ABEX III - PROJETO DE SOFTWARE- 1030784

2023-01

| **Professor(a)(es)** | RADAMÉS PEREIRA  VIVIANE DUARTE BONFIM |
| --- | --- |

Atributos de uma boa especificação de Projeto de Software:

* Clareza
* Não Ambígua
* Completa
* Simples
* Bem escrita

**UNIVERSIDADE COMUNITÁRIA REGIONAL DE CHAPECÓ - UNOCHAPECÓ**

**ÁREA DE CIÊNCIAS EXATAS E AMBIENTAIS**

**SISTEMAS DE INFORMAÇÃO/CIÊNCIA DA COMPUTAÇÃO**

Site para Abex de Veterinária

Equipe: **Affonso Henrique Ely**

**Ivanilson Schwingel**

**Julio Cesar Murmann Prestes**

**Julia Letícia Teston**

**Gabriel Chitolina Thomé**

**Naieli Andreia Loeblein**

Introdução

## Resumo do Projeto

Descreve-se aqui o sistema a ser desenvolvido.

Site para postagem dos alunos da Abex de veterinária, no qual os alunos irão solicitar uma postagem e o(a) professor(a) terá a opção de aprovar, editar ou recusar o trabalho. Quando aprovado, será disponibilizado para todos os usuários que acessarem o site poderem ler, curtir e/ou compartilhar;

## Plataforma de desenvolvimento

Descreve-se aqui uma primeira visão das tecnologias para desenvolvimento do projeto de software.

Exemplo: Tecnologias, Sistema de edição; Maquina virtual…

Front-end: React

Back-end: Node

## Plataforma de operação

Descreve-se aqui uma primeira visão das tecnologias para operacionalização.

Exemplo: Onde sistema será executado: Cliente-servidor…

O sistema será executado em um servidor na nuvem. O cliente o utilizará na

WEB, podendo utilizá-lo de diversos navegadores.

## Definições e siglas

Não se aplicará.

## Perspectiva do produto

Este projeto tem como foco auxiliar no controle de informações do curso de veterinária, da grade de abex, onde alunos podem postar seus trabalhos para serem avaliados e receberem um retorno sobre seus projetos. E este local de postagem também servirá como um repositório de informações para futuras turmas da Abex de Veterinária.

### Modos de operação

### Identificam-se aqui os modos requeridos de operação, tais como: Back-End-Front-End, Móvel, Stand-Alone, …

Back-end e Front-end

### Requisitos de adaptação ao ambiente

Não há nenhum necessário.

## Funções do produto

Identificam-se aqui as principais funções que o produto desempenhará, descrevendo de forma sintética o objetivo de cada uma.

Exemplo: Caso de uso: função que será realizada.

Visitante: Precisa de caso de uso?

Aluno(a):

* Postar o trabalho;
* Fazer o login com a conta da Unochapecó;

Professor(a):

* Aceitar, recusar ou editar o trabalho do aluno;
* Faz o login com a conta da unochapecó;

Usuário qualquer:

* Visualizar, curtir e/ou compartilhar as postagens dos alunos;

## Características dos usuários

Descrevem-se aqui as principais características dos grupos de usuários esperados para o produto, tais como cargo ou função, permissão de acesso, frequência de uso, nível de instrução, proficiência no processo de negócio e proficiência em informática.

Exemplos: usuários do nível gerencial; operacionais;

Professor(a), aluno(a) ou um usuário qualquer(visitante);

## Restrições

Descrevem-se aqui aspectos técnicos e gerenciais que possam limitar as opções dos desenvolvedores, tais como restrições legais.

Exemplo: sistemas de vendas de trabalho, sistema fiscal de vendas.  
Normas e padrões da UNOCHAPECÓ ou restrições/obrigações de design de website.

## Hipóteses de trabalho

Descrevem-se aqui fatores que não são restrições limitativas do desempenho, como na subseção anterior, mas fatores cuja alteração requer modificações na ER, como, por exemplo, versão a ser utilizada do ambiente operacional ou plataforma de desenvolvimento.

Exemplos: Linguagem de programação que depende de apis e hardware...

Fatores que podem impactar na tecnologia utilizada para desenvolver.

Não se aplicará

# Requisitos específicos

## Interfaces externas

### Visão geral

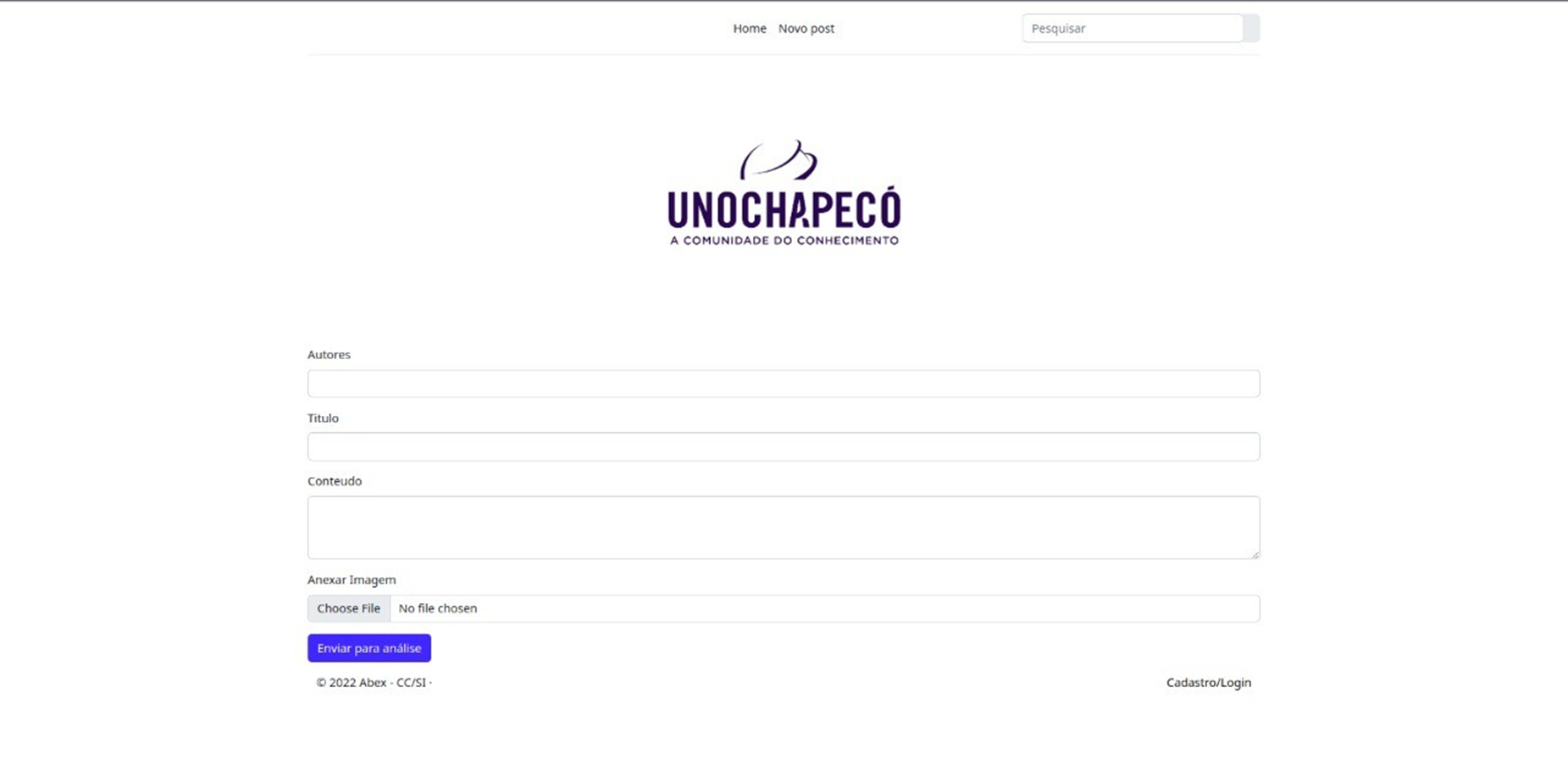
Descreve-se aqui, de forma detalhada, todas as entradas e saídas do produto.

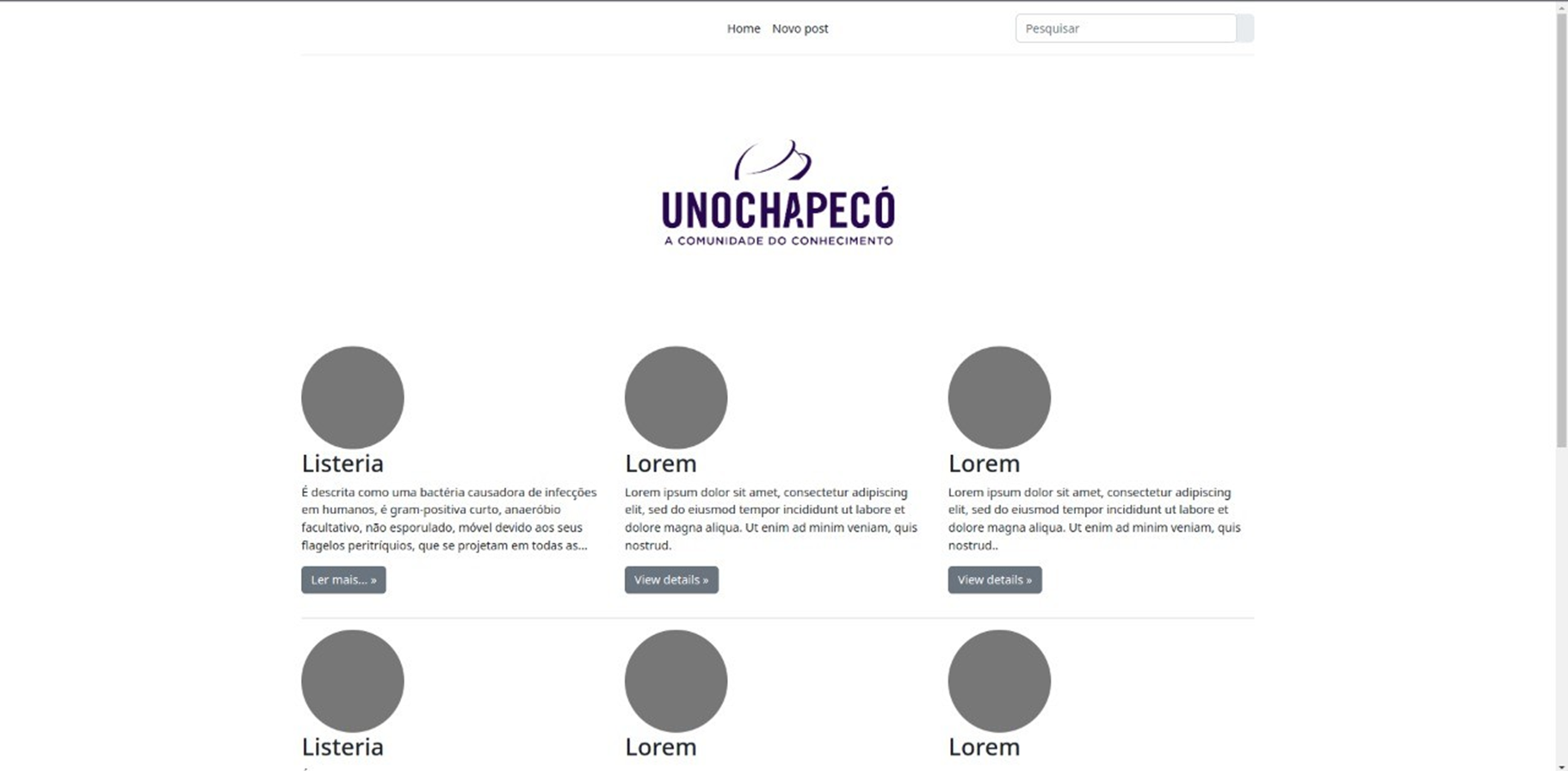
Aluno irá solicitar postagem de trabalho e o(a) professor(a) irá aprovar, editar ou recusar a postagem.

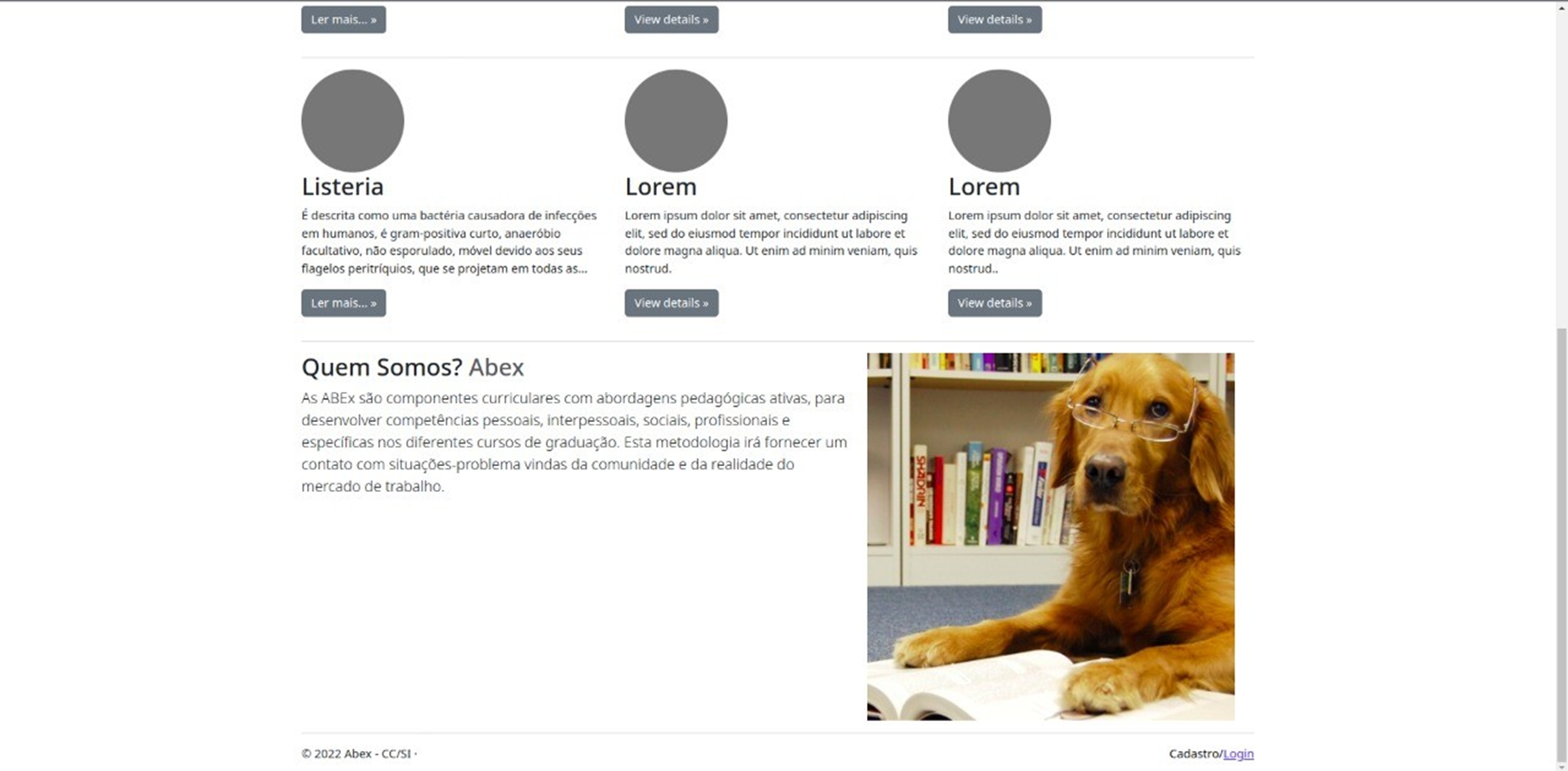
### Requisitos para interfaces gráficas de usuário

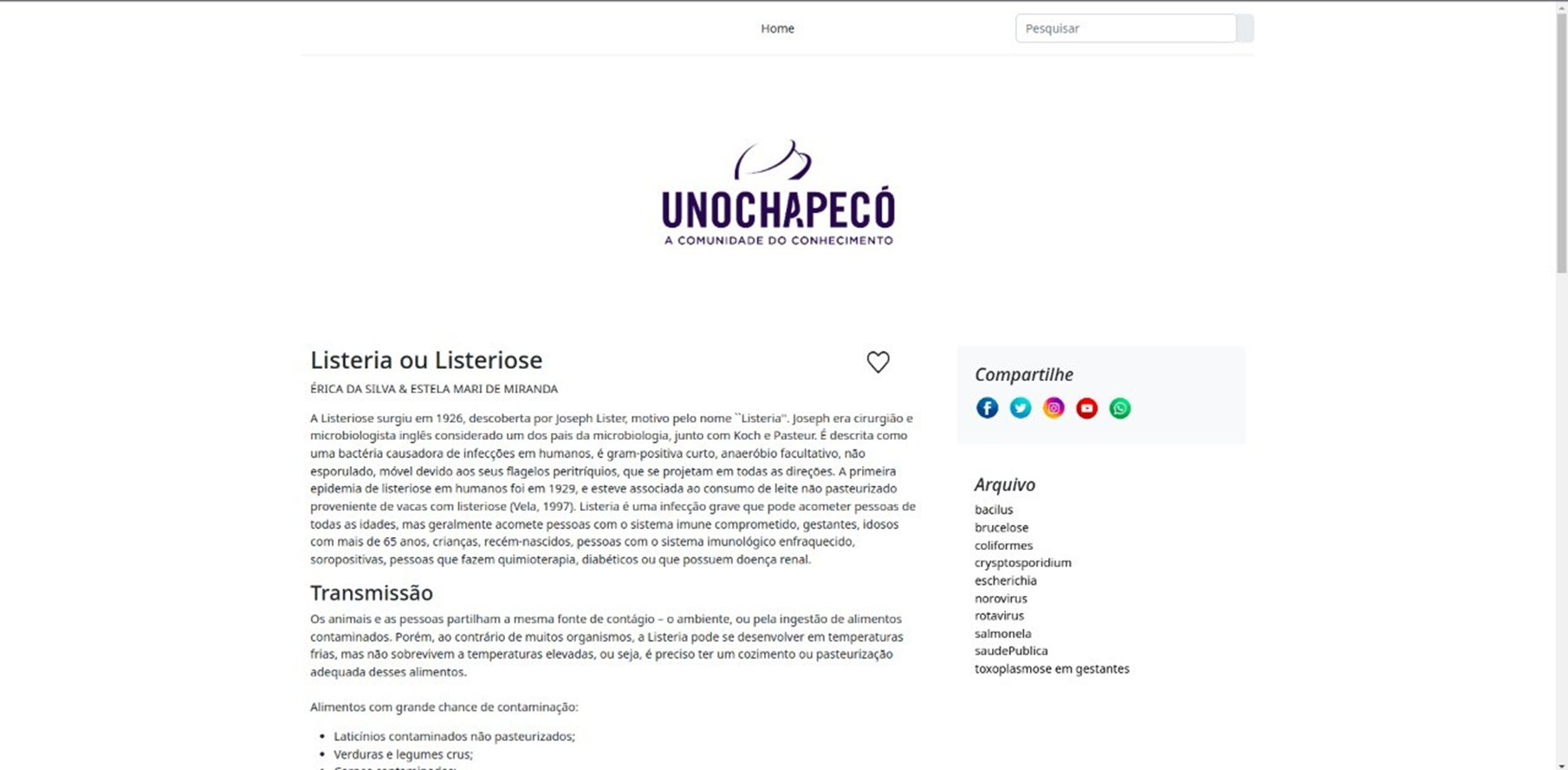
Sugere-se, no caso de interfaces gráficas, a inclusão dos seguintes elementos:

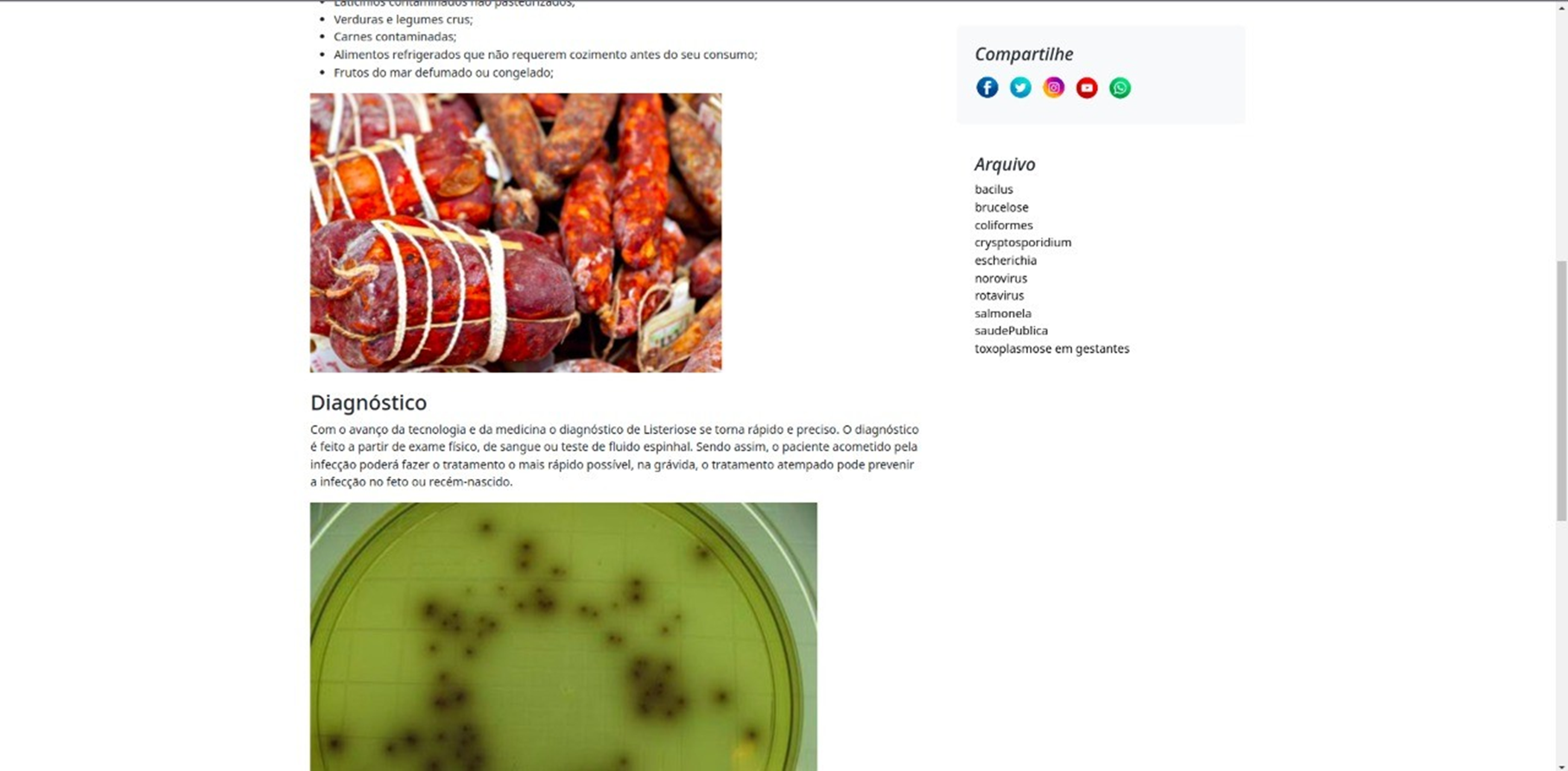
* Um esboço do layout gráfico sugerido para a interface;
* Uma descrição dos relacionamentos com outras interfaces;
* Um diagrama de estados/atividades, caso necessário para melhor entender-se o comportamento requerido da interface;
* Uma lista dos campos de dados da interface;
* Uma lista dos comandos da interface;
* BPM;

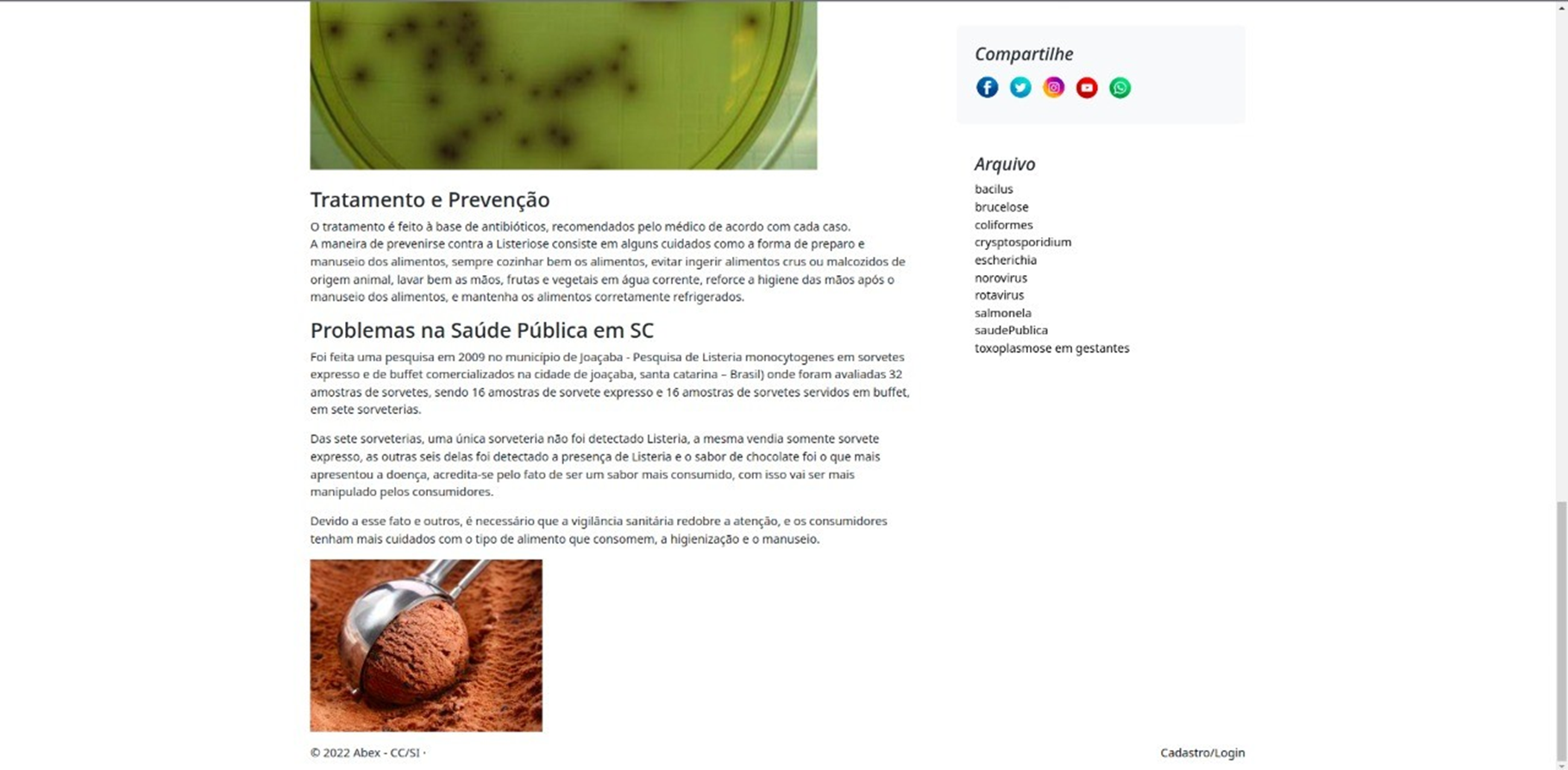


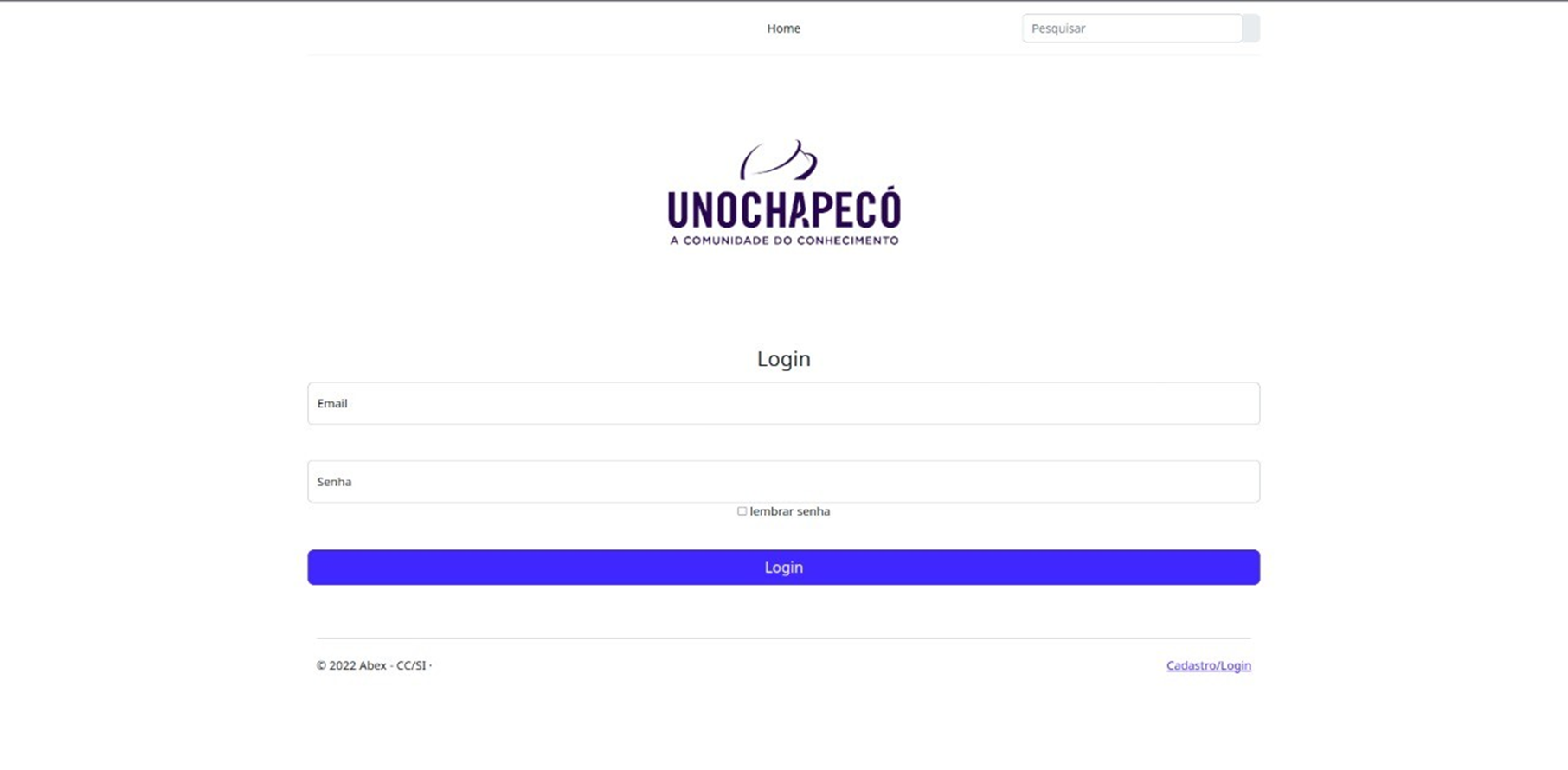


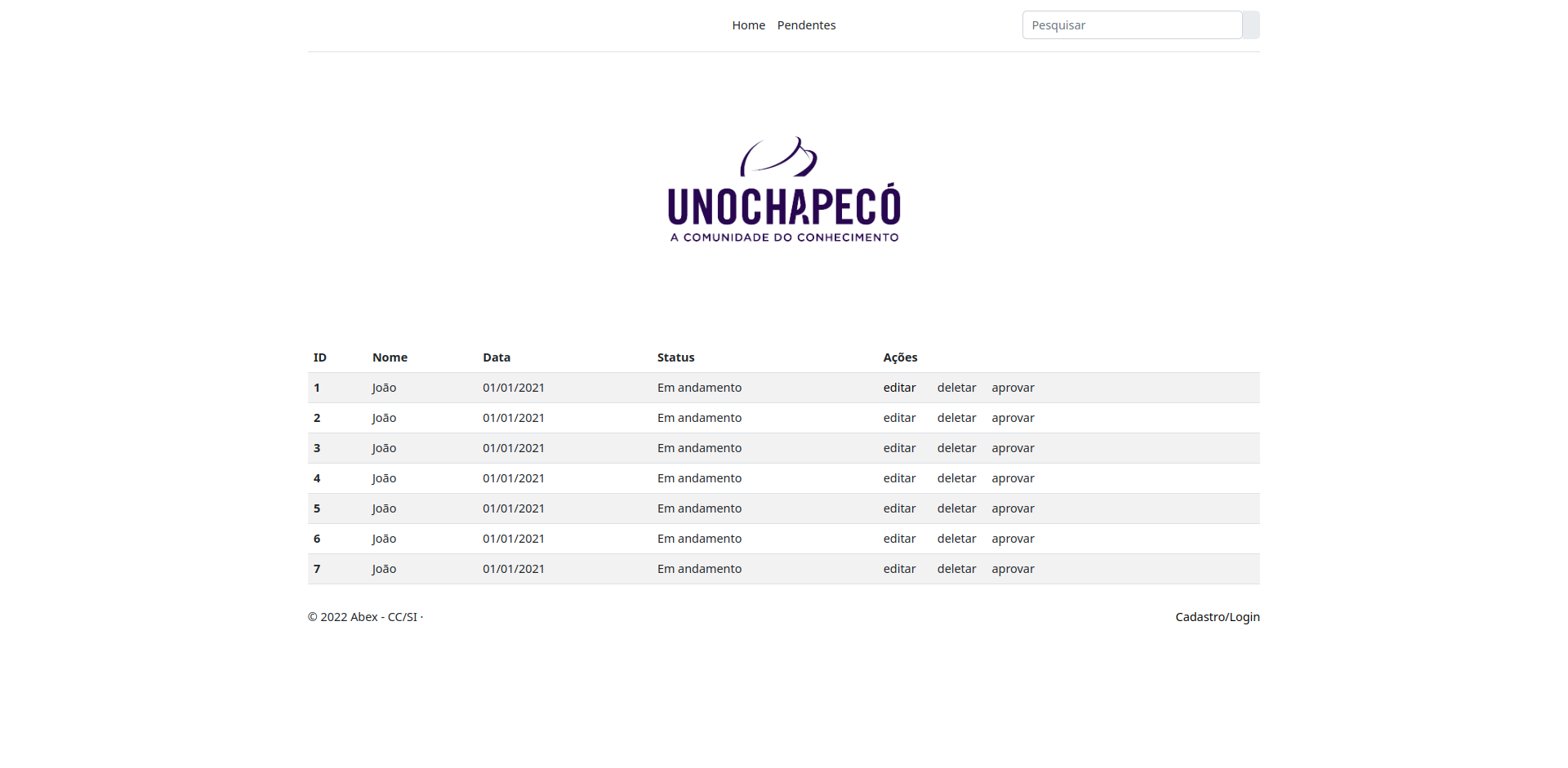






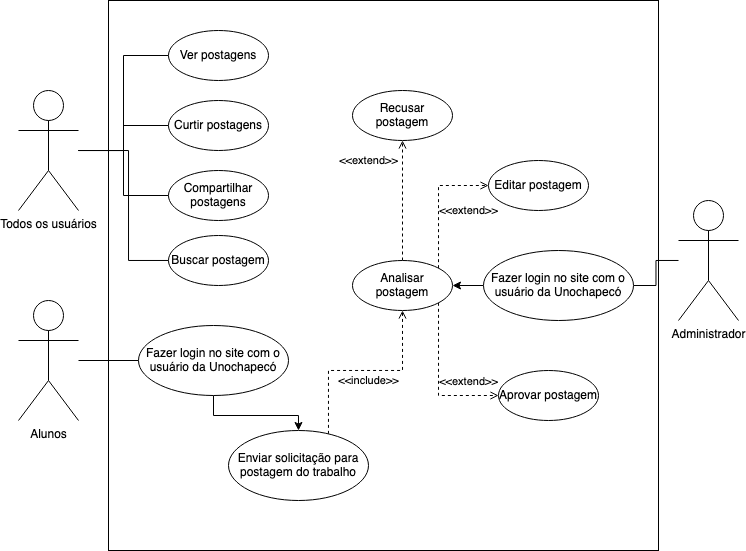






## Requisitos funcionais

### Diagramas de casos de uso

****

### Fluxos dos casos de uso

| **RF1. “Ver Postagens”** | | |
| --- | --- | --- |
| **Descrição:** Ver todas as postagens. O sistema deve exibir as postagens já aprovadas pelo administrador. | | |
| **Atores:** Todos os usuários | | |
| **Pré-Condições:** Conexão com internet e acesso a aba de postagens | | |
| **Pós-Condições:** Postagens listadas com sucesso. | | |
| **Fluxo básico:**   1. O usuário acessa a página e todas as postagens serão exibidas. | | |
| **Requisito não funcional** | Restrição | Categoria |
| NF 1.1 - “Visibilidade das postagens” | Usuário qualquer poderá ver as postagens aprovadas, tanto visitante, quanto aluno ou administrador | Segurança |
| NF 1.2 - “Interface gráfica” | Uso de Design responsivo nas interfaces gráficas. | Usabilidade |
| NF 1.3 - “Compatibilidade do site.” | Compatibilidade com windows, linux, mac os android e ios. | Compatibilidade |

| **RF2. “Curtir Postagens”** | | |
| --- | --- | --- |
| **Descrição:** O usuário terá a opção de curtir as postagens. | | |
| **Atores:** Todos os usuários | | |
| **Pré-Condições:** Conexão com internet e acesso a aba de postagens | | |
| **Pós-Condições:** Postagens curtida com sucesso. | | |
| **Fluxo básico:**   1. O usuário acessa uma postagem e clica no botão de curtir. | | |

| **RF3. “Compartilhar Postagens”** | | |
| --- | --- | --- |
| **Descrição:** O usuário terá a opção de compartilhar as postagens. | | |
| **Atores:** Todos os usuários | | |
| **Pré-Condições:** Conexão com internet e acesso a aba de postagens | | |
| **Pós-Condições:** Postagens compartilhadas com sucesso. | | |
| **Fluxo básico:**   1. O usuário acessa uma postagem e clica na opção de compartilhamento desejada. | | |

| **RF4. “Fazer login no site com o usuário da Unochapecó”** | | |
| --- | --- | --- |
| **Descrição:** Login dos alunos e/ou do administrador a partir do e-mail e senha da Unochapecó. | | |
| **Atores:** Administrador ou aluno | | |
| **Pré-Condições:** Estar com o site aberto e conectado a internet, e com a conta matrícula e login ativo. | | |
| **Pós-Condições:** Administrador ou aluno logado com sucesso | | |
| **Fluxo básico:**   1. O administrador/aluno informa e-mail e senha da Unochapecó; 2. E após isso confirma com o captcha. | | |
| **Requisito não funcional** | Restrição | Categoria |
| NF 1.1 - “Validação do usuário” | Validação do usuário com o captcha que é gerado. | Segurança |
| NF 1.2 - “Requisito legal de dados pessoais” | O sistema deverá atender a Lei Geral da Proteção de Dados Pessoais, onde apenas o nome do aluno e imagem (opcional) irá constar na postagem. | Legais |

| **RF3. “Enviar solicitação para postagem do trabalho”** | | |
| --- | --- | --- |
| **Descrição:** Escrever o trabalho, anexar imagens caso necessário, e encaminhar para análise do administrador. | | |
| **Atores:** Aluno | | |
| **Pré-Condições:** Usuário estar logado no site, conexão com internet e ter acesso a aba de envio | | |
| **Pós-Condições:** Trabalho enviado com sucesso | | |
| **Fluxo básico:**   1. O Aluno cria uma nova solicitação de postagem, escreve e envia os dados necessários para análise do administrador, onde o status do trabalho ficará como pendente. | | |
| **Requisito não funcional** | Restrição | Categoria |
| NF 1.1 “Restrição de acesso" | Somente os alunos terão acesso. | Segurança |

| **RF4. “Aprovar postagem”** | | |
| --- | --- | --- |
| **Descrição:** O administrador aprova a postagem do aluno e, após isso, o trabalho será redirecionado para a home, onde terão todas as postagens aprovadas pelo administrador. | | |
| **Atores:** Administrador | | |
| **Pré-Condições:** Administrador estar logado no site, conexão com internet e acesso ao menu de pendências. | | |
| **Pós-Condições:** Status do trabalho alterado para aprovado e postagem publicada com sucesso. | | |
| **Fluxo básico:**   1. O Administrador entra na tela de pendências, abre o trabalho de um aluno, clicando em cima daquele que ele deseja analisar, a partir da lista de pendentes. 2. Altera o status do trabalho para aprovado. | | |
| **Requisito não funcional** | Restrição | Categoria |
| NF 1.1 “Verificação de cargo” | Somente o administrador terá acesso a lista de pendentes e para alterar o status da postagem. | Segurança |

| **RF5 .“Editar postagem”** | | |
| --- | --- | --- |
| **Descrição:** O administrador edita a postagem do aluno e, após concluir, será redirecionada para a home, onde terão todas as postagens aprovadas pelo administrador. | | |
| **Atores:** Administrador | | |
| **Pré-Condições:** Administrador estar logado no site, conexão com internet e ter acesso ao menu de pendentes | | |
| **Pós-Condições:** Status do trabalho alterado para aprovado depois da edição e postagem publicada com sucesso. | | |
| **Fluxo básico:**   1. O Administrador entra na tela de pendências, abre o trabalho de um aluno, clicando em cima daquele que ele deseja analisar, a partir da lista de pendentes. 2. Edita o trabalho e após isso edita o status do trabalho para aprovado. | | |

| **RF 6. “Recusar postagem”** | | |
| --- | --- | --- |
| **Descrição:** O administrador recusa a postagem do aluno. | | |
| **Atores:** Administrador | | |
| **Pré-Condições:** Administrador estar logado no site, conexão com internet e acesso ao menu de pendentes | | |
| **Pós-Condições:** O status do trabalho é alterado para recusado. | | |
| **Fluxo básico:**   1. O Administrador entra na tela de pendências, abre o trabalho de um aluno, clicando em cima daquele que ele deseja analisar, a partir da lista de pendentes. 2. Analisa o trabalho e após isso muda o status para recusado. | | |

| **RF 6. “Buscar postagem”** | | |
| --- | --- | --- |
| **Descrição:** Usuário irá buscar por palavra chave do título da postagem no campo de busca | | |
| **Atores:** Todos os usuários | | |
| **Pré-Condições:** Basta conexão com internet e acessar o site | | |
| **Pós-Condições:** Resultado da busca com os trabalhos correspondentes a busca. | | |
| **Fluxo básico:**   1. O Usuário acessa o site e clica no campo de busca. 2. Digita a palavra-chave que deseja e da enter. 3. Serão listados os trabalhos que possuírem aquela(s) palavra(s)-chave. | | |
| **Fluxo alternativo:**   1. O Usuário acessa o site e clica no campo de busca. 2. Digita a palavra-chave que deseja e da enter. 3. Não é encontrado nenhum trabalho com aquela(s) palavra(s)-chave. | | |

## Requisitos não-funcionais

### Requisitos de desempenho

* Requisito de resposta do sistema: O sistema deverá responder a ação realizada em um intervalo de 2s.
* Requisito de temporização: O sistema deve efetuar leitura de dados 5 vezes por segundo.

### Requisitos de dados persistentes

Descrevem-se aqui estruturas lógicas de dados persistentes (que mantém seu valor após a execução do programa) que sejam usadas pelo produto. Cada estrutura de dados pode ser, por exemplo, um arquivo convencional ou uma tabela em um banco de dados.

INCLUIR AQUI O MODELO DE BANCO DE DADOS

### Restrições ao desempenho

Restrições de projeto impostas por padrões externos, com influência da legislação..

Legislação: Há várias leis e regulamentações que podem afetar um site de veterinária, dependendo da localização do negócio. Por exemplo, alguns países exigem que os registros médicos dos animais de estimação sejam mantidos em formato de papel ou eletrônico, e que as informações sejam armazenadas de forma segura e confidencial. Outras leis podem exigir a obtenção de licenças ou permissões específicas para operar um negócio de veterinária.

Conformidade com padrões: Além das leis e regulamentações, há padrões de segurança e privacidade que um site de veterinária deve seguir. Por exemplo, o site pode precisar estar em conformidade com a Lei Geral de Proteção de Dados (LGPD) para proteger informações pessoais dos usuários.

Restrições de conteúdo: Algumas legislações podem restringir o tipo de conteúdo que um site de veterinária pode exibir, como proibir a venda de medicamentos sem prescrição veterinária ou a divulgação de informações médicas que possam ser consideradas enganosas ou falsas.

Limitações técnicas: Algumas funcionalidades podem ter limitações técnicas que restringem o desenvolvimento de certos recursos, como a capacidade de integrar com sistemas legados de gerenciamento de clínicas ou de telemedicina.

Restrições financeiras: O orçamento disponível para o projeto pode limitar o escopo ou a qualidade do trabalho realizado, como a implementação de recursos avançados de inteligência artificial ou o desenvolvimento de um aplicativo móvel personalizado.

### Atributos de Qualidade

Funcionalidades:

Login do(a) aluno(a);

Login do(a) professor(a);

Postar trabalho (aluno)

Aprovar, editar ou recusar trabalho (professor)

Ler, curtir e/ou compartilhar posts (todos os usuários)

Confiabilidade:

Preza pelo menor grau de erros, e quando ocorrem, consegue se recuperar de forma rápida e assertiva;

Usabilidade:

Fácil e rápido de usar, possuindo conceitos fáceis, visíveis e intuitivos para o usuário, sempre de acordo com as normas propostas;

Eficiência:

O software possui apenas os recursos necessários, tendo um tempo de resposta e processamento de no máximo 3s;

Manutenibilidade:

Facilidade para encontrar e resolver problemas, e da mesma forma, intuitivo e simples para testar.

Portabilidade:

É de fácil disponibilidade e acesso aos usuários, se adaptando tanto para desktop quanto para mobile;

# Análise de UCP

As tabelas de escopo de valor do produto e tempo de desenvolvimento com Use Case Points - UCP.

1. **Modelo Conceitual**
2. **Diagrama de Atividades**
3. **Diagrama de Sequência**
4. **Diagrama de Classes**

Referências:

*IEEE Std. 830 – 1993. IEEE Recommended Practice for Software Requirements Specifications.*

*IEEE ISO/IEC/IEEE 29148 – 2011. IEEE Systems and software engineering — Life cycle processes — Requirements engineering*

**OBSERVAÇÃO: Os itens deste modelo de especificação, recomendado pela IEEE, poderão ser complementados com novos itens caso sejam justificáveis.**