

Documento de Modelagem de Software e Arquitetura de Sistemas do Projeto Integrador

Requisitos da Disciplina - Modelagem de Software e Arquitetura de Sistemas

- 1- Identificar 6 requisitos funcionais do sistema
- 2- Identificar 6 requisitos não-funcionais do sistema
- 3- Construir 3 casos de uso
- 4- Apresentar o projeto de arquitetura da implementação do sistema.

1 - Requisitos funcionais do sistema:

- **Cadastro de Usuário:** Permitir que os clientes se cadastrem na loja virtual, inserindo informações como nome, endereço, e-mail e senha.
- **Login de usuário:** Se o usuário já tiver se cadastrado antes, ele pode realizar seu login com os dados informados anteriormente.
- **Navegação de Produtos:** Disponibilizar uma interface intuitiva para os usuários navegarem pelos produtos da loja, com opções de pesquisa e categorias.
- **Visualizar um produto específico:** Ao clicar em um produto da tela de produtos, ela é levada à página específica desse produto, onde pode visualizar suas características e seu valor.
- **Levar ao contato de compra:** Ao clicar em comprar, levar o usuário à um contato da vendedora para realizar a compra daquele produto específico.
- **Visualização do header:** Possuir um header visível para o usuário clicar na página que deseja acessar, a qualquer momento da navegação.

2 - Requisitos Não-Funcionais do Sistema:

- **Responsividade:** Garante um sistema responsivo, para funcionar em diferentes plataformas.
- **Criptografia:** Possuir medidas de criptografia para as senhas dos usuários.
- **Usabilidade:** Priorizar uma interface intuitiva e amigável para o usuário, facilitando a navegação e a realização de compras.
- **Escalabilidade:** Projetar o sistema para lidar com o crescimento futuro, tanto em termos de número de usuários quanto de produtos.

- **Disponibilidade:** Assegurar alta disponibilidade do sistema, minimizando períodos de inatividade para manutenção ou atualizações.
- **Compatibilidade com Navegadores:** Ser compatível com os principais navegadores, proporcionando uma experiência consistente para todos os usuários.

3 - Casos de Uso:

Realizar Compra

Ator principal: Cliente

Fluxo:

O cliente acessa a home.

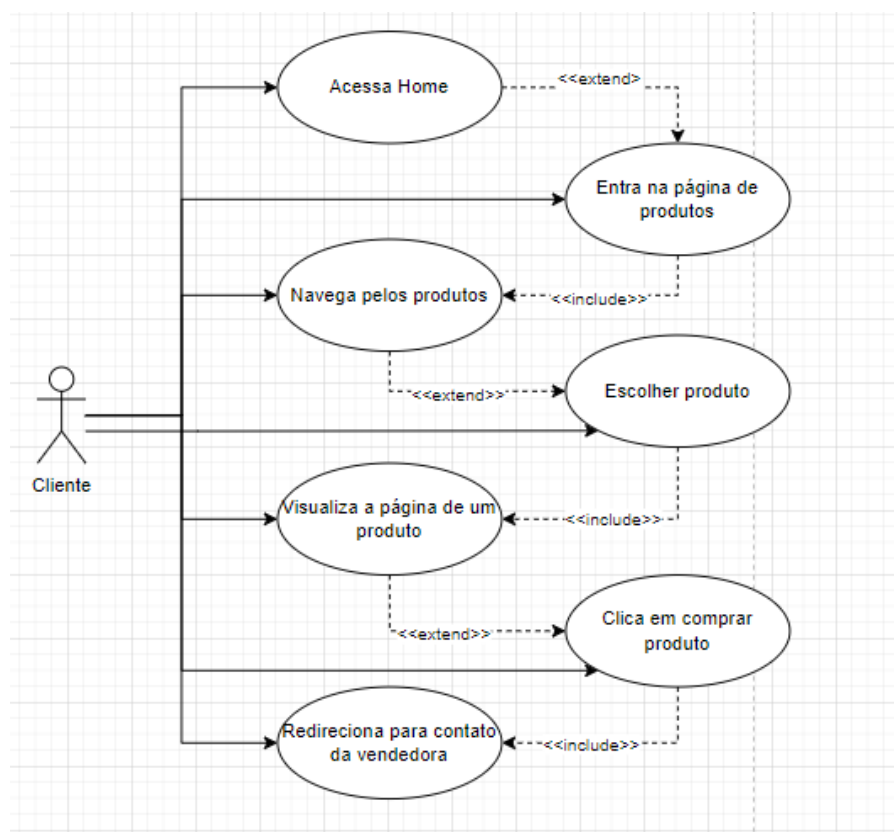
O cliente entra na página de produtos através do header.

O cliente navega pelos produtos.

O cliente escolhe um produto e clica no mesmo, entrando no anúncio desse produto.

O cliente visualiza as informações do produto.

O cliente clica em comprar, assim entrando em contato com a vendedora.



Fazer cadastro

Ator principal: Cliente

Fluxo:

O cliente acessa a home.

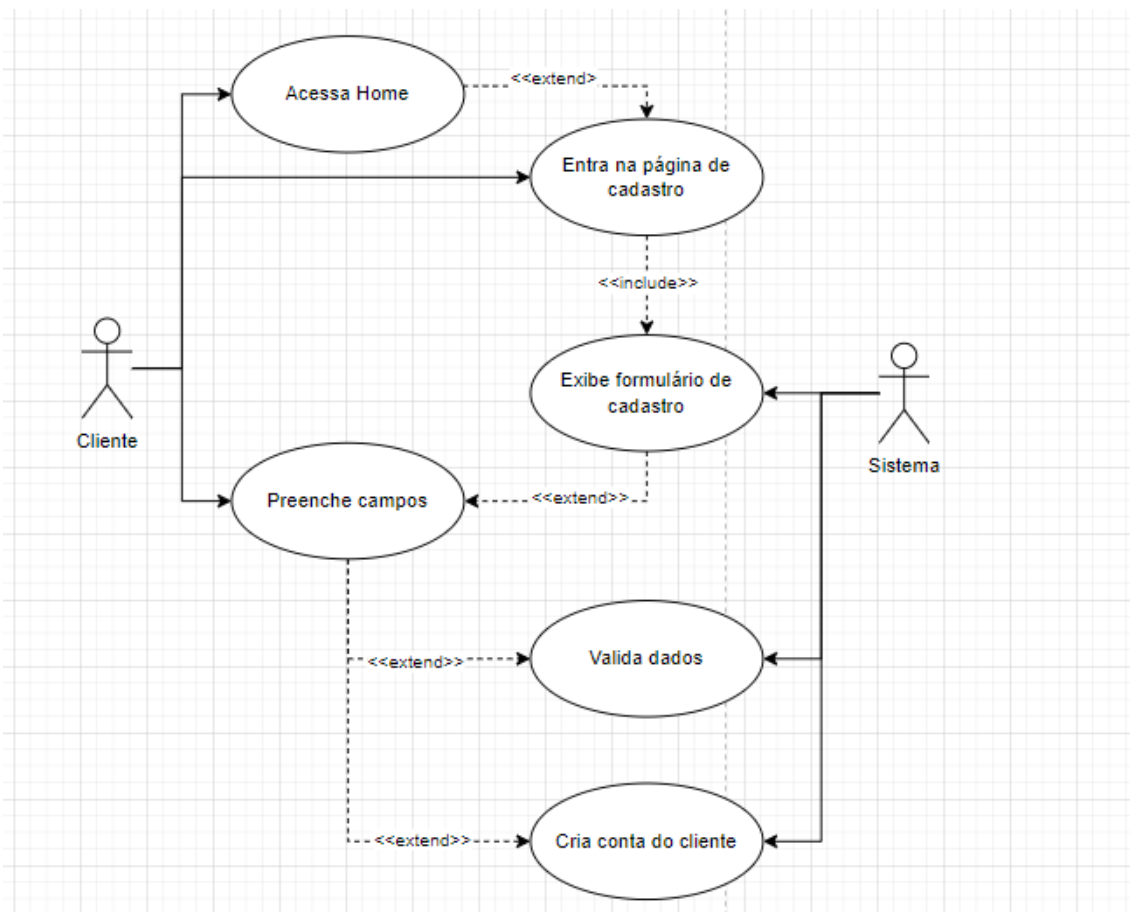
Cliente acessa a página de cadastro através do header.

O sistema exibe um formulário de cadastro contendo campos como nome, e-mail senha e confirmação de senha.

O cliente preenche os campos obrigatórios.

O sistema valida os dados inseridos pelo cliente, verificando a integridade dos campos e a formatação correta do e-mail.

Se todos os dados estiverem corretos, o sistema cria uma conta para o cliente, armazenando as informações no banco de dados.



Pesquisar produtos

Ator principal: Cliente

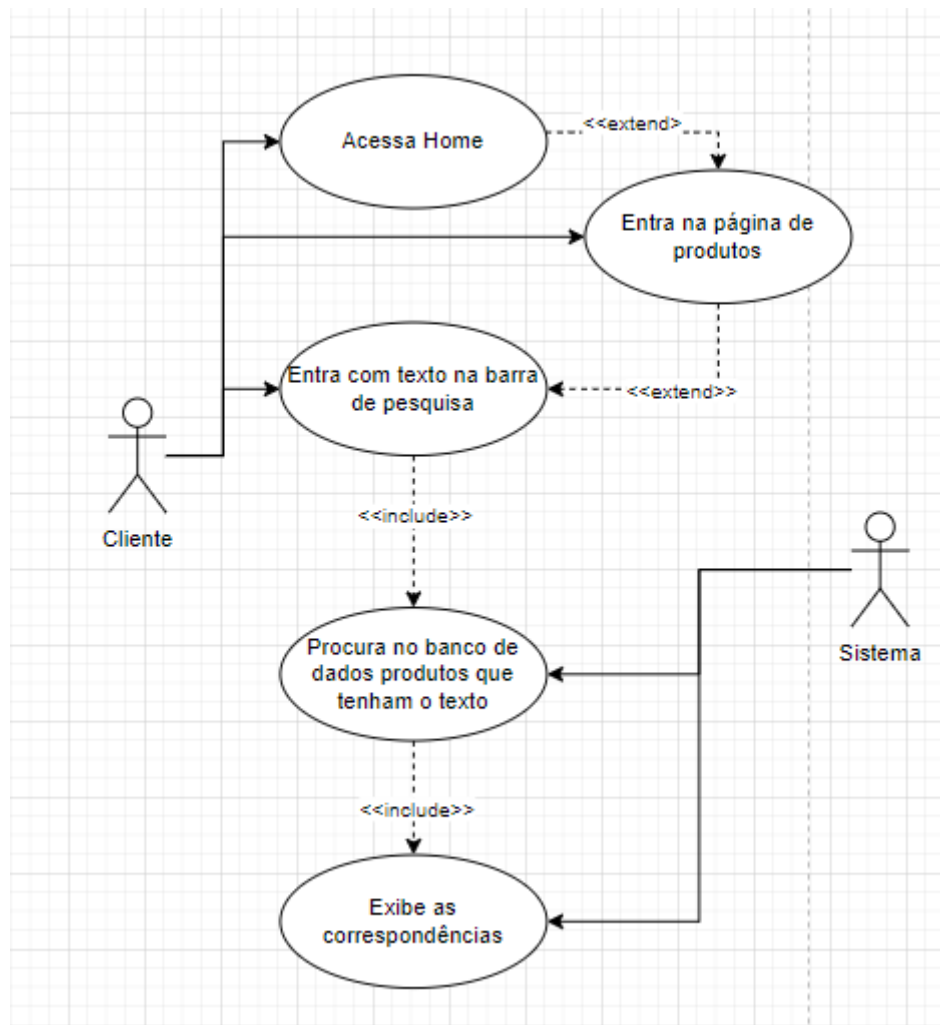
Fluxo:

O cliente acessa a home.

O cliente entra na página de produtos através do header.

O cliente entra com um texto na barra de texto.

O sistema exibe apenas os produtos cadastrados no banco de dados, que contêm o texto fornecido no nome.



4 - Projeto de Arquitetura:

Estrutura MVC (Model-View-Controller) no Projeto:

O projeto adota a arquitetura MVC (Model-View-Controller) para organizar e estruturar sua implementação. O MVC é um padrão de arquitetura de software que separa as responsabilidades do sistema em três componentes principais: Model (Modelo), View (Visão) e Controller (Controlador). Essa abordagem oferece diversos benefícios em termos de manutenção, escalabilidade e modularidade.

Características do MVC no Projeto:

Model (Modelo):

Responsável pela gestão de dados e lógica de negócios.

Interage com o banco de dados para recuperar e persistir informações.

Independente da interface do usuário.

View (Visão):

Responsável pela apresentação dos dados ao usuário.

Não contém lógica de negócios; apenas exibe as informações fornecidas pelo Model.

Pode ser adaptada para diferentes dispositivos ou interfaces sem alterar a lógica subjacente.

Controller (Controlador):

Recebe solicitações do usuário e atualiza o Model e a View conforme necessário.

Contém a lógica que coordena a interação entre o Model e a View.

Facilita a comunicação bidirecional entre o Model e a View.

