



CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DANILO BARBIERI FIGUEIREDO
FERNANDA NUNES CALDANI
GIOVANA GIORGI BIGATI
LAURA SANTORO CRUZ
PEDRO HENRIQUE FIRMANI AZEVEDO

TÂNIA CAMILA KOCHMANSCKY GOULART

PROJETO DE EXTENSÃO UNIVERSITÁRIA

Londrina
2024

SUMÁRIO

1	INTRODUÇÃO	2
1.1	PROPOSTA E OBJETIVOS	2
1.2	JUSTIFICATIVA	3
2	REQUISITOS	4
2.1	Requisitos Funcionais	4
2.2	Requisitos Não Funcionais	6
3	RECURSOS E AMBIENTE DE DESENOVIMENTO	7
3.1	Metodologia	7
3.2	Linguagem de Programação	7
3.3	Frameworks	9
3.4	Bibliotecas e Dependências	10
3.5	Banco de dados	12
3.6	Teste e Segurança	13
3.7	Segurança	14
4	WORKFLOW	15
4.1	Especificação de Caso de Uso	15
4.2	Diagrama Entidade Relacionamento	16
4.3	Diagramas UML	17
5	PROTÓTIPO	19
5.1	Telas	19

1 INTRODUÇÃO

O projeto de extensão denominado "Uni HomeMatch", em desenvolvimento pelos alunos de Ciência da Computação e Engenharia de Software da Universidade Filadélfia de Londrina, destina-se a complementar os conhecimentos teóricos obtidos na universidade por meio de práticas. Juntamente com a orientação dos professores do curso, permite ao acadêmico a compreensão da importância dessa associação para a capacidade de um raciocínio correto, lógico e prático.

O aluguel de quartos propõe ao estudante de diversos cursos a chance de possuir menos gastos e lidar com a convivência com outros estudantes, sejam do mesmo curso ou de diferentes áreas. Porém, o diferencial do projeto exposto, é que as pessoas participantes contam com o acompanhamento regular de todo o processo e podem colocar seus gostos e preferências como meio de localizar pessoas que possuem e compartilham, na maioria das vezes, os mesmos ideais e discutirem, sobre o aluguel, por um chat virtual. Assim, no instante em que as dúvidas surgem já são sanadas e os universitários são estimulados a perceber e analisar detalhes construtivos na locação durante o momento da visita.

1.1 PROPOSTA E OBJETIVOS

O presente projeto tem como objetivo geral propor uma solução para a falta de comunicação entre universitários para realizar um acordo de divisão de apartamentos. Como objetivos específicos podemos citar:

- Realizar um sistema de acompanhamento regular do processo de aluguel para garantir a satisfação e segurança dos estudantes;
- Criar uma plataforma que permita aos participantes compartilharem seus gostos e preferências para encontrar colegas de moradia compatíveis;
- Debater sobre questões relacionadas ao aluguel por meio de um chat virtual, visando esclarecer dúvidas e fornecer suporte aos usuários;
- Estimular os universitários a analisarem e avaliarem detalhes construtivos durante as visitas aos locais de moradia;
- Estimular a criação de um ambiente de convivência harmonioso e colaborativo entre os estudantes que compartilham o mesmo espaço de moradia.

1.2 JUSTIFICATIVA

O sistema proposto fornece uma plataforma que facilita a conexão entre locadores e interessados na área universitária. O sistema oferece filtros e preferências que são considerados por universitários que procuram quartos e apartamentos para alugar como localização, perfil do usuário, entre outros e organizar as ofertas de acordo com a melhor concordância, além de um chat virtual para facilitar a comunicação do locador, locatário e a atualização regular do processo de aluguel.

Quando se faz necessário uma mudança pela faculdade, muitas vezes o universitário encontra dificuldade em encontrar moradias que combinem com seu perfil ou caibam em seus requerimentos, já que os existentes serviços de locação não são adaptados às suas necessidades. Visamos aprimorar e simplificar a experiência da moradia universitária, fornecendo um espaço em que o estudante possa encontrar usuários que compartilhe seus ideais ou que seja da mesma área de estudo, melhorando as chances de uma boa vivência na moradia, além de superar quaisquer complicações que poderiam vir a acontecer devido a perfis incompatíveis entre locador e locatário, falta de comunicação ou não transparência do processo.

2 REQUISITOS

2.1 REQUISITOS FUNCIONAIS

Gerenciamento de Usuários:

- Material educativo sobre a importância dos documentos contábeis e como eles contribuem para o processo contábil.
- Pop-ups informativos sobre a necessidade e importância dos documentos relevantes.

Gerenciamento de Imóveis:

- Permite que locadores cadastrem seus imóveis, incluindo detalhes como localização, número de quartos, preço e fotos.
- Permite que locadores atualizem informações sobre seus imóveis, como disponibilidade ou preço.
- Implementa filtros para encontrar imóveis com base em localização, preço, número de quartos e preferências específicas.

Comunicação:

- Fornece uma plataforma de chat virtual para facilitar a comunicação entre locadores e locatários.
- Implementa um sistema de notificações para informar os usuários sobre atualizações de imóveis, mensagens de chat e outros eventos relevantes.

Processo de Aluguel:

- Permite que usuários interessados em um imóvel enviem uma solicitação de aluguel diretamente através da plataforma.
- Fornece uma ferramenta para acompanhar o status do processo de aluguel, desde a solicitação até a assinatura do contrato.
- Implementa um sistema para que os usuários possam avaliar e revisar o processo de aluguel e os locadores após a conclusão do contrato.

Suporte e Ajuda:

- Oferece uma seção de FAQ para resolver dúvidas comuns sobre o processo de aluguel e a utilização da plataforma.
- Permite que os usuários entrem em contato com o suporte técnico ou atendimento ao cliente para resolver problemas ou questões específicas.

Sistema de Pagamento Integrado:

- Implementa um sistema de pagamento eletrônico para permitir o pagamento unificado na plataforma.
- Garante a segurança e a privacidade das informações financeiras dos usuários.

2.2 REQUISITOS NÃO FUNCIONAIS

Segurança:

- Implementa medidas de segurança robustas para proteger as informações pessoais e financeiras dos usuários.
- Garante que os usuários tenham controle sobre suas informações pessoais e que a plataforma respeite a privacidade dos usuários.

Usabilidade:

- A plataforma deve ter uma interface de usuário intuitiva e fácil de usar.
- A plataforma deve ser acessível a todos os usuários, incluindo aqueles com deficiências.

Interoperabilidade:

- A plataforma deve ser capaz de integrar-se com outros sistemas ou plataformas.
- A plataforma deve ser compatível com uma variedade de dispositivos e navegadores.

Manutenibilidade:

- Adota uma arquitetura modular para facilitar a manutenção e a atualização do sistema.
- Integrar ferramentas de monitoramento e registro de logs.
- Utiliza um sistema de controle de versão para gerenciar o código-fonte.

Escalabilidade:

- A plataforma deve ser capaz de escalar para acomodar um aumento no número de usuários e na quantidade de dados, sem comprometer a performance.
- Para usuários distribuídos geograficamente, a plataforma deve oferecer suporte a georreplicação para garantir acesso rápido e confiável, independentemente da localização.

3 RECURSOS E AMBIENTE DE DESENVOLVIMENTO

3.1 METODOLOGIA

O Scrum é caracterizado por sua abordagem dinâmica e flexível, o que lhe permite fazer ajustes contínuos e se adaptar rapidamente às mudanças, sendo uma metodologia ágil. Devido à sua eficácia em promover habilidades interpessoais como flexibilidade, resiliência e espírito colaborativo entre os membros da equipe, essa metodologia é amplamente adotada em vários setores, incluindo empresas de tecnologia e projetos pessoais.

A metodologia Scrum é baseada em ciclos iterativos chamados Sprints, que geralmente duram um mês, permitindo a entrega incremental de funcionalidades. A equipe realiza reuniões diárias, conhecidas como Daily Scrum, durante cada Sprint para avaliar o progresso e planejar as próximas etapas. O Scrum também destaca a importância das pessoas em vez dos processos, criando uma cultura aberta para sugestões e mudanças, sem limitações ao projeto original. Isso leva a uma maior satisfação do cliente, aumento da taxa de sucesso nos projetos e geração de resultados estratégicos.

A implementação do Scrum inclui a designação de papéis estratégicos, como o Product Owner, encarregado da visão do produto, e o Scrum Master, responsável por otimizar o fluxo de entrega e organizar os recursos necessários. Durante os Sprints, são os desenvolvedores que realizam o trabalho na equipe.

Escolher o Scrum como metodologia de gestão de projetos traz vantagens importantes, tais como o desenvolvimento de habilidades interpessoais, aumento da flexibilidade e resiliência da equipe, promoção de uma cultura aberta à mudança e garantia da satisfação do cliente por meio de entregas incrementais e adaptativas. Equipes que buscam eficiência, adaptabilidade e colaboração contínua encontram no Scrum uma escolha ideal devido a essas vantagens.

3.2 LINGUAGEM DE PROGRAMAÇÃO

JavaScript foi escolhido como a linguagem de programação para nossa aplicação devido a várias vantagens que fazem dela a opção perfeita para desenvolver aplicações web modernas e dinâmicas. JavaScript, criada em 1996, é uma linguagem de programação de alto nível conhecida por ser versátil e capaz de ser executada diretamente no navegador, dispensando a necessidade de um compilador. Isso facilita sua integração com HTML e CSS, proporcionando assim uma experiência rica e interativa para o usuário.

Sua acessibilidade para desenvolvedores iniciantes e experientes torna o JavaScript fácil de aprender, sendo uma de suas principais vantagens. Adicionalmente, a linguagem possibilita desenvolver funcionalidades dinâmicas em páginas web, o que melhora consideravelmente a experiência do usuário com animações, interações personalizadas e respostas em tempo real.

JavaScript também tem fama de ser compatível com várias plataformas e navegadores, o que garante que as aplicações desenvolvidas sejam acessíveis e funcionem eficientemente em diferentes dispositivos. Isso é essencial para garantir que as aplicações web sejam responsivas e acessíveis a todos os usuários, independentemente do dispositivo que utilizam.

O JavaScript também apresenta vários frameworks e bibliotecas que simplificam a criação de aplicações complexas, permitindo que os desenvolvedores foquem na lógica de negócios e na experiência do usuário, em vez de lidar com detalhes técnicos mais específicos. Isso abrange frameworks para criar aplicativos web, móveis e até mesmo jogos eletrônicos, exibindo a versatilidade e alcance de aplicações do JavaScript.

Para complementar o uso do JavaScript em nossa aplicação, decidimos utilizar o TypeScript, uma linguagem que adiciona tipagem estática ao JavaScript. O TypeScript, desenvolvido pela Microsoft e lançado em 2012, é um superconjunto do JavaScript que compila para JavaScript puro, o que significa que qualquer código JavaScript é também um código TypeScript válido. A principal vantagem do TypeScript é a introdução de tipos estáticos, que ajudam a identificar erros de forma antecipada durante o desenvolvimento, facilitando a manutenção e melhorando a robustez do código.

A adição de tipagem estática torna o código mais seguro e previsível, permitindo que os desenvolvedores detectem erros comuns, como o uso de variáveis incorretas ou a passagem de argumentos inesperados para funções, antes mesmo da execução. Isso é particularmente útil em aplicações complexas, onde a tipagem estática pode reduzir significativamente o número de bugs e o tempo gasto em testes e depuração.

O TypeScript também oferece suporte a recursos avançados de programação, como interfaces, classes e módulos, que facilitam a escrita de código limpo, organizado e escalável. Além disso, ele é altamente compatível com ferramentas modernas de desenvolvimento, como editores de código, sistemas de build e integração contínua, aumentando a produtividade da equipe de desenvolvimento.

Outra vantagem importante do TypeScript é a facilidade de migração de projetos JavaScript existentes. Como TypeScript é um superconjunto de JavaScript, os desenvolvedores podem gradualmente adotar a tipagem estática em partes específicas do projeto sem a necessidade de reescrever o código existente. Essa abordagem incremental permite que equipes aproveitem os benefícios da tipagem estática e ou-

tras melhorias sem interromper o fluxo de desenvolvimento.

Em resumo, o uso combinado de JavaScript e TypeScript permite desenvolver aplicações web modernas e dinâmicas com maior segurança, manutenção simplificada e uma experiência de desenvolvimento mais eficiente.

3.3 FRAMEWORKS

• FRAMEWORK FRONT-END

Optamos pelo framework React para desenvolvimento frontend da nossa aplicação por conta de diversas vantagens que o tornam uma escolha excepcional para projetos modernos e dinâmicos. O Facebook criou e mantém a biblioteca JavaScript React, que é famosa por sua eficiência, flexibilidade e capacidade de criar interfaces de usuário simples, rápidas e escaláveis para aplicações web.

A arquitetura baseada em componentes do React é uma das principais vantagens, pois permite reutilizar código e economizar tempo e esforço dos desenvolvedores. Isso quer dizer que, em lugar de escrever numerosos códigos para as mesmas funcionalidades, os desenvolvedores podem criar componentes reutilizáveis que podem ser facilmente integrados em diversas partes da aplicação. Além do mais, o código garante maior estabilidade e facilidade de manutenção porque qualquer alteração em um componente não afeta as outras partes da aplicação.

A comunidade de desenvolvedores do React é muito forte na indústria, com muitos contribuidores e uma variedade de recursos disponíveis, incluindo tutoriais, artigos e respostas em fóruns como Stack Overflow e Quora. Caso encontre problemas ou tenha dúvidas, é provável que você receba soluções e suporte de especialistas na comunidade.

O React é mais fácil de aprender do que outros frameworks frontend populares, como Angular e Vue. Empresas e grandes marcas podem começar a desenvolver seus projetos rapidamente, o que facilita a adoção do React. O aumento da produtividade dos desenvolvedores também é contribuído pela facilidade de aprendizado e pelas extensões e ferramentas de desenvolvimento disponíveis para o React, como o React Developer Tools.

• FRAMEWORK BACK-END

Optamos pelo framework Express.js para o desenvolvimento do backend de nossa aplicação por conta de sua simplicidade, flexibilidade e eficiência. O Express.js, construído sobre o Node.js, é um framework minimalista que provê uma variedade extensa de ferramentas e funcionalidades para criar aplicações web confiáveis e escaláveis. Ele torna mais simples a criação de servidores HTTP, o

gerenciamento de requisições e respostas, bem como a integração com bancos de dados para desenvolver serviços backend em aplicativos online e móveis.

Mais de 1900 empresas em todo o mundo usam o Express.js, mostrando sua popularidade e confiabilidade. Com a inclusão de tecnologias como roteamento, middleware, motores de template e integração com banco de dados, ele possibilita a criação de aplicações web, serviços backend e APIs RESTful. Além disso, tem uma comunidade de suporte forte que oferece uma ampla variedade de recursos, ferramentas e frameworks para ajudar no desenvolvimento.

Entretanto, é fundamental observar que o Express.js possui suas limitações, tais como a falta de recursos integrados como autenticação, autorização, validação, testes e logging. Para incorporar estas características, os programadores devem confiar em módulos ou bibliotecas de outros fornecedores, o que pode aumentar a complexidade e a gestão das dependências do projeto.

Mesmo com essas limitações, o Express.js continua sendo uma ótima opção para desenvolvedores experientes que buscam maior flexibilidade e personalização em seus aplicativos da web. O Express.js é uma escolha atraente para muitos projetos devido à sua base em JavaScript, que proporciona uma comunidade robusta de suporte e facilidade de aprendizado.

3.4 BIBLIOTECAS E DEPENDÊNCIAS

Para o desenvolvimento de nossa aplicação, optamos por uma combinação de bibliotecas e tecnologias que oferecem flexibilidade, eficiência e uma experiência de usuário rica e interativa.

- **Filtro**

Para gerenciar o estado da aplicação e a interface do usuário no frontend, utilizamos React Query. Esta biblioteca simplifica o gerenciamento de estado assíncrono, permitindo que os desenvolvedores busquem, sincronizem e atualizem o estado da aplicação de forma eficiente. Garantir que os dados exibidos na interface do usuário estejam sempre atualizados e consistentes é crucial para uma boa experiência do usuário.

Adicionalmente, utilizamos o Redux para o gerenciamento de estado no frontend, o que possibilita uma gestão previsível do estado da aplicação. Isso simplifica o desenvolvimento de interfaces de usuário complexas e melhora a experiência do usuário, garantindo a consistência dos dados em toda a aplicação.

- **Sistema de Mensagem**

Decidimos usar WebSocket com a biblioteca Socket.IO para facilitar a implementação do protocolo e oferecer recursos extras para aplicações em tempo real, como um sistema de mensagens. O Socket.IO permite uma comunicação bidirecional em tempo real entre o cliente e o servidor, facilitando a entrega instantânea de mensagens e atualizações. A biblioteca também é compatível com a criação de salas e namespaces, que permitem agrupar usuários para bate-papo em grupo e enviar mensagens para vários clientes ou grupos específicos. Ela ainda fornece recursos de reconexão automática, assegurando uma experiência contínua e confiável, mesmo em caso de interrupções na conexão.

Para armazenar as mensagens do sistema de mensagens, utilizamos um banco de dados relacional. Isso garante a persistência das mensagens, permitindo que sejam recuperadas e exibidas quando necessário, além de facilitar a implementação de funcionalidades como busca de mensagens, recuperação de histórico de conversas e gerenciamento de salas de chat.

- **Integração de Pagamento**

Optamos pela API de Pagamento PagSeguro para integrar pagamentos em nossa aplicação, uma solução confiável e segura com diversas funcionalidades para processar pagamentos online. A API facilita a integração de vários métodos de pagamento, como cartões de crédito, boletos bancários e transferências bancárias, permitindo transações eficientes e seguras. Ela também oferece recursos avançados, como a análise de risco de fraude, para proteger as transações contra atividades suspeitas.

Implementamos criptografia SSL/TLS para assegurar a segurança das transações, garantindo que os dados transmitidos entre o cliente e o servidor estejam protegidos contra interceptações ou alterações. A implementação de SSL/TLS garante a segurança e confiabilidade das comunicações entre a aplicação e a API de PagSeguro, assegurando a integridade e confidencialidade dos dados.

- **Design Responsivo**

Para assegurar que nosso aplicativo tenha um design responsivo, utilizamos frameworks CSS como Bootstrap e Materialize CSS. Esses frameworks são fundamentais para criar interfaces de usuário que se adaptam automaticamente a diferentes tamanhos de tela, proporcionando uma experiência consistente em dispositivos móveis, tablets e desktops. O Bootstrap, por exemplo, é desenvolvido com uma abordagem mobile-first, utilizando media queries para ajustar o layout e os elementos da interface conforme o tamanho da viewport.

Adicionalmente, utilizamos bibliotecas de estilização como @emotion/react, @emotion/styled, e styled-components para facilitar a criação de componentes perso-

nalizados e responsivos, permitindo uma personalização mais detalhada da interface do usuário.

- **Sistema de Autenticação**

Para proteger as credenciais dos usuários armazenadas no banco de dados, utilizamos o bcrypt para hashing de senhas. A função de hashing de senha bcrypt utiliza o algoritmo Blowfish para armazenar senhas de forma segura, garantindo a preservação da segurança das senhas dos usuários mesmo em caso de comprometimento do banco de dados.

Para a autenticação, implementamos tokens JWT (JSON Web Tokens), que permitem a representação segura de informações entre duas partes. Após a autenticação bem-sucedida, geramos tokens de acesso e atualização usando JWT, que são enviados ao usuário. Esses tokens permitem que a aplicação mantenha o estado da sessão do usuário sem precisar armazenar informações de sessão no servidor, sendo usados para autenticar as solicitações subsequentes.

Utilizamos middleware de autenticação JWT para proteger os endpoints da API, garantindo que somente usuários autenticados possam utilizá-los. Todos os endpoints de autenticação são protegidos por HTTPS para assegurar a segurança das comunicações, criptografando as informações transmitidas entre o cliente e o servidor, como credenciais de login e tokens de autenticação.

3.5 BANCO DE DADOS

O motivo de escolhermos o MySQL como sistema de gerenciamento de banco de dados relacional para nossa aplicação é uma série de vantagens que o tornam excepcional para projetos modernos e dinâmicos. O MySQL, criado pela Oracle Corporation, é um sistema de gerenciamento de banco de dados relacional open source que fornece uma solução eficiente para armazenar e organizar dados.

O MySQL é amplamente popular e confiável, sendo adotado por grandes empresas da web como Facebook, Twitter, YouTube, Google e Yahoo, entre outros. Isso é uma das principais vantagens do MySQL. A eficácia do MySQL em lidar com grandes volumes de dados e requisitos de desempenho é demonstrada por isso.

O MySQL é famoso por sua capacidade de escalabilidade, conseguindo lidar com grandes volumes de dados e suportando replicação e balanceamento de carga. Essa característica a torna uma escolha escalável para empresas que necessitam expandir seus bancos de dados gradualmente.

O MySQL é reconhecido pela sua segurança, fornecendo funcionalidades de autenticação, autorização e criptografia de dados que contribuem para a manutenção da segurança dos dados. Essa é uma necessidade crucial para as empresas que precisam guardar informações confidenciais.

3.6 TESTE E SEGURANÇA

Nós utilizamos o framework **Jest** para testes de unidade e integração. O Jest é capaz de escrever testes para validar o comportamento do código, incluindo a lógica de negócios e a interação com APIs. O Jest também é capaz de testar vulnerabilidades de segurança em APIs, como controle inadequado de acesso, falhas criptográficas, injeções e design inseguro, garantindo assim a capacidade da nossa aplicação para resistir a ataques reais.

Para testar a segurança, utilizamos o **OWASP ZAP** e o **Burp Suite**. Identificar vulnerabilidades em tempo real, como falhas de autenticação, gerenciamento de sessão e segurança de API, é essencial com o uso dessas ferramentas automatizadas de teste de penetração. Quando testamos a aplicação de forma dinâmica (DAST) com essas ferramentas, podemos assegurar que nossa aplicação está protegida contra ameaças comuns e pode lidar eficazmente com ataques.

O **ESLint** é usado para realizar a auditoria de código, ajudando na identificação e correção de problemas no código, incluindo vulnerabilidades comuns em JavaScript, como cross-site scripting (XSS). O ESLint desempenha um papel crucial no processo de assegurar a qualidade, pois permite que os desenvolvedores mantenham seu código limpo e seguro desde o início do desenvolvimento.

Além disso, utilizamos o JWT (**JSON Web Token**) para autenticação, garantindo uma representação segura de informações entre duas partes. O JWT permite que a aplicação mantenha o estado da sessão do usuário de forma segura, sem precisar armazenar informações de sessão no servidor. Isso ajuda a proteger contra ataques de sessão e outras ameaças relacionadas à autenticação.

As dependências do projeto são verificadas em busca de vulnerabilidades conhecidas pelo **NPM Audit** durante o gerenciamento. Garantir que todas as bibliotecas e pacotes usados estejam atualizados é crucial para garantir que estejam livres de vulnerabilidades exploráveis por atacantes.

Também, nós implementamos criptografia **HTTPS** para assegurar a segurança das comunicações entre o cliente e o servidor, protegendo os dados transmitidos contra interceptações.

Por fim, adotamos políticas de segurança robustas, incluindo a autenticação de dois fatores (**2FA**) para aumentar a segurança dos usuários e a conformidade **com a Lei Geral de Proteção de Dados (LGPD)** para garantir a proteção dos dados pessoais dos usuários.

3.7 SEGURANÇA

A Lei Nº 13.709, de 14 de agosto de 2018, conhecida como a Lei Geral de Proteção de Dados (LGPD), estabelece regras para a coleta, tratamento, armazenamento e compartilhamento de dados pessoais no Brasil. A Lei Nº 13.853, de 2019, atualiza e complementa a LGPD, introduzindo novas diretrizes e reforçando a proteção dos dados pessoais.

A LGPD e sua atualização são fundamentais para garantir a privacidade e a segurança dos dados pessoais dos usuários. A lei estabelece direitos para os indivíduos, incluindo o direito de acessar, corrigir, excluir e restringir o uso de seus dados pessoais. Além disso, a LGPD exige que as empresas implementem medidas de segurança adequadas para proteger os dados pessoais, incluindo a criptografia de dados e a implementação de controles de acesso.

A atualização da LGPD, conforme estabelecida pela Lei Nº 13.853, de 2019, traz mudanças significativas, incluindo a introdução de novas regras para a coleta de dados, a definição de novos direitos dos usuários e a implementação de novas obrigações para as empresas. A lei também enfatiza a importância da transparência e da comunicação clara com os usuários sobre como seus dados são coletados, usados e protegidos.

Para estar em conformidade com a LGPD e sua atualização, as empresas devem revisar suas práticas de coleta, tratamento e compartilhamento de dados pessoais. Isso inclui a implementação de políticas de privacidade claras, a obtenção de consentimento informado dos usuários e a garantia de que os dados pessoais sejam coletados, usados e armazenados de forma segura.

A LGPD e sua atualização são diretrizes importantes para garantir a proteção dos dados pessoais no Brasil. Ao aderir a essas leis, as empresas podem não apenas cumprir suas obrigações legais, mas também construir confiança com seus usuários, melhorando a experiência do usuário e a reputação da empresa.

Para mais detalhes, você pode acessar o texto completo da LGPD **neste link**.

4 WORKFLOW

4.1 ESPECIFICAÇÃO DE CASO DE USO

Este documento visa especificar o caso de uso Cadastro de Novo Usuário do sistema Uni HomeMatch, permitindo que estudantes universitários - os atores principais deste caso de uso - criem uma nova conta na plataforma.

• Fluxo Básico de Eventos

- 1) Inicialmente, o ator principal deve abrir a página inicial do Uni HomeMatch e clicar no botão 'Cadastrar-se'.
- 2) O sistema irá redirecionar o usuário para uma página com o formulário de informações necessárias para uma conta.
- 3) O ator deve inserir seu nome completo, sexo, data de nascimento, CPF, email e senha.
- 4) Após verificar que todas as informações estão corretas, o usuário confirma o cadastro clicando em 'Enviar'.
- 5) O sistema valida as informações fornecidas, incluindo a verificação da disponibilidade do e-mail.
- 6) Caso a validação seja efetiva, o sistema confirma o cadastro da conta e concede acesso à plataforma Uni HomeMatch.
- 7) Cada cadastro terá a opção de cadastramento de imóvel para os locadores.

• Fluxos Alternativos

- Dados Inválidos: Se o usuário não preencher todos os campos obrigatórios ou inserir dados inválidos, o sistema exibe uma mensagem de erro e solicita correções.
- E-mail já cadastrado: Se o e-mail informado já estiver em uso, o sistema notifica o usuário e sugere a recuperação de senha.
- Cancelamento de Cadastro: O usuário decide cancelar o cadastro durante o processo.

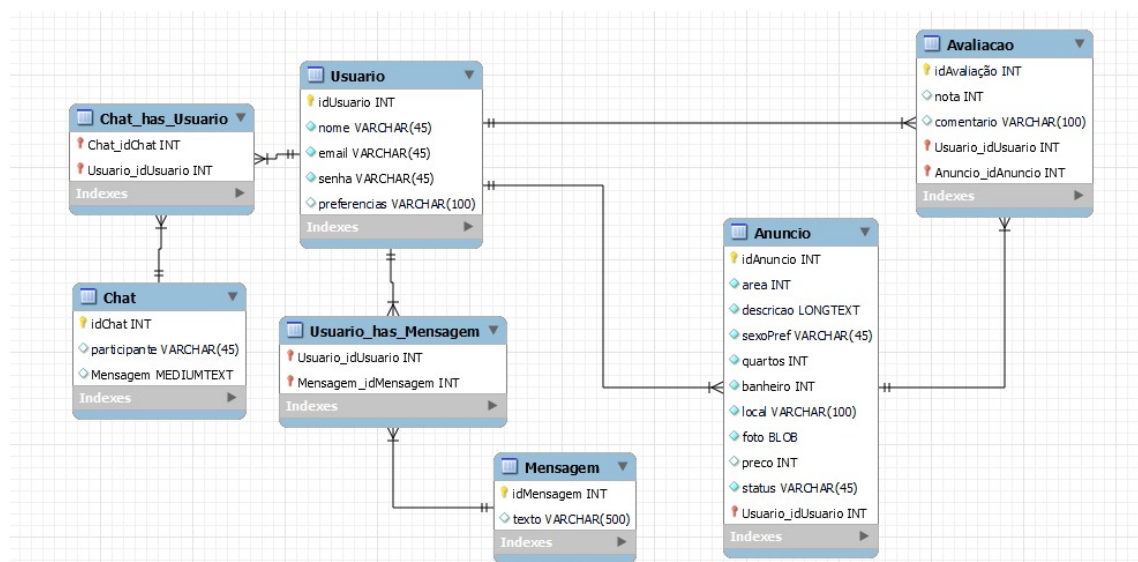
• Condições Prévias

- Acesso a Internet: O usuário deve ter acesso a internet.

- E-mail Válido: O usuário deve ter um endereço de e-mail válido.
- Idade Mínima: O usuário deve ter a idade mínima de 18 anos para criar uma conta
- **Condições Posteriores**
 - Interação com as Postagens: Será autorizado no sistema a interação e visualização total das postagens de locação dos imóveis.
 - Cadastramento de Imóveis: Apenas após o cadastro do usuário será possível efetuar o cadastro de imóveis.

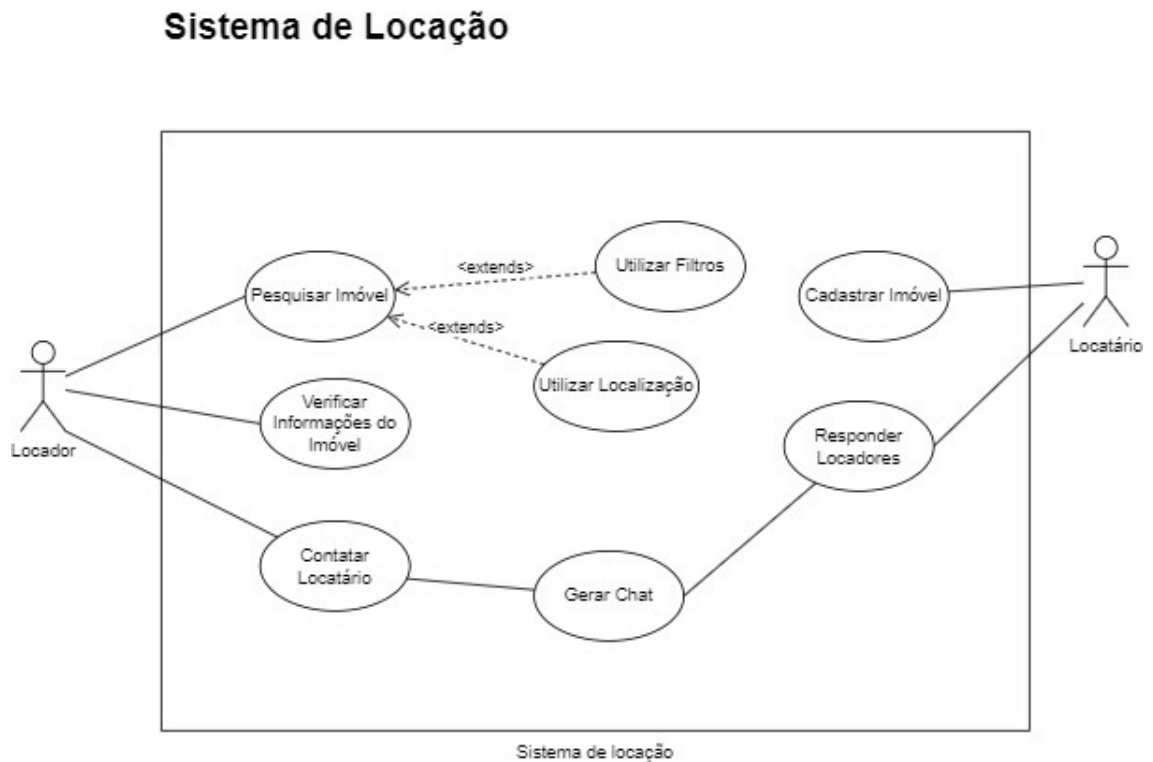
4.2 DIAGRAMA ENTIDADE RELACIONAMENTO

- DER

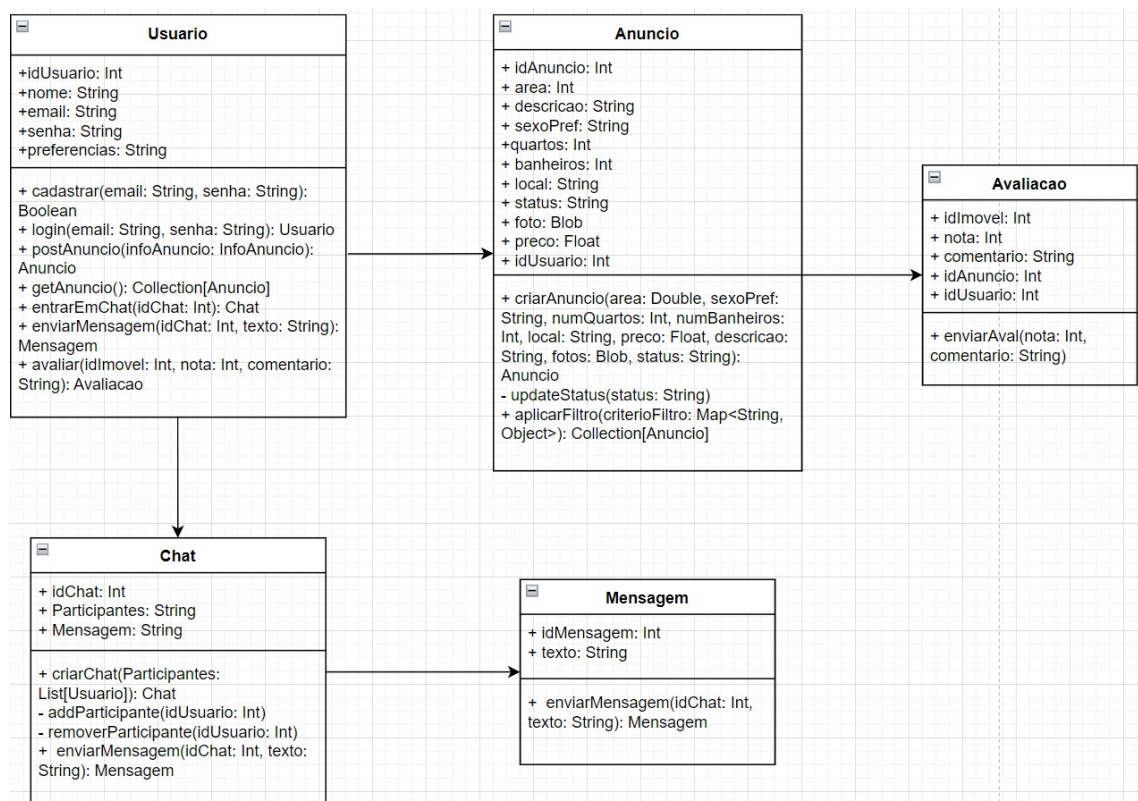


4.3 DIAGRAMAS UML

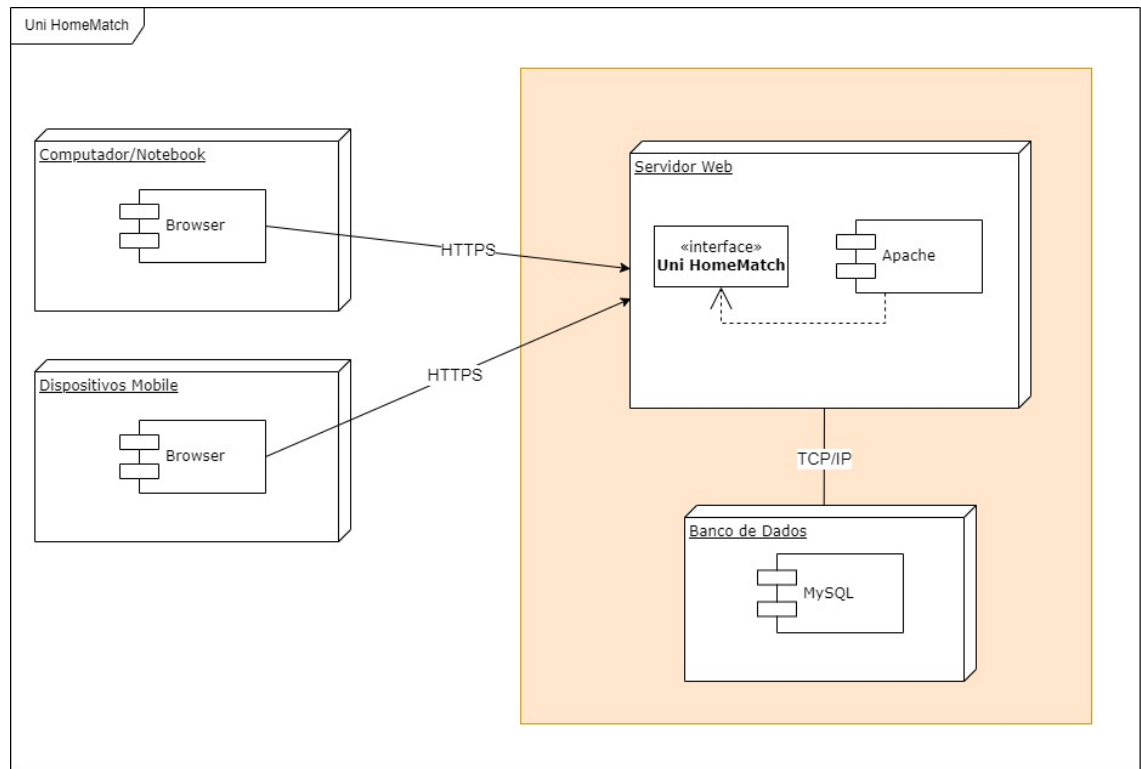
- DIAGRAMA DE CASO DE USO



- DIAGRAMA DE CLASSE



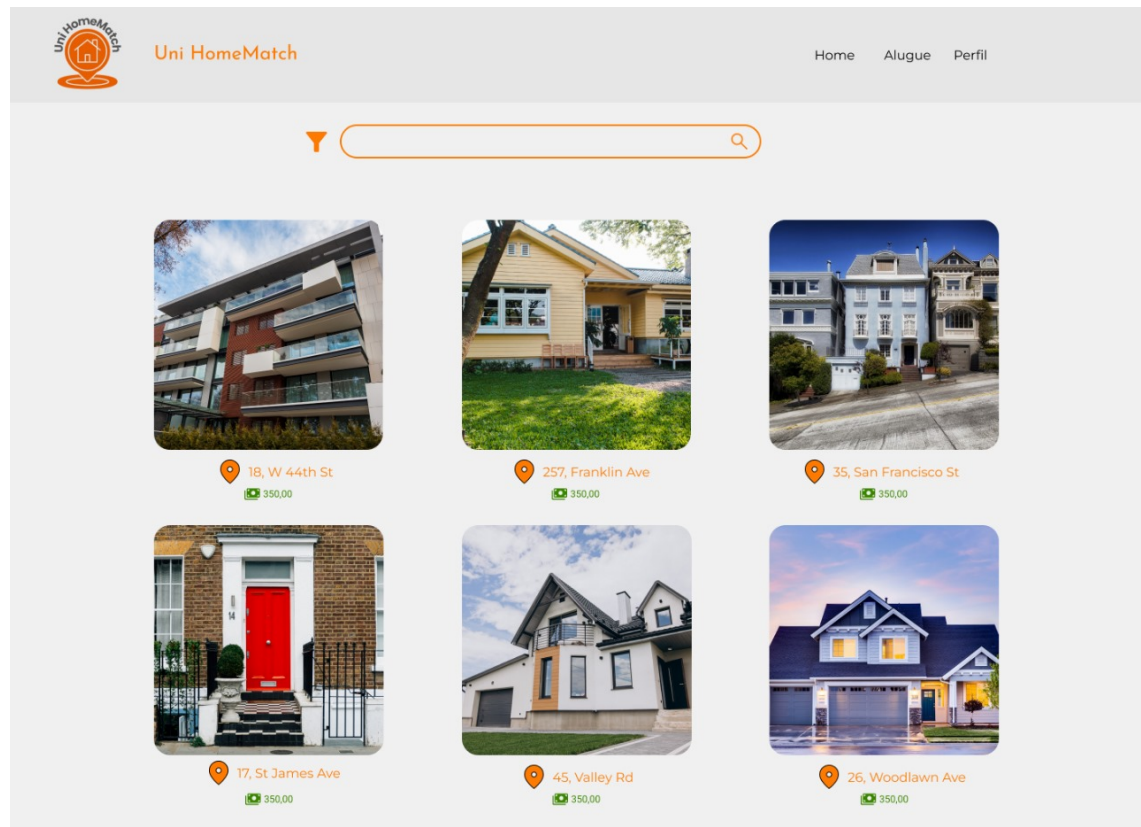
- DIAGRAMA DE IMPLANTAÇÃO



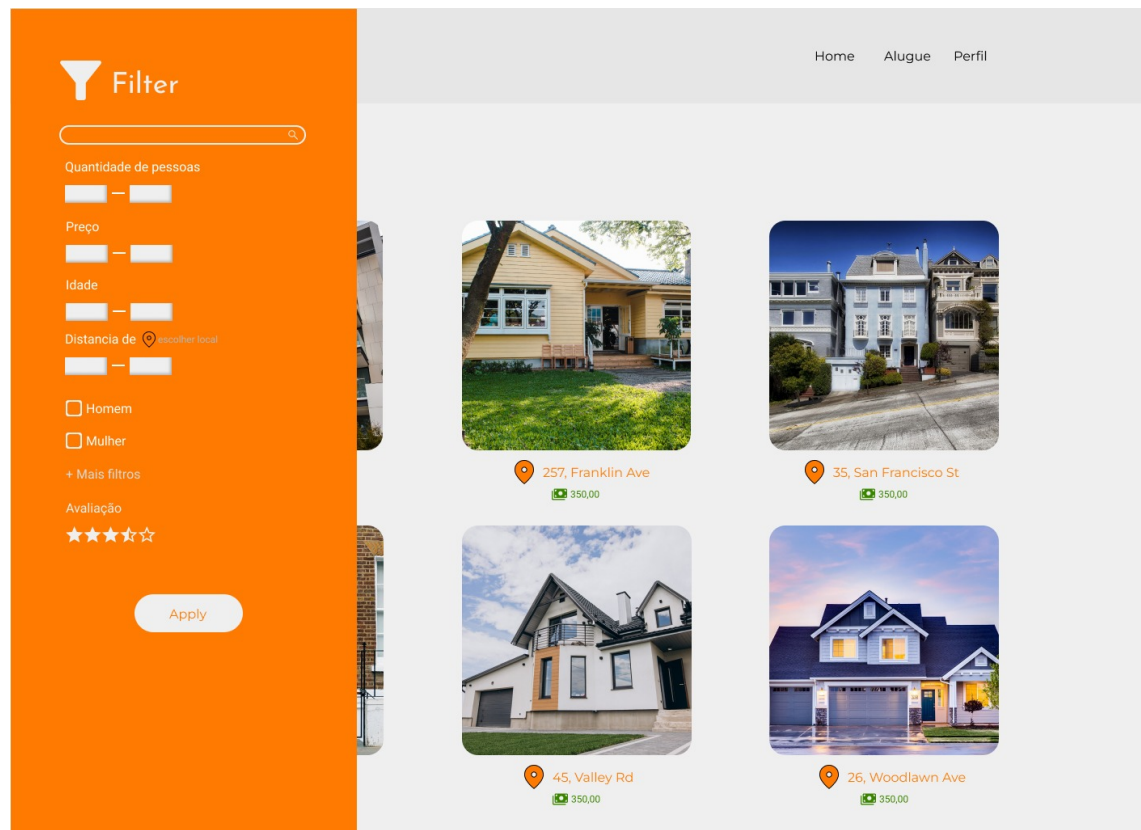
5 PROTÓTIPO

5.1 TELAS

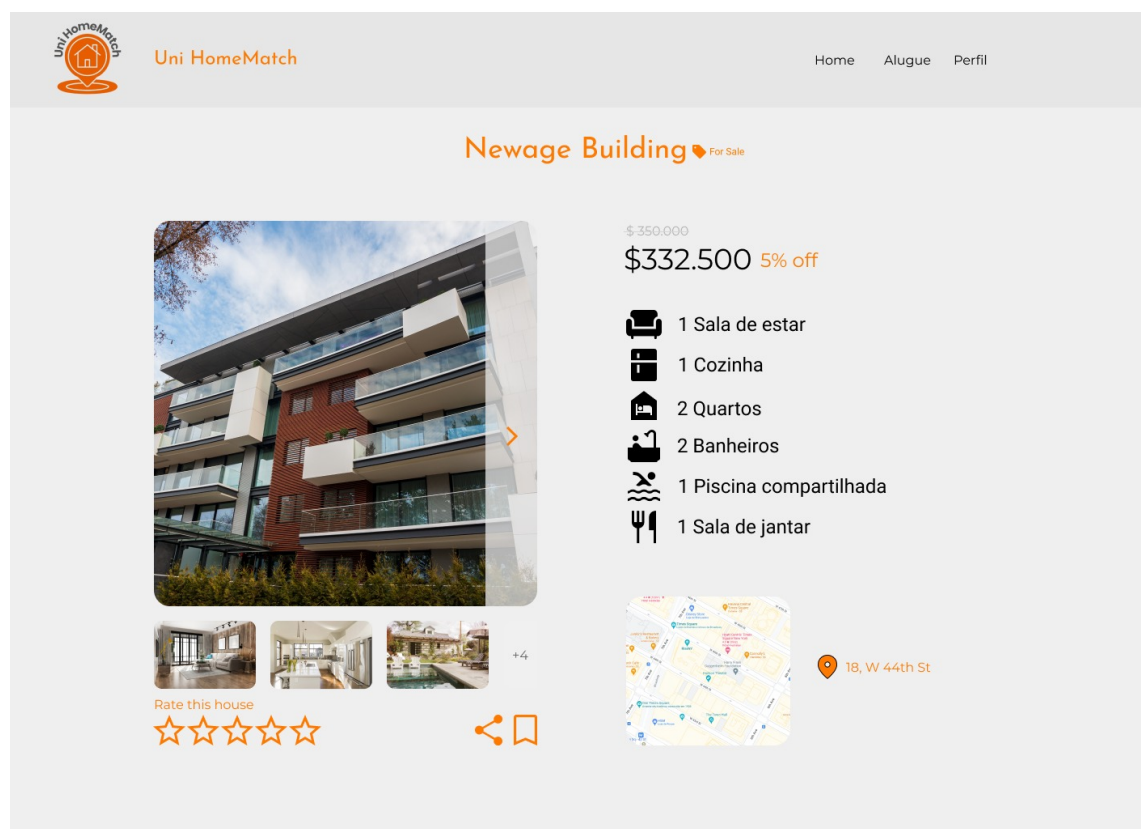
- TELA INICIAL



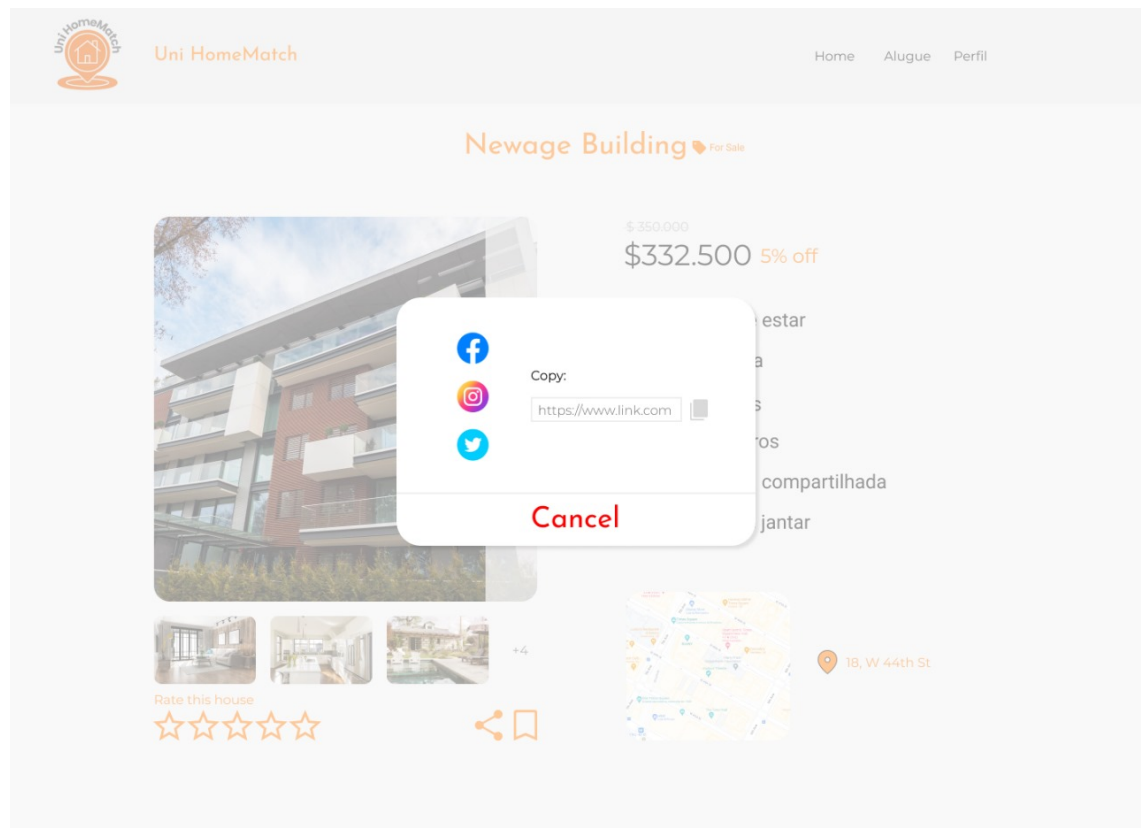
• FILTRO



• TELA DE ESPECIFICAÇÃO



- COMPARTILHAMENTO



• TELA DE PERFIL

