeto de uma aplicação *web* baseada no formato de sistemas PDV (Ponto De Venda), com a finalidade de auxiliar seus usuários no gerenciamento de estoque e produtos dos seus negócios. A *API* é simples e prática, possuindo atividades de login e cadastro de usuários e manipulação de objetos usando o conceito *CRUD* (*Create, Read, Update e Delete*) juntamente com a arquitetura de *software MVC* (*Model, View e Controller*) para estruturar o código e facilitar tanto no seu desenvolvimento, quanto na manutenção e adição de novas funcionalidades (PÚBLIO, Angelo).

O projeto visa melhorar a experiência de usuários que utilizam de métodos antiquados e ultrapassados para organização de negócio, otimizando o tempo de tarefas repetitivas e importantes.

2. METODOLOGIA

As tarefas de desenvolvimento do projeto foram separadas em 3 partes, sendo documentação, desenvolvimento e operação. Cada parte possui suas etapas para serem desenvolvidas, mas toda a equipe irá trabalhar em cada parte do projeto (todos os integrantes desenvolvendo a documentação, o desenvolvimento e a operação). Haverá reuniões para todos tirarem dúvidas entre si e pesquisarem juntos melhores soluções para os problemas.

Para melhorar na organização do desenvolvimento, a equipe seguirá as seguintes etapas:

- Análise de requisitos
- Documentação
 - Documentação formal do software
 - Diagrama de caso
 - Diagrama de fluxo de atividades
 - Diagrama de classes
 - Modelagem do banco de dados
 - Protótipo das telas do sistema
- Desenvolvimento
 - Back-end da aplicação (CRUD)
 - Design front-end da aplicação (telas)
 - Otimizar codificação de segurança
 - Realizar testes manuais/aplicar testes unitários
- Operações
 - Deploy (hospedagem) da aplicação
- Entregar o projeto

3. CRONOGRAMA

A previsão para a finalização da documentação e desenvolvimento do projeto é de 1 semana para cada parte, totalizando 2 semanas. Já para operações, em 1 dia será realizada a hospedagem.

As tabelas a seguir representam as etapas a serem cumpridas e os integrantes que irão realizá-las dentro de 1 semana:

DOCUMENTAÇÃO	
Etapas	Integrantes
Documentação formal do software	João P.
Diagrama de caso	Guilherme, Dinho, Hugo
Diagrama de fluxo de atividades	Rafael silva, Marcos Vinicius
Diagrama de classes	Igor C
Modelagem do banco de dados	Amanda, Luciano, André
Protótipo das telas do sistema	Karol

DESENVOLVIMENTO	
Etapas	Integrantes
Backend da aplicação (CRUD)	João P, Igor C, Luciano, Hugo Duarte
Design front-end da aplicação (telas)	Amanda, Karol, Rafael, Guilherme, André
Otimizar codificação de segurança	João P,
Realizar testes manuais (ou aplicar testes unitários)	Marcos, Dinho

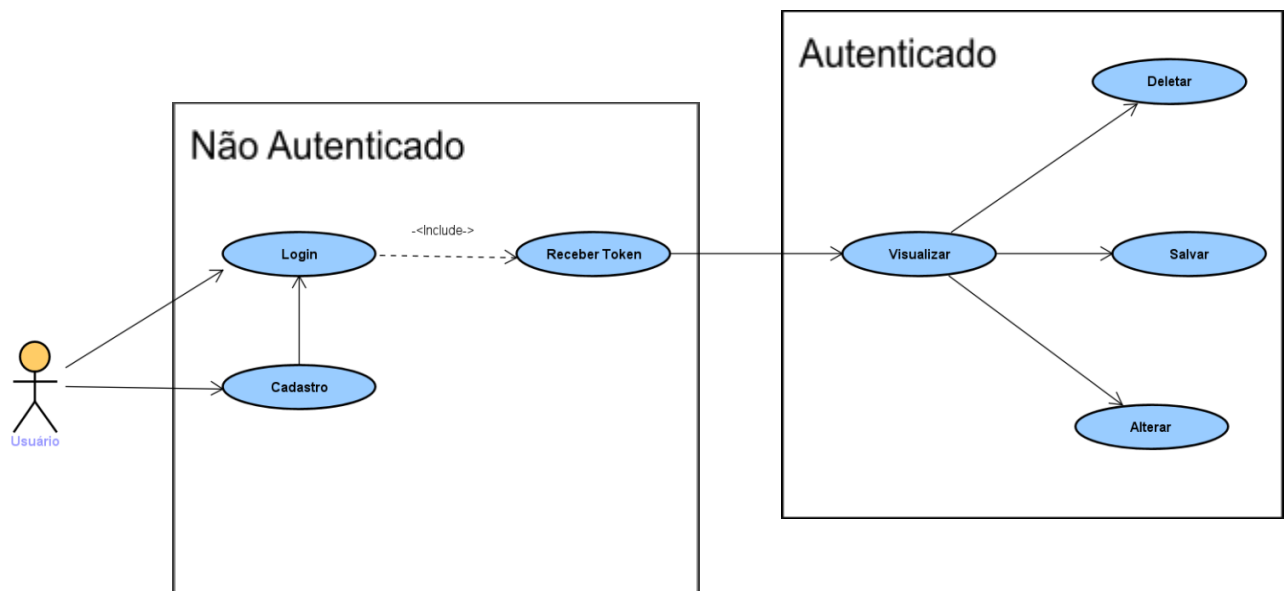
OPERAÇÃO	
Etapas	Integrantes
Hospedagem	TODOS

4. PLANEJAMENTO DO SITE

Ao acessar o *link* da documentação da aplicação (<https://pointg8t.herokuapp.com/api/swagger-ui/index.html?configUrl=/api/v3/api-docs/swagger-config#/>), o usuário recebe como resposta da requisição uma página feita pelo *framework* *Swagger* que pode ser acessada por qualquer usuário, sem precisar estar logado (com o *token* armazenado). Caso queira acessar o sistema responsável pela manipulação do estoque, o usuário deve realizar o login e, caso não possua suas credenciais cadastradas, deverá se cadastrar, recebendo assim, um *token* para verificar autenticação e autorização dentro do sistema.

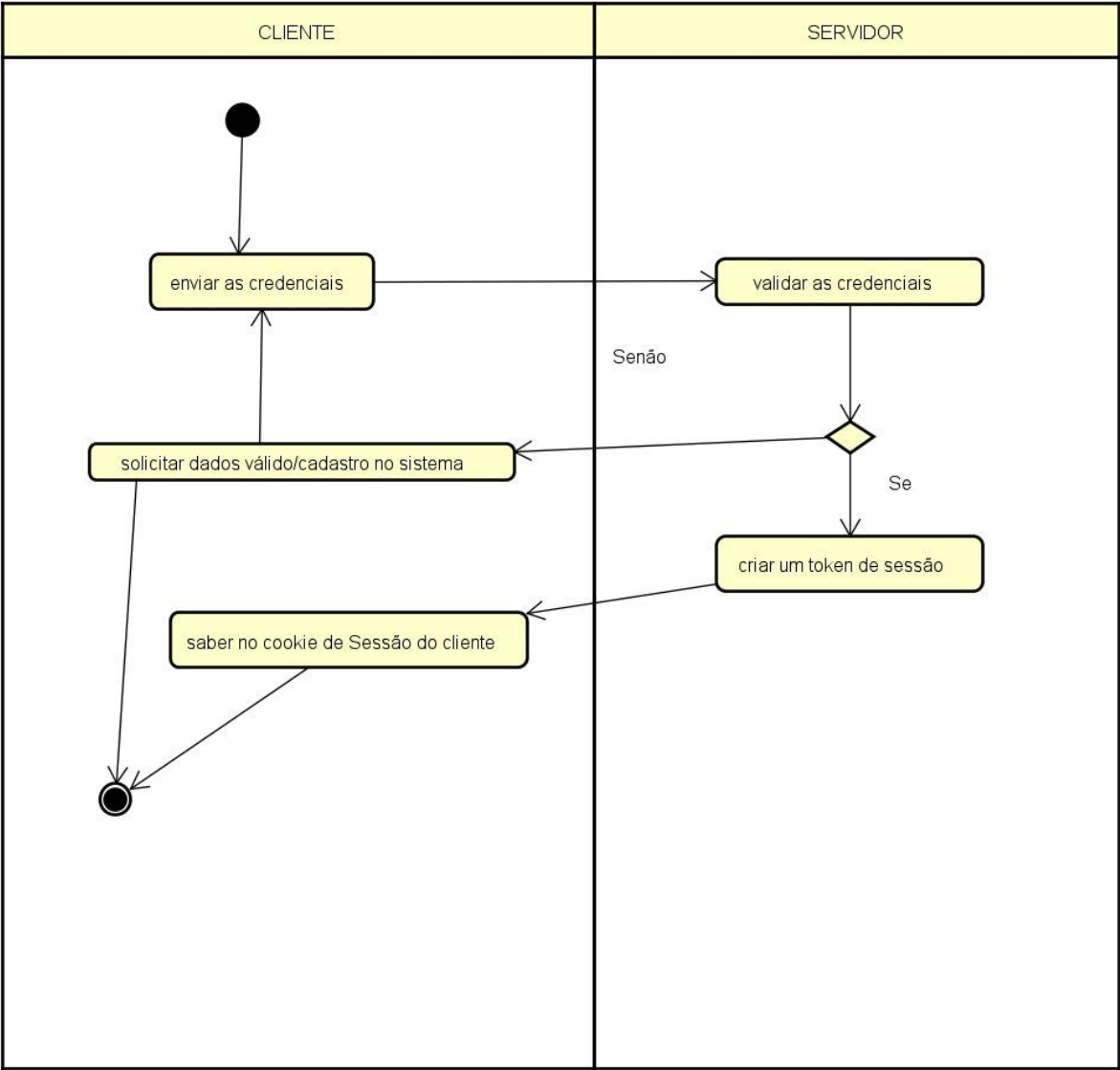
As imagens a seguir representam o fluxo de atividades e caso de uso do site:

Figura 1 - Caso de uso Point G8T.



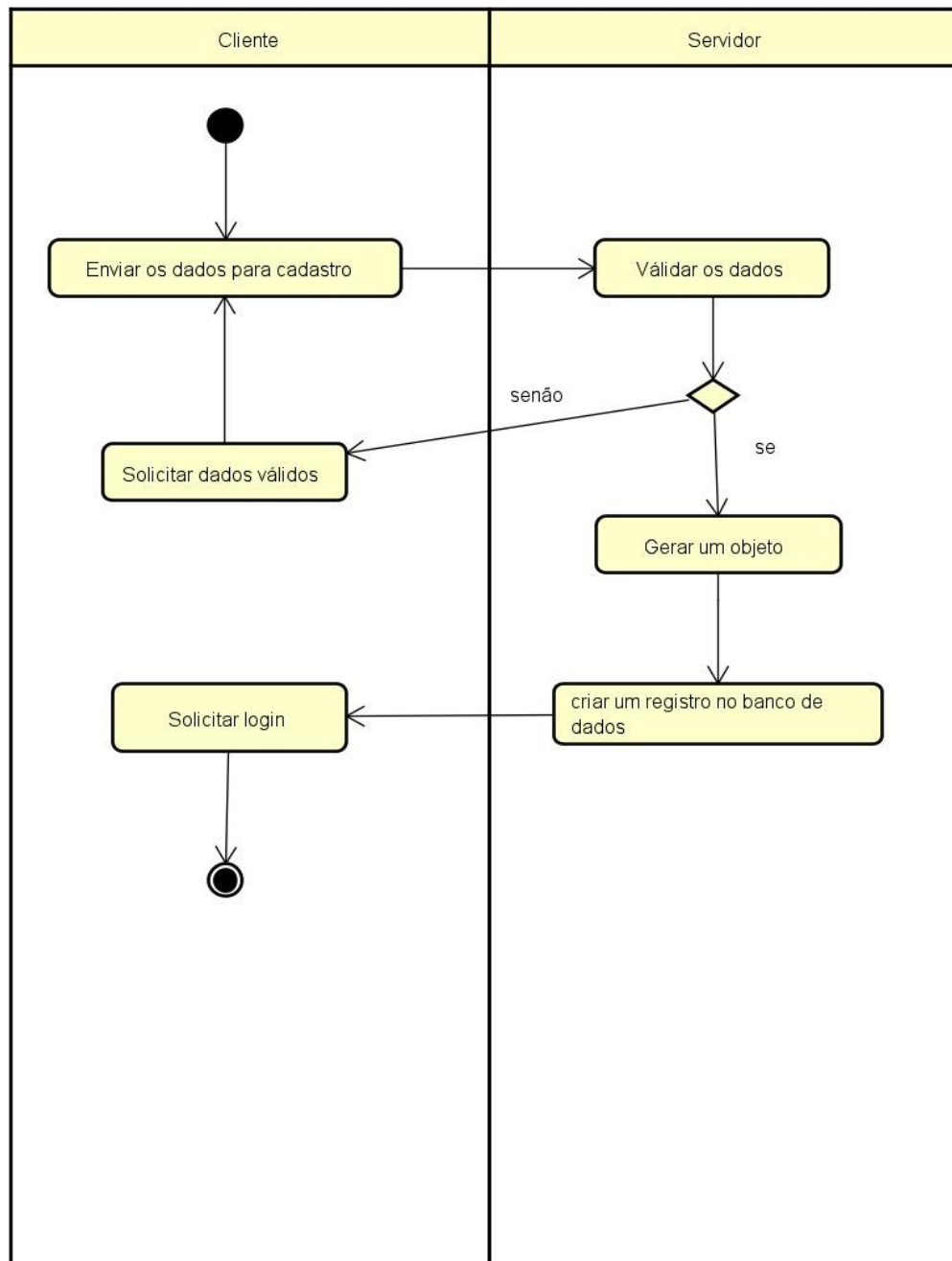
Fonte: O Autor, 2022.

Figura 2 - Fluxo de Atividade Login Point G8T.



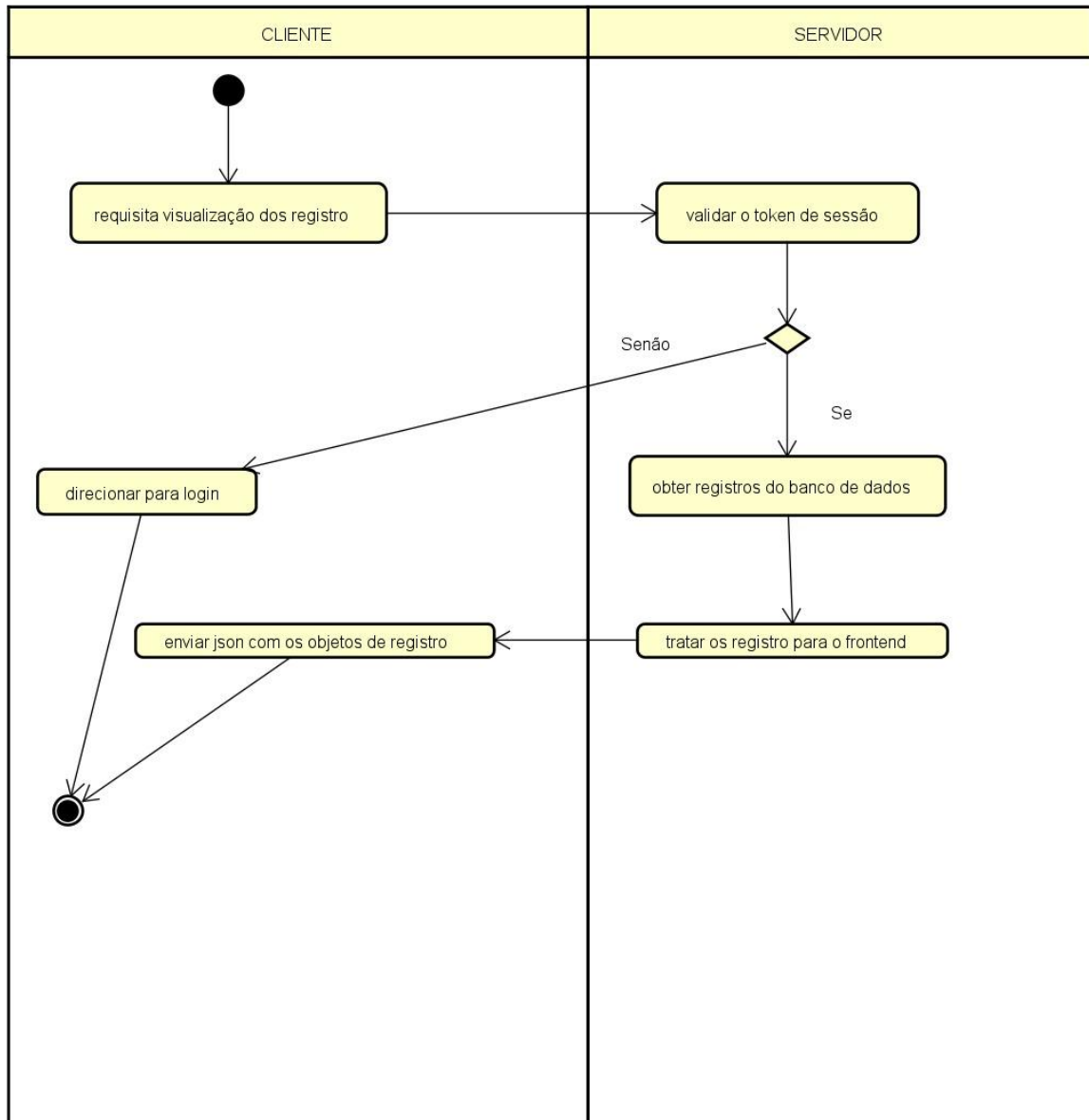
Fonte: O Autor, 2022.

Figura 3 - Fluxo de Atividade Cadastro Point G8T.



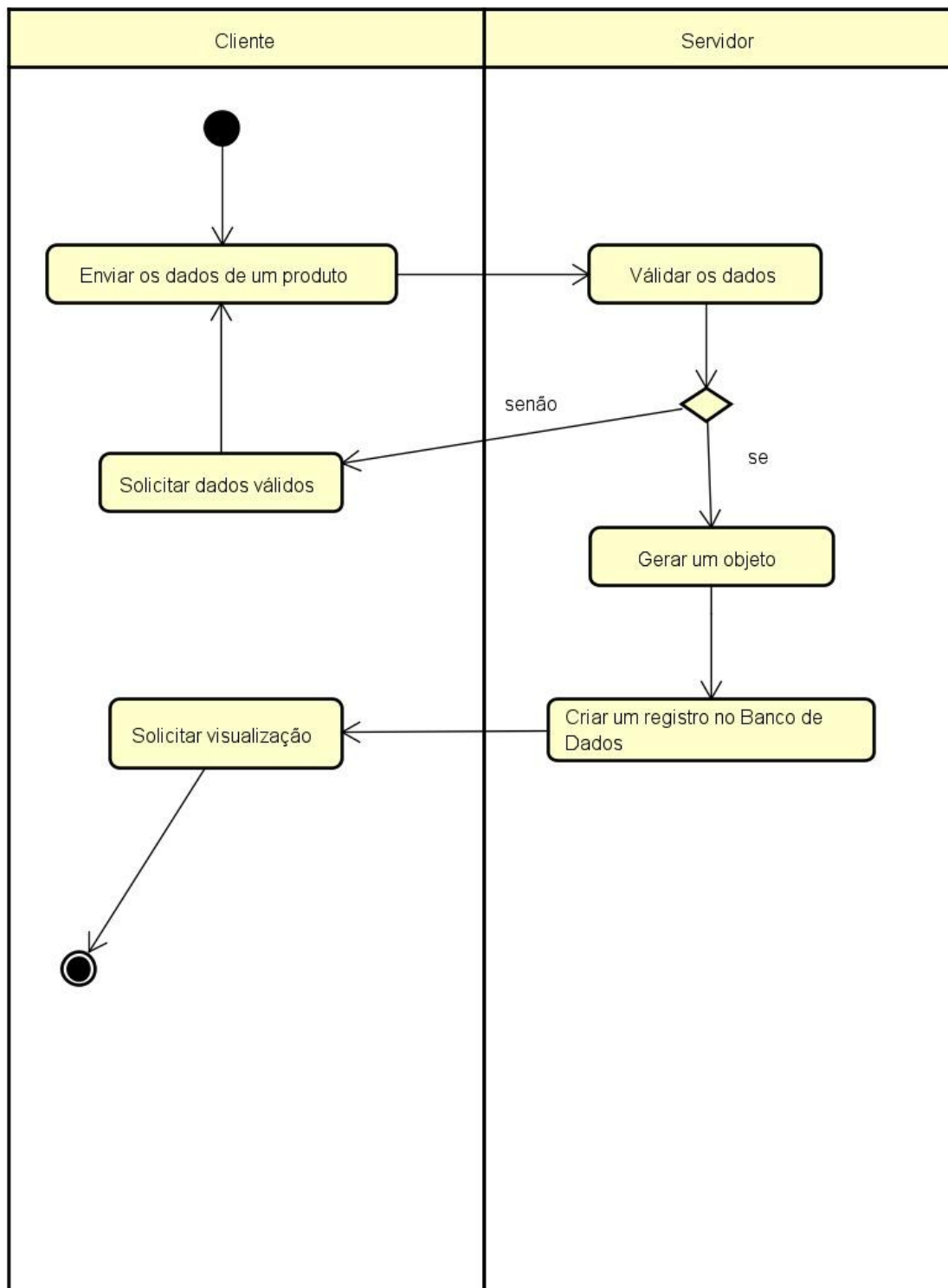
Fonte: O Autor, 2022.

Figura 4 - Fluxo de Atividade Visualizar Point G8T.



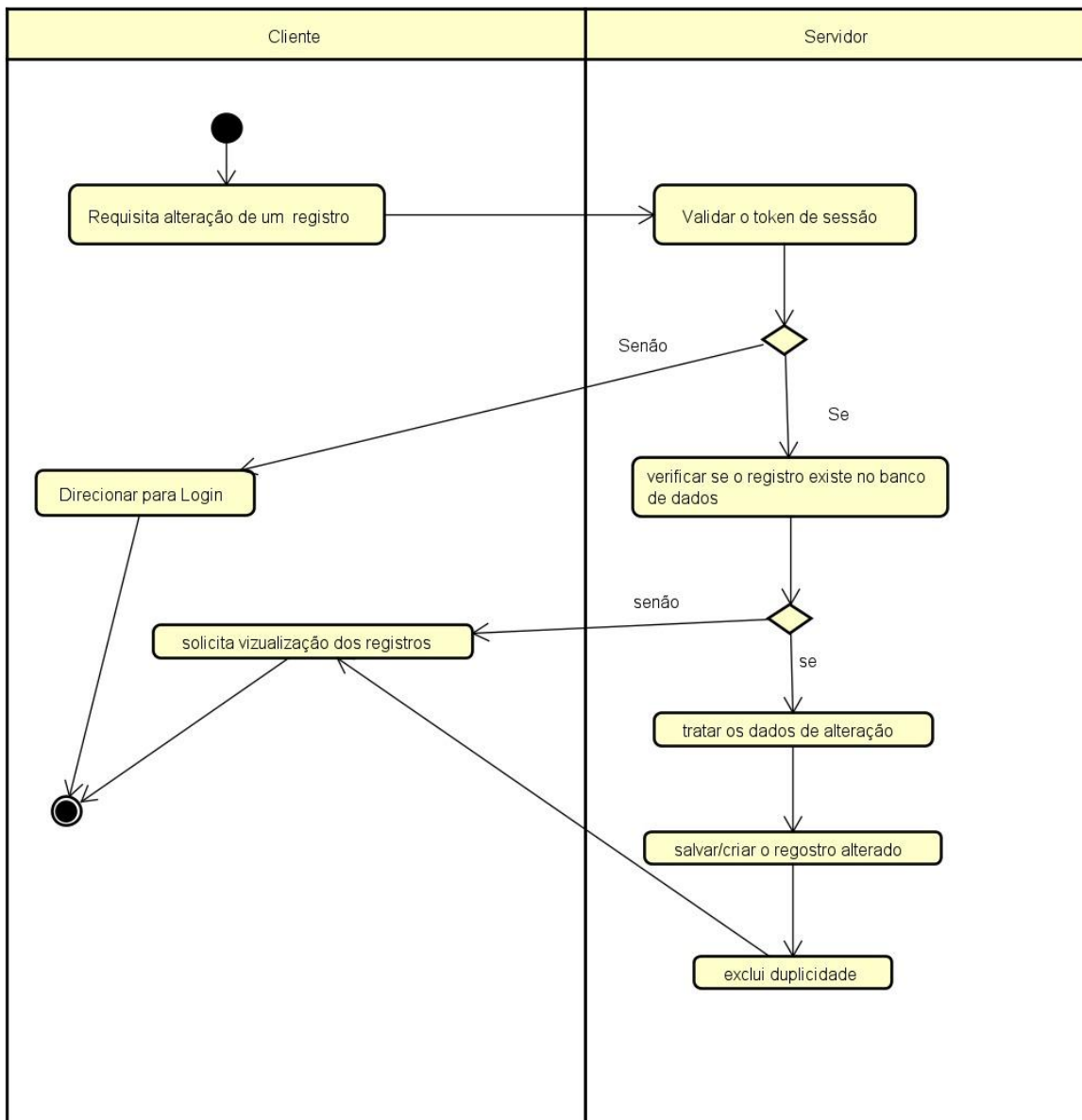
Fonte: O Autor, 2022.

Figura 5 - Fluxo de Atividade Salvar Point G8T.



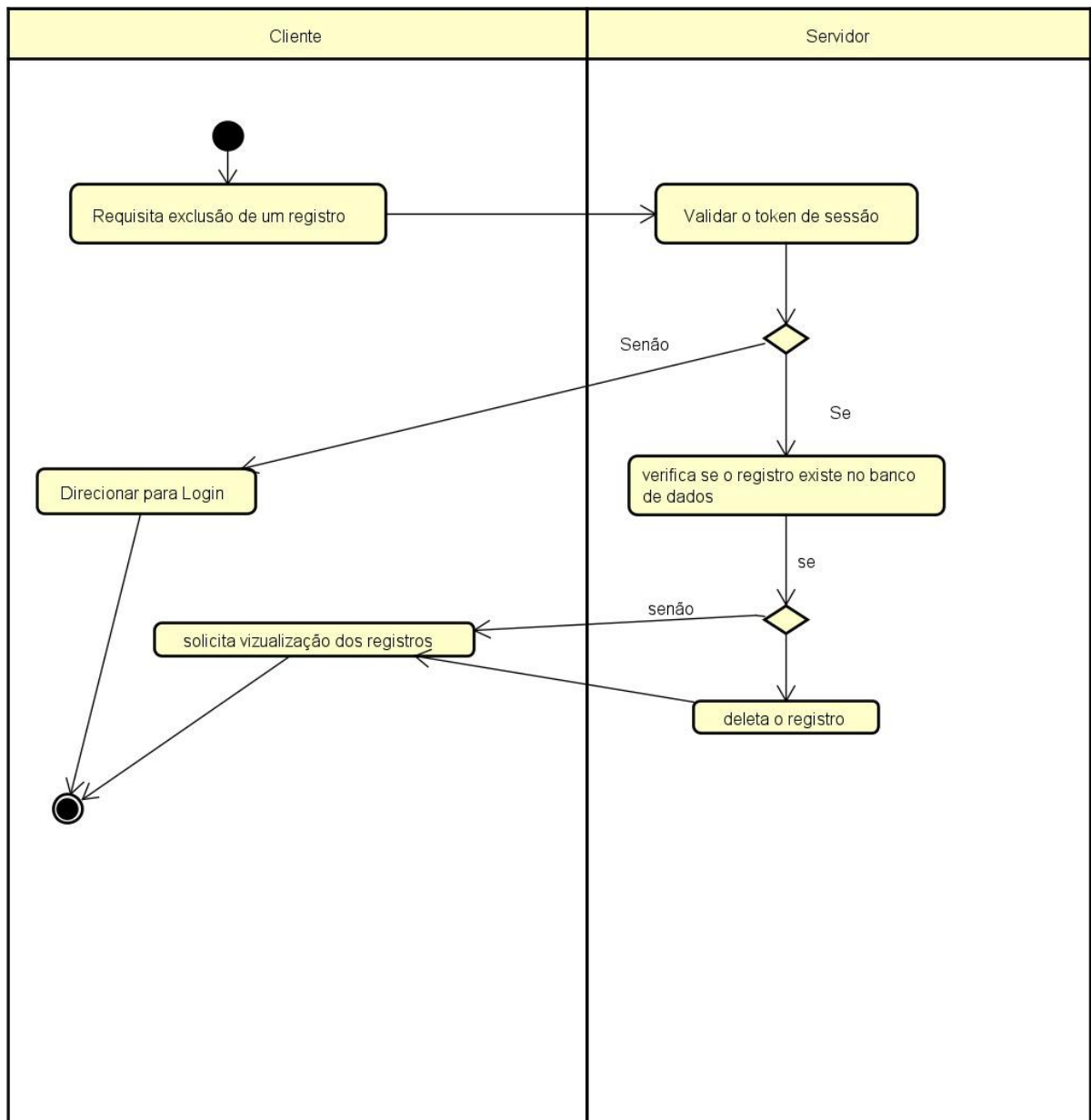
Fonte: O Autor, 2022.

Figura 6 - Fluxo de Atividade Alterar Point G8T.



Fonte: O Autor, 2022.

Figura 7 - Fluxo de Atividade Deletar Point G8T.



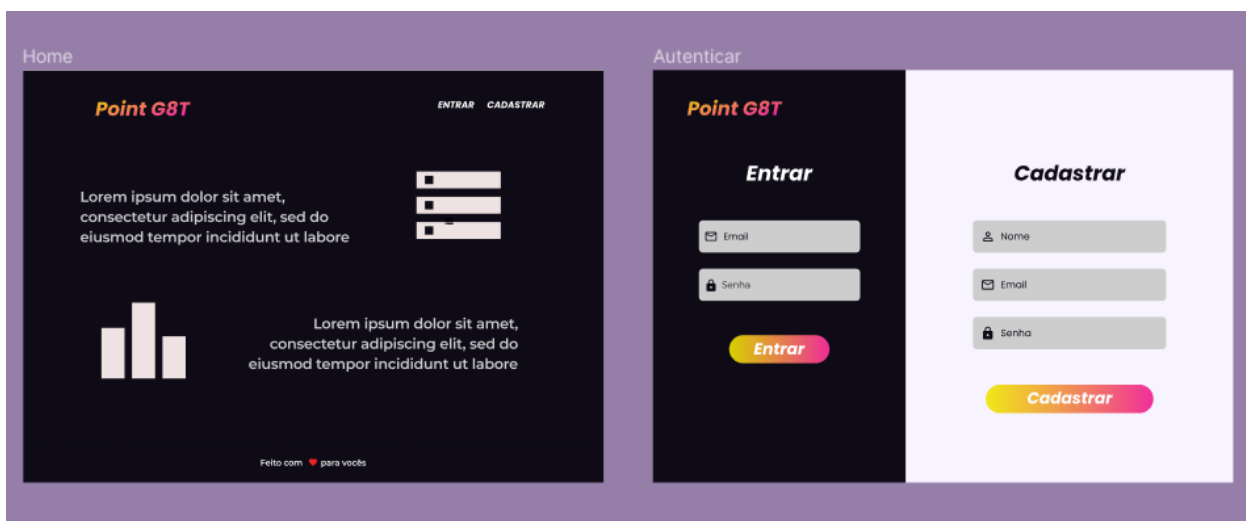
Fonte: O Autor, 2022.

Imagens criadas utilizando o software Astah UML.

5. TELAS DO WEBSITE

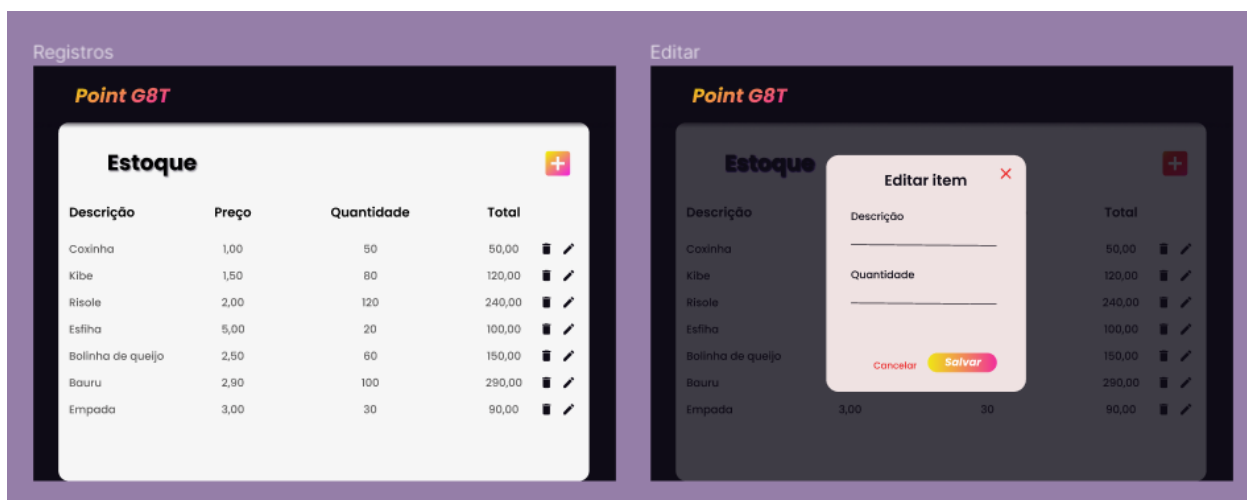
As imagens a seguir representam a prototipação das telas presentes no sistema:

Figura 8 - Telas: Inicial e Autenticação.



Fonte: O Autor, 2022.

Figura 9 - Telas: Registro e Edição.



Fonte: O Autor, 2022.

Imagens criadas pelo autor.

Disponível em:

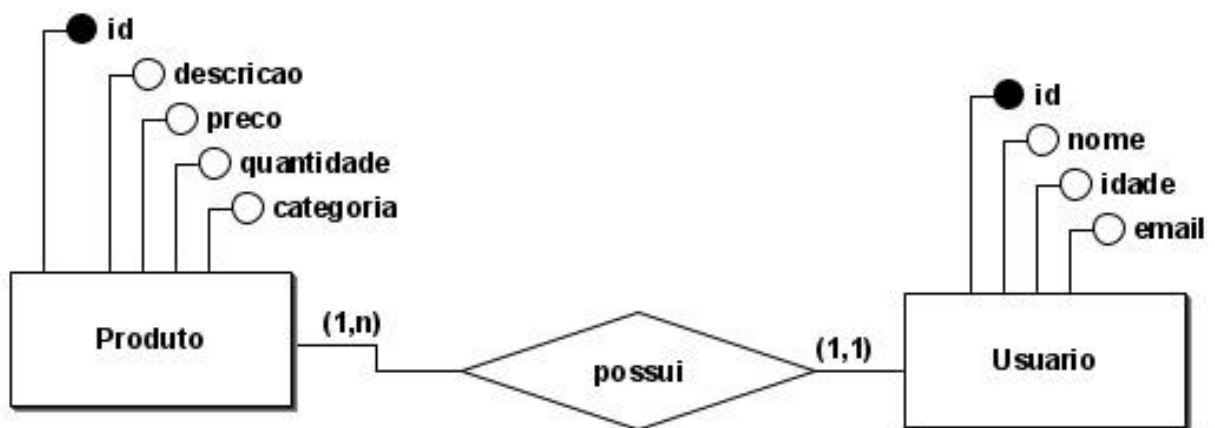
<<https://www.figma.com/file/yiCp0ASbryeyl3KFmTMcFn/POINT-G8T?node-id=0%3A1>.

6. MODELAGEM DE DADOS

As entidades identificadas foram: Usuario e Produto. A entidade Usuario permite que o sistema manipule objetos responsáveis por representar os usuários que estejam acessando o sistema, armazenando o seu e-mail, nome, dentre outros atributos. Já no caso da entidade Produto, é para manipular os objetos principais da aplicação, tornando possível a visualização, alteração, exclusão e criação desses objetos, tendo uma relação de 1 para N (onde um produto só poderá pertencer a 1 usuario e 1 usuario poderá pertencer a N produtos).

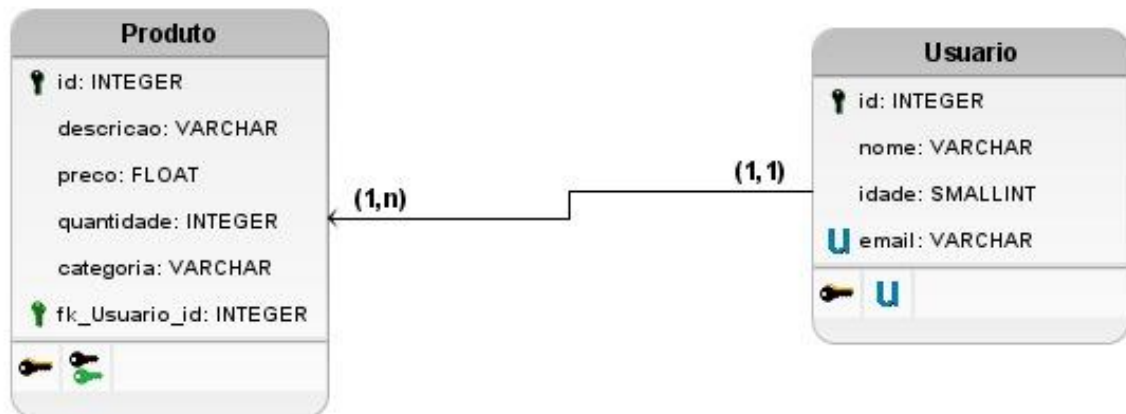
As imagens a seguir representam os modelos conceitual e lógico da aplicação:

Figura 10 - Modelagem de Banco de Dados Conceitual.



Fonte: O Autor, 2022.

Figura 11 - Modelagem de Banco de Dados Lógica.



Fonte: O Autor, 2022.

Imagens criadas utilizando o software BrModelo.

7. LÓGICA DE PROGRAMAÇÃO

A lógica foi pensada para conseguir realizar as quatro operações do *CRUD* de maneira simples, usando a arquitetura *MVC* para auxiliar, por se tratar de uma excelente arquitetura, que otimiza os processos internos do sistema (ZUCHER, Vitor, 2020). Há as classes *Usuario* e *Produto* para possibilitar a manipulação desses objetos através do *back-end* da aplicação, onde alguns critérios deverão ser atendidos para possibilitar a ação de requisição que foi solicitada pelo usuário, tais como, verificar se o *token* do usuário está presente na sessão, se o corpo da requisição é compatível com o método de conversão para o objeto que está a ser criado ou manipulado, se o objeto a ser deletado ou alterado existe de fato no banco de dados, dentre outras verificações.

Os diagramas a seguir demonstram as classes a serem criadas na aplicação:

Figura 12 - Diagrama de Classe.



Fonte: O Autor, 2022.

Imagem criada utilizando o software Astah UML.

8. SEGURANÇA DA INFORMAÇÃO

Até o presente momento, a aplicação pode ser acessada por qualquer pessoa, mas somente os dados que forem criados por ela serão possíveis de visualizar e manipular! Pois a medida de segurança de autorização a acesso dos dados está fortemente ligada com o *token* que será gerado pelo sistema para cada usuário, armazenando (de maneira encriptada) algumas informações que serão checadas para validar cada requisição que o usuário fizer.

Mesmo adotando medidas de segurança eficazes, POINT G8T não se responsabilizará por vazamento de dados nem adotamos qualquer postura referente a LPDG, pois este é um projeto de fins acadêmicos, para obter conhecimento e ganhar habilidades de programação prática.

REFERÊNCIAS

PÚBLIO, Angelo (2022), "*CRUD: o que é este conceito no Desenvolvimento de Sistemas.*" Blog Angelo Públio, acesso em 03/2022: <https://angelopublico.com.br/blog/crud>.

ZUCHER, Vitor (2020), "*O que é padrão MVC? Entenda arquitetura de softwares!*" Lewagon, acesso em 03/2022: <https://www.lewagon.com/pt-BR/blog/o-que-e-padrao-mvc>.