

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

**Economia Compartilhada - Aplicativo de Aluguel de
Itens - Alugaí**

João Roizen Fontana

RELATÓRIO DA DISCIPLINA DE PROJETO FINAL 2

CENTRO TÉCNICO CIENTÍFICO - CTC
DEPARTAMENTO DE INFORMÁTICA
Curso de Graduação em Ciência da Computação

Rio de Janeiro, Julho de 2024



João Roizen Fontana

Economia Compartilhada - Aplicativo de Aluguel de Itens - Alugaí

Relatório de Projeto Final 2, apresentado ao
Curso de Ciência da Computação da PUC-Rio
como requisito parcial para a obtenção do título
de Bacharel em Ciência da Computação.

Orientador: Edmundo Bastos Torreão

Departamento de Informática

Rio de Janeiro, Julho de 2024

Agradecimentos

Gostaria de expressar meus mais profundos agradecimentos a todos que me apoiaram e incentivaram ao longo desta jornada acadêmica. Em primeiro lugar, agradeço a D's por iluminar meu caminho e me dar força em todos os momentos.

Ao meu pai, Deoclécio Antônio Fontana, meu mais sincero agradecimento por todo o apoio, suporte e motivação oferecidos durante todos esses anos de graduação e sempre. Sua presença foi essencial para a minha trajetória.

Ao meu irmão, Arthur Roizen Fontana, agradeço pela constante disposição em me ajudar e apoiar. Sua colaboração foi de grande importância para a minha caminhada.

À minha namorada, Nicolle Goldbach, minha profunda gratidão por ser meu porto seguro, minha companheira e minha fonte de inspiração. Sua presença ao meu lado tornou esta jornada muito mais significativa.

Por fim, ao meu orientador, Edmundo Torreão, meu mais profundo reconhecimento pelo acompanhamento e ajuda inestimável durante todo este processo. Sua orientação foi crucial para a realização deste trabalho.

A todos vocês, o meu mais sincero muito obrigado.

Resumo

Fontana, João Roizen. Torreão, Edmundo Bastos. Alugaí, aplicativo de aluguel de itens. Rio de Janeiro, 2024. (número de páginas 60) Relatório de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este projeto apresenta o desenvolvimento de um aplicativo de aluguel de itens, inserido no contexto da economia compartilhada.

A aplicação propõe uma plataforma que facilita a conexão entre locadores e locatários, promovendo a interação social, redução de desperdícios e geração de renda extra.

Com foco na experiência do usuário, o sistema inclui funcionalidades como cadastro de itens, sistema de reputação, diversas modalidades de pagamento e opções de entrega/retirada.

As tecnologias utilizadas incluem Node.js, Express.js, MongoDB, React, entre outras, escolhidas pela sua eficiência, escalabilidade e suporte da comunidade.

Palavras-Chave: Economia compartilhada, Web, React

Abstract

Fontana, João Roizen. Torreão, Edmundo Bastos. Alugaí, item rental application. Rio de Janeiro, 2024. (number of pages 60) Final Project Report – Department of Informatics. Pontifical Catholic University of Rio de Janeiro.

This project details the development of an item rental application within the context of the sharing economy.

The application aims to create a platform that facilitates the connection between renters and tenants, promoting social interaction, waste reduction, and extra income generation.

Focusing on user experience, the system includes features such as item registration, reputation system, multiple payment methods, and delivery/pick-up options.

The technologies used include Node.js, Express.js, MongoDB, React, among others, chosen for their efficiency, scalability, and community support.

Keywords: Shared economy, Web, React

Sumário

1. Introdução.....	1
2. Situação atual.....	2
3. Proposta e Objetivos do trabalho.....	3
4. Tecnologias utilizadas.....	5
5. Plano de ação.....	7
6. Diagrama de Classes.....	8
7. Lista de Requisitos Funcionais.....	9
8. Diagrama de Caso de Uso.....	10
9. Descrição de Caso de Uso.....	11
10. Teste Realizados.....	42
11. Considerações Finais.....	55
12. Trabalhos Futuros.....	56
13. Referências Bibliográficas.....	58

1. Introdução

A economia compartilhada tem se tornado uma tendência crescente, revolucionando a forma como interagimos com bens e serviços. Nesse contexto, este trabalho de conclusão de curso visa a concepção e implementação de um aplicativo de aluguel de itens, que se enquadra nesse paradigma. Trata-se de uma plataforma inovadora que busca uma nova forma de interação entre locadores e locatários, ampliando a vida útil de bens pessoais e oferecendo uma oportunidade de renda extra de maneira mais sustentável e acessível. O objetivo é proporcionar uma experiência prática e eficiente para todos os usuários envolvidos.

A ideia central por trás deste aplicativo é criar um espaço onde locatários possam encontrar uma ampla variedade de itens categorizados, simplificando a busca por objetos desejados. O aplicativo busca transcender o mero ato de alugar itens, incorporando os princípios da economia compartilhada. Queremos proporcionar aos usuários a oportunidade de se conectar com outras pessoas, compartilhar, economizar recursos, além de oferecer a possibilidade de renda extra e contribuir para a redução do desperdício, promovendo uma abordagem mais sustentável e colaborativa.

Neste cenário de constante transformação digital e mudanças nos padrões de consumo, a criação desse aplicativo se revela uma resposta às demandas contemporâneas por conveniência, acessibilidade e interação social.

2. Situação atual

A economia compartilhada é o comércio com a promessa de conexão humana, onde indivíduos dividem, compartilham, uma parte de si além do comércio[1], assim define o cofundador do Airbnb Joe Gebbia, empresa que é pioneira e referência no setor. Nessa perspectiva, podemos perceber uma tendência global de conscientização em relação ao desperdício, da necessidade de uma maior interação social e um crescimento no número de indivíduos que buscam alguma forma de complementar a renda, uma pesquisa publicada pelo instituto Instituto Real Time Big Data mostrou que 48% dos entrevistados buscam renda extra para lidar com as despesas mensais[2].

Dessa maneira, segundo um outro estudo publicado pela Brookings Institution, a economia compartilhada inevitavelmente se tornará uma parte importante da economia global, estima-se que ela cresça de 14 bilhões de dólares em 2014 para 335 bilhões de dólares até 2025. Essa estimativa se baseia no rápido crescimento da Uber e Airbnb como indicativo[3].

No contexto atual, podemos notar que existe o conceito de se alugar itens, seja lojas de aluguel de roupas ou empresas de aluguel de carros, porém ainda é feito de uma maneira tradicional e descentralizada, onde para se alugar diferentes itens, o indivíduo precisa contratar o serviço de diferentes empresas, com diferentes regras. Ademais, pelo fato de ser, em sua grande maioria, um serviço contratado entre usuário e empresa, não se tem os benefícios da economia compartilhada, como a redução dos custos, uma maior variedade de produtos, mais flexibilidade, além dos proveitos citados anteriormente, como a redução do desperdício, a possibilidade de geração de renda extra e conexão entre as pessoas.

Assim, evidencia-se a necessidade de uma abordagem mais integrada, que adote os princípios "peer-to-peer" (de pessoa para pessoa), buscando consolidar todos os benefícios inerentes a esse novo modelo econômico e à transformação na interação social.

3. Proposta e Objetivos do trabalho

O objetivo deste trabalho é o desenvolvimento de uma aplicação de aluguel de itens que vai funcionar como um marketplace online que conecta diretamente os usuários que buscam anunciar ou alugar itens de uma maneira dinâmica, simples e confiável.

Para isso, o foco será na inovação na experiência de alugar, conectando os usuários e construindo confiança por meio da engenharia do projeto, uma vez que segundo o cofundador do Airbnb, Joe Gebbia, o fator mais importante na hora de alguém confiar um bem seu, sua residência no caso do Airbnb, a outra pessoa desconhecida é uma boa reputação[1].

Com isso, será implementado um sistema robusto de reputação, tanto para locadores como para locatários, onde será necessário dados pessoais para se fazer o cadastro, que vão ser mantidos em sigilo como CPF e reconhecimento facial para verificação da identidade. Em cada anúncio terá as fotos e descrição do produto, os dados do anunciante e dados referentes à análise deste usuário, como comentários e notas de outros usuários que já alugaram o item de um determinado locador, também será criado categorias pré definidas para avaliar, como se o locador é experiente, sua taxa de resposta e quanto tempo demora em média para responder e a opção de falar diretamente com o mesmo . Dessa maneira, também será desenvolvido um sistema de reputação equivalente para o locatário, com seus dados, notas e comentários de locadores que já alugaram para ele.

Outro requisito fundamental da aplicação é criar um processo direto para se anunciar e alugar um item, onde o locador consiga completar a operação em poucos passos e o locatário também consiga alugar de uma maneira dinâmica, o objetivo é que o usuário fique sempre a poucos clicks de alugar um item[4], também será implementado diversas modalidades de transações de pagamentos para que atenda às diferentes necessidades dos usuários.

Além disso, eventualmente será implementado diferentes formas de entrega/retirada de itens, onde os usuários poderão acordar entre si qual atende melhor suas necessidades. A primeira será a retirada no endereço fornecido pelo locador, a segunda será uma integração com motoristas de aplicativos para eles realizarem a retirada e entrega, semelhante como já ocorre nos delivery de comida,

podendo ser até mesmo uma integração com o uber flash que faz esse serviço de entrega de artigos pessoais, a terceira opção será o uso de “smart lockers” em ambientes públicos seguros, como em lojas de conveniência e shoppings, eles funcionam de maneira que assim que é entregue o cliente recebe um código de recolha via e-mail ou mensagens de texto e para resgatar a encomenda, é digitado o código em uma tela sensível ao toque e, a porta atribuída abre para a recuperação do pacote, o que permite soluções de autoatendimento totalmente automatizadas em todo o processo de utilização e torna toda a experiência rápida, eficiente e conveniente.

O projeto não propõe a interação com autorizadores de pagamentos.

Portanto, os requisitos listados serão implementados em uma aplicação web que seja responsiva e dinâmica.

4. Tecnologias Utilizadas

Durante a disciplina de Programação Web, INF 1407, meu interesse pelo desenvolvimento de aplicações web cresceu significativamente, motivando-me a aprofundar meus conhecimentos na área. A escolha das tecnologias e bibliotecas utilizadas no projeto foi fundamentada em critérios de eficiência, escalabilidade e suporte da comunidade.

Tecnologias Utilizadas e Justificativas:

1. Node.js e Express.js:

- **Justificativa:** A combinação de Node.js com Express.js é ideal para construir APIs robustas e escaláveis. Node.js permite o uso de JavaScript tanto no frontend quanto no backend, facilitando o desenvolvimento e manutenção do código. Express.js, sendo um framework minimalista e flexível, proporciona um desenvolvimento rápido e eficiente de servidores.[6][7][8]

2. MongoDB e Mongoose:

- **Justificativa:** Escolhi utilizar MongoDB no projeto por causa do meu interesse e curiosidade em trabalhar com bancos NoSQL e explorar suas funcionalidades e diferenças. Mongoose facilita a modelagem de dados e a interação com MongoDB, oferecendo uma estrutura mais organizada e controle sobre os dados.[9][10][11]

3. React:

- **Justificativa:** React é uma biblioteca JavaScript amplamente utilizada para construir interfaces de usuário interativas e responsivas. Sua abordagem baseada em componentes permite um desenvolvimento modular e reutilizável, além de uma manutenção mais simples do código.[12][13]

4. Redux:

- **Justificativa:** Para gerenciar o estado da aplicação de forma eficiente e previsível, Redux é a escolha ideal. Ele ajuda a centralizar o estado, facilitando o rastreamento de mudanças e a depuração do código.[14][15]

5. **Axios:**

- **Justificativa:** Axios é uma biblioteca de requisições HTTP que simplifica a comunicação com a API backend. Sua facilidade de uso e suporte a promessas tornam a gestão de chamadas assíncronas mais intuitiva e eficiente.[16][17]

6. **React Bootstrap:**

- **Justificativa:** Utilizar React Bootstrap facilita a criação de uma interface de usuário moderna e responsiva, aproveitando os componentes pré-estilizados do Bootstrap adaptados para React.[18]

7. **JWT (jsonwebtoken) e Bcryptjs:**

- **Justificativa:** Para garantir a segurança da aplicação, JSON Web Tokens (JWT) são utilizados para autenticação segura de usuários. Bcryptjs é usado para criptografar senhas, garantindo a proteção dos dados sensíveis dos usuários.[19][20][21]

8. **Nodemon:**

- **Justificativa:** Nodemon reinicia automaticamente o servidor ao detectar mudanças no código, melhorando a eficiência no processo de desenvolvimento.

9. **React Google Charts:**

- **Justificativa:** Dado que o projeto está sendo desenvolvido em React, react-google-charts se torna uma escolha natural devido à sua integração perfeita com a biblioteca, facilitando a implementação e manutenção do código.
- acelera o desenvolvimento, pois fornece componentes pré-construídos para gráficos, reduzindo a quantidade de código que precisa ser escrito e testado.[22][23]

10. **Cloudinary:**

- **Justificativa:** APIs robustas que simplificam o processo de upload, transformação e gerenciamento de mídia. Isso reduz a complexidade do código e facilita a manutenção e atualização do sistema.[24]

A escolha dessas tecnologias e bibliotecas foi guiada pela busca por uma solução robusta, escalável e de fácil manutenção, refletindo as melhores práticas de desenvolvimento web atuais. Este projeto não só consolidou meus conhecimentos adquiridos na disciplina, mas também me preparou para desafios reais no campo do desenvolvimento web.[25][26][27]

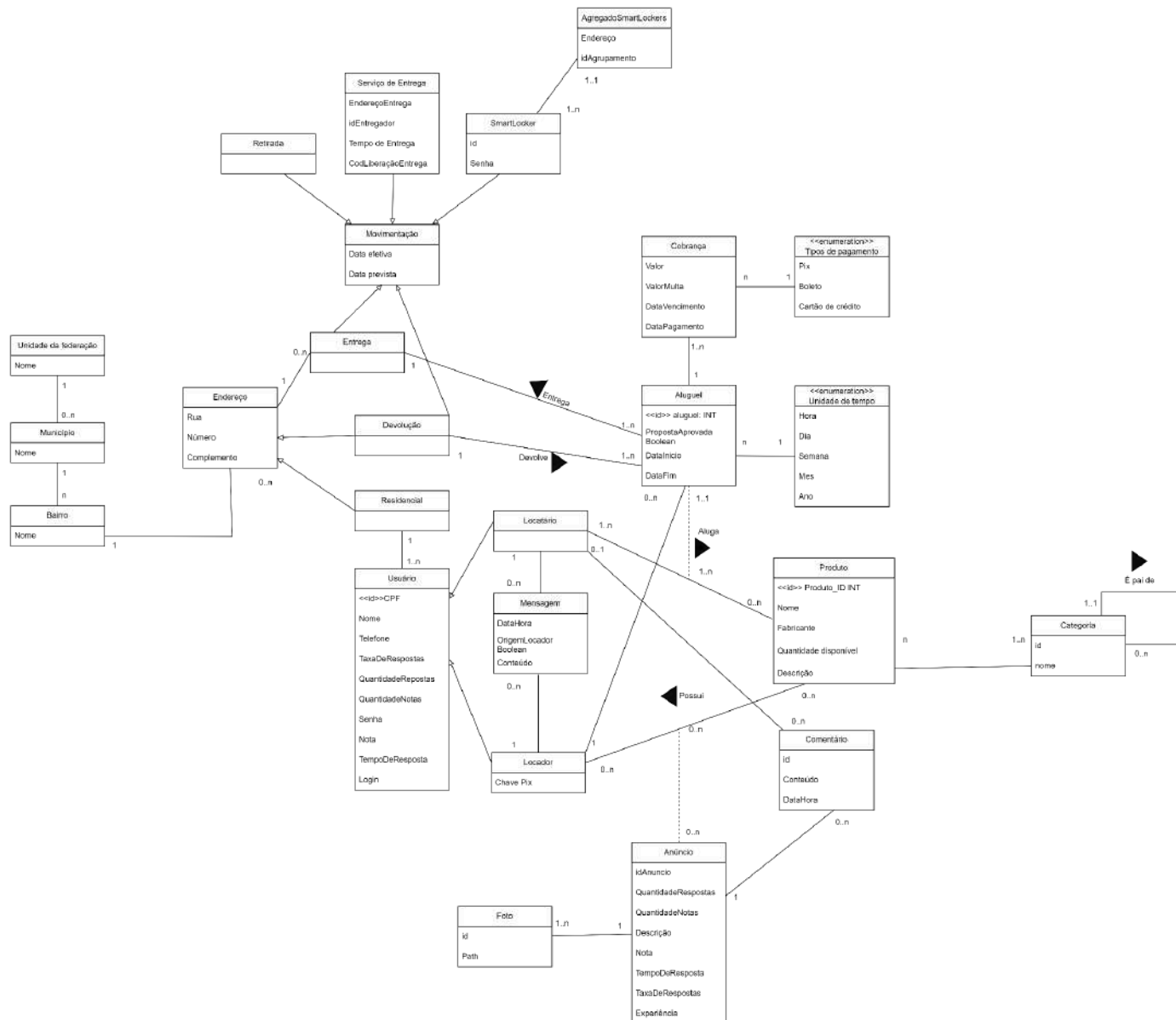
5. Plano de ação

Com o intuito de alcançar os objetivos delineados para garantir o progresso eficaz do projeto e o desenvolvimento do sistema, iniciaremos obtendo informações detalhadas e documentação de todos os requisitos associados ao software. Posteriormente, nossos esforços serão na modelagem de dados. Para evitar equívocos, realizaremos uma análise e avaliação cuidadosas das tecnologias e ferramentas mais viáveis para a implementação do projeto.

Foi elaborado um cronograma para orientar as atividades a serem realizadas ao longo do projeto.

[illegible]

6. Diagrama de Classes



7. Lista de Requisitos Funcionais

Usuário pode Cadastrar Usuário

Usuário pode Cadastrar, Alterar e Excluir Produto

Usuário pode Cadastrar, Alterar e Excluir Categoria do Produto

Usuário pode Cadastrar, Alterar e Excluir Anúncio

Usuário pode Cadastrar, Alterar e Excluir Foto

Usuário pode Cadastrar, Alterar e Excluir Avaliação

Usuário pode Cadastrar, Alterar e Excluir Comentário

Usuário pode Cadastrar, Alterar e Excluir Nota

Usuário pode Buscar Produtos

Usuário pode Buscar Comentários

Usuário pode Buscar Nota

Usuário pode Buscar Avaliação

Usuário pode Buscar Informação de Taxa de Resposta

Usuário pode Buscar Informação de Tempo de Resposta

Usuário pode Buscar Endereço de Retirada

Usuário pode Selecionar Serviço de Entrega

Usuário pode Selecionar Smart Locker

Usuário pode Enviar Mensagem

Usuário pode Receber Mensagem

Usuário pode Realizar Pagamento

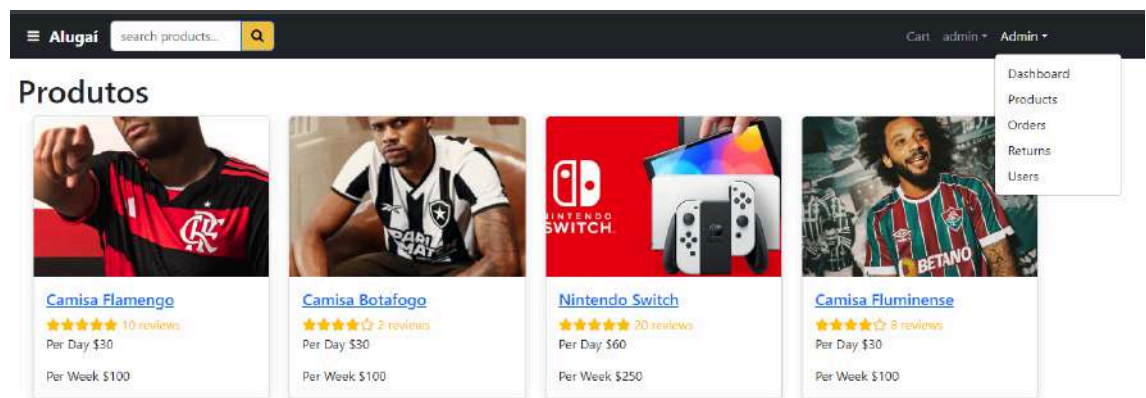
8. Diagrama de Caso de uso



9. Descrição de Caso de Uso

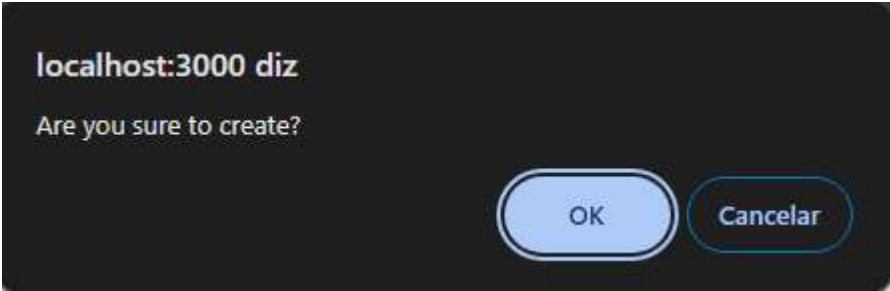
Obs: Os casos de uso são considerados os requisitos funcionais conforme [5]

Descrição de Caso de Uso UC01 – Cadastrar Produto



The screenshot shows the 'Products' page of the Alugai application. The header includes the Alugai logo, a search bar, and links for Cart, admin, and Admin. The main content area displays a table with columns: ID, NAME, PRICE, CATEGORY, BRAND, and ACTIONS. The table contains four rows of product data. A 'Create Product' button is visible in the top right corner.

ID	NAME	PRICE	CATEGORY	BRAND	ACTIONS
66915383f38ef19ad7b4b8b2	Camisa Flamengo		Esporte	Adidas	Edit Delete
66917e1e8cd6abfcd818ffc	Camisa Botafogo		Esporte	Reebok	Edit Delete
66917f708cd6abfcd81902a	Nintendo Switch		Video Game	Nintendo	Edit Delete
6693eeea38401377a22cb014	Camisa Fluminense		Esporte	Umbro	Edit Delete



Alugai

search products...

Carrinho Admin Admin

Product

6693f23438401377a22cb051

Name

Camisa Vasco

Slug

Camisa Vasco

Price Per Day

15

Price Per Week

70

Image File

https://res.cloudinary.com/d85wndme/image/upload/v1720971879/qfsgobdv

Upload File

Escolher arquivo

vasco.jpg

Category

Esporte

Brand

Kappa

Count In Stock

50

Price Per Day

15

Price Per Week

70

Image File

https://res.cloudinary.com/d85wndme/image/upload/v1720971879/qfsgobdv

Upload File

Escolher arquivo

vasco.jpg

Category

sample category

Brand

Kappa

Count In Stock

50

Description

Camisa do Clube Vasco da Gama ano 2024

Rating

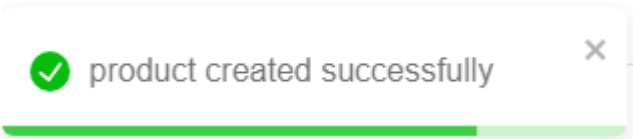
3.8

Number of Reviews

33

Create

All rights reserved



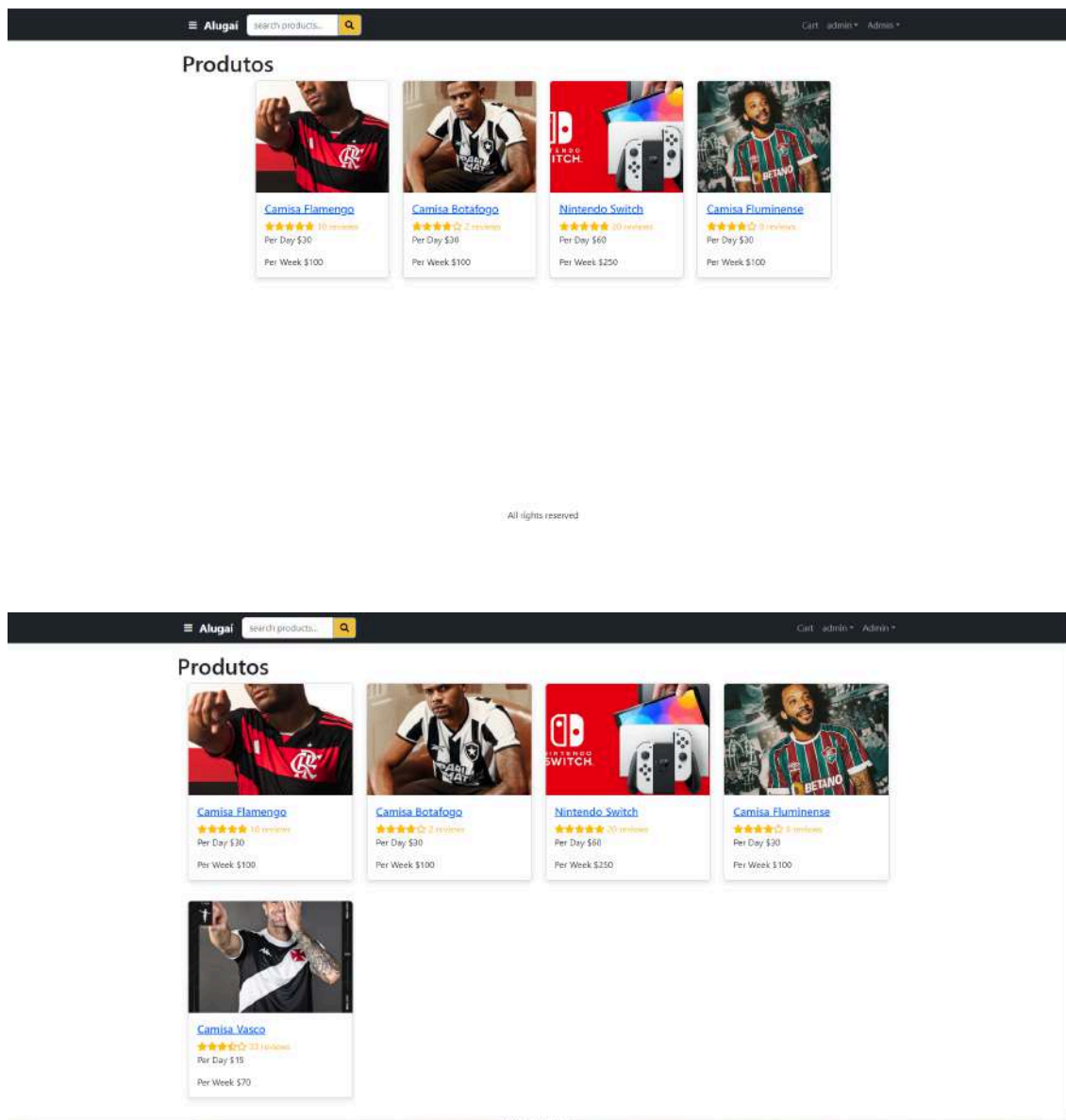
Descrição de Caso de Uso UC01 - Cadastrar Produto

Nome:	Cadastrar Produto
Objetivo:	Permitir que o administrador cadastre um novo produto na plataforma.
Atores:	Administrador
Pré-condição:	Administrador está autenticado na plataforma.
Pós-condição (cenário de sucesso):	Produto é cadastrado com sucesso na plataforma.
Pós-condição (cenário de insucesso):	Produto não é cadastrado na plataforma.
Trigger:	O administrador seleciona a opção 'Create Product' na tela de listagem de produtos.
Fluxo Principal:	1. O administrador acessa a tela de listagem de produtos e seleciona a opção 'Create Product'. 2. O sistema apresenta uma nova tela contendo os campos necessários para o cadastro do produto: 'Name', 'Slug', 'Price Per Day', 'Price Per Week', 'Image', 'Category', 'Brand', 'Count In Stock', 'Description', 'Rating', 'Number of Reviews'. 3. O administrador preenche os campos obrigatórios: 'Name', 'Slug', 'Price Per Day', 'Price Per Week', 'Image', 'Category', 'Brand', 'Count In Stock', 'Description', 'Rating', 'Number of Reviews'. 4. O administrador seleciona a opção 'Submit'. 5. O sistema valida os dados e salva as informações do produto. 6. O sistema exibe uma mensagem de sucesso informando que o produto foi cadastrado com êxito. 7. O sistema redireciona o administrador de volta à tela de listagem de produtos.
Fluxos Alternativos:	[A1] Administrador deseja cancelar o cadastramento: 1. O sistema cancela a operação de cadastro e retorna à tela de listagem de produtos.
Fluxos de Exceção	[E1] Campos obrigatórios não preenchidos: 1. O sistema apresenta uma mensagem de erro indicando quais campos precisam ser preenchidos. 2. O sistema retorna à tela de cadastro do produto. [E2] Dados inválidos: 1. O sistema identifica dados inválidos ou inconsistentes. 2. O sistema apresenta uma mensagem de erro indicando quais dados precisam ser corrigidos. 3. O sistema retorna à tela de cadastro do produto.

Pontos de Extensão:	
Regras de Negócio	[RN1] Os campos 'Name', 'Slug', 'Price Per Day', 'Price Per Week', 'Image', 'Category', 'Brand', 'Count In Stock', 'Description', 'Rating', 'Number of Reviews' são obrigatórios. [RN2] O preço por dia e o preço por semana devem ser valores válidos e maiores que zero. [RN3] É necessário fazer o upload de uma imagem do produto. [RN4] A categoria do produto deve ser selecionada a partir de uma lista pré-definida.

Descrição de Caso de Uso UC02– Cadastrar Anúncio

Acordado com o orientador, foi implementado o anúncio em conjunto com o cadastramento do produto, de forma que ao criar um produto novo, ele vai automaticamente para a Home Page do site.



Descrição de Caso de Uso UC02 - Cadastrar Anúncio

Nome:	Cadastrar Anúncio
Objetivo:	Permitir que o administrador cadastre um novo anúncio na plataforma.
Atores:	Administrador

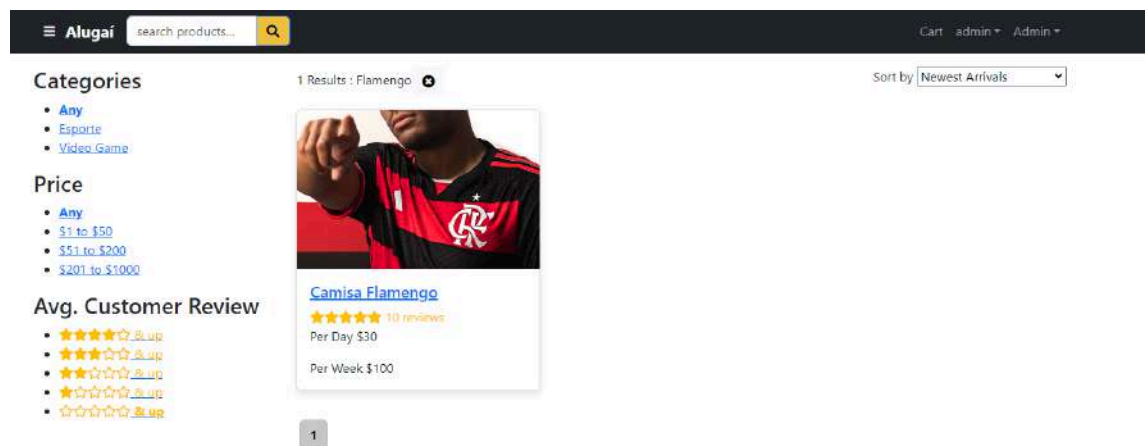
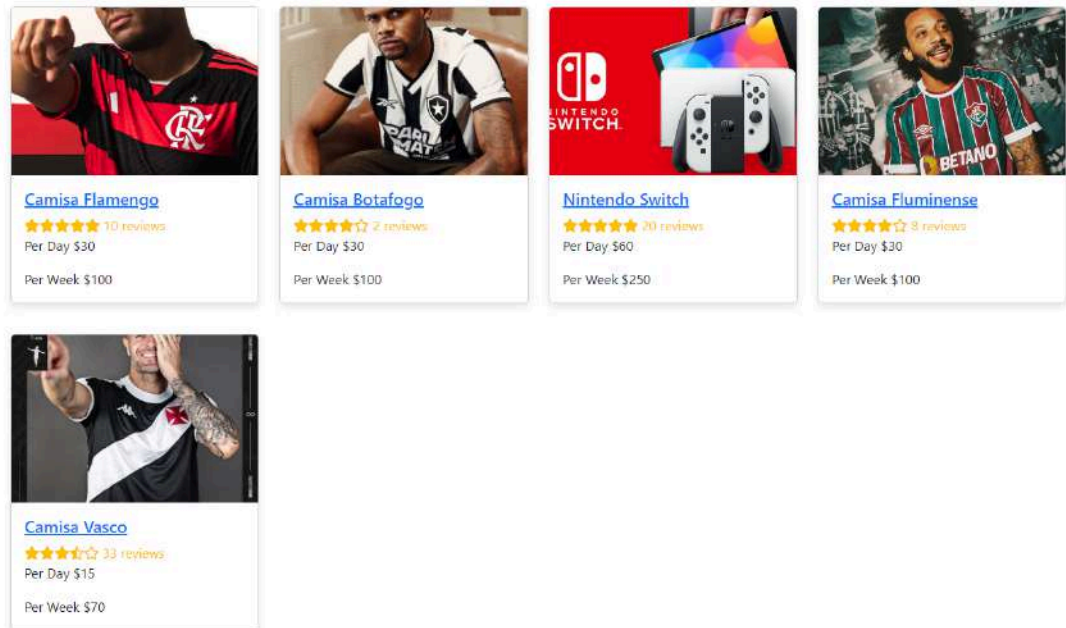
Pré-condição:	Administrador está autenticado na plataforma.
Pós-condição (cenário de sucesso):	O produto é cadastrado com sucesso na plataforma e aparece na listagem de produtos na tela inicial.
Pós-condição (cenário de insucesso):	Produto não é cadastrado na plataforma.
Trigger:	O administrador seleciona a opção 'Create Product' na tela de listagem de produtos.
Fluxo Principal:	1. O administrador acessa a tela de listagem de produtos e seleciona a opção 'Create Product'. 2. O sistema apresenta uma nova tela contendo os campos necessários para o cadastro do produto: 'Nome', 'Slug', 'Preço por Dia', 'Preço por Semana', 'Imagem', 'Categoria', 'Marca', 'Quantidade em Estoque', 'Descrição', 'Avaliação', 'Número de Avaliações'. 3. O administrador preenche os campos obrigatórios: 'Nome', 'Slug', 'Preço por Dia', 'Preço por Semana', 'Imagem', 'Categoria', 'Marca', 'Quantidade em Estoque', 'Descrição', 'Avaliação', 'Número de Avaliações'. 4. O administrador seleciona a opção 'Submit'. 5. O sistema valida os dados e salva as informações do produto. 6. O sistema exibe uma mensagem de sucesso informando que o produto foi cadastrado com êxito. 7. O sistema redireciona o administrador de volta à tela de listagem de produtos.
Fluxos Alternativos:	[A1] Administrador deseja corrigir os dados: 1. O sistema retorna ao passo 1 do Fluxo Principal. [A2] Administrador deseja cancelar o cadastramento: 2. O sistema cancela a operação de cadastro e retorna à tela de listagem de produtos.
Fluxos de Exceção	[E1] Campos obrigatórios não preenchidos: 1. O sistema apresenta uma mensagem de erro indicando quais campos precisam ser preenchidos. 2. O sistema retorna ao passo 1 do Fluxo Principal. [E2] Dados inválidos: 1. O sistema identifica dados inválidos ou inconsistentes. 2. O sistema apresenta uma mensagem de erro indicando quais dados precisam ser corrigidos. 3. O sistema retorna ao passo 1 do Fluxo Principal.
Pontos de Extensão:	

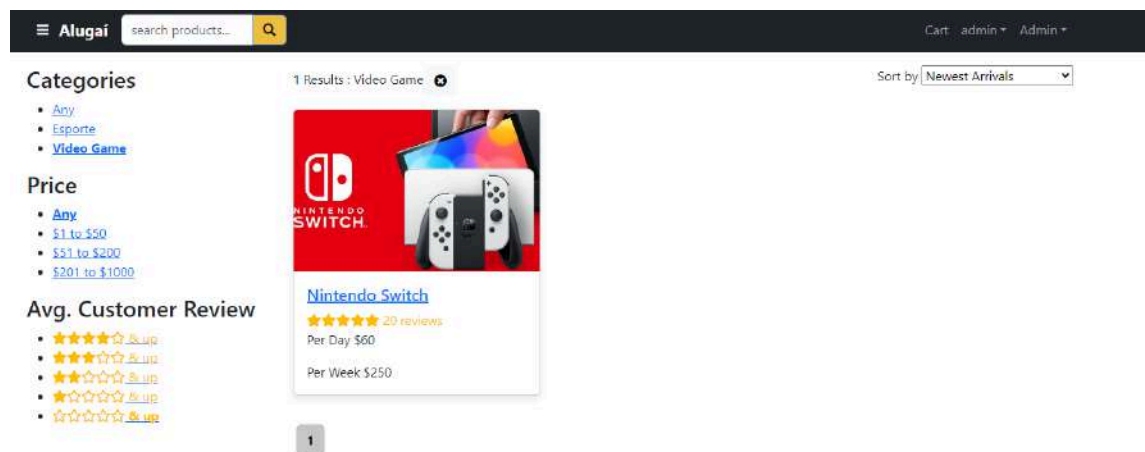
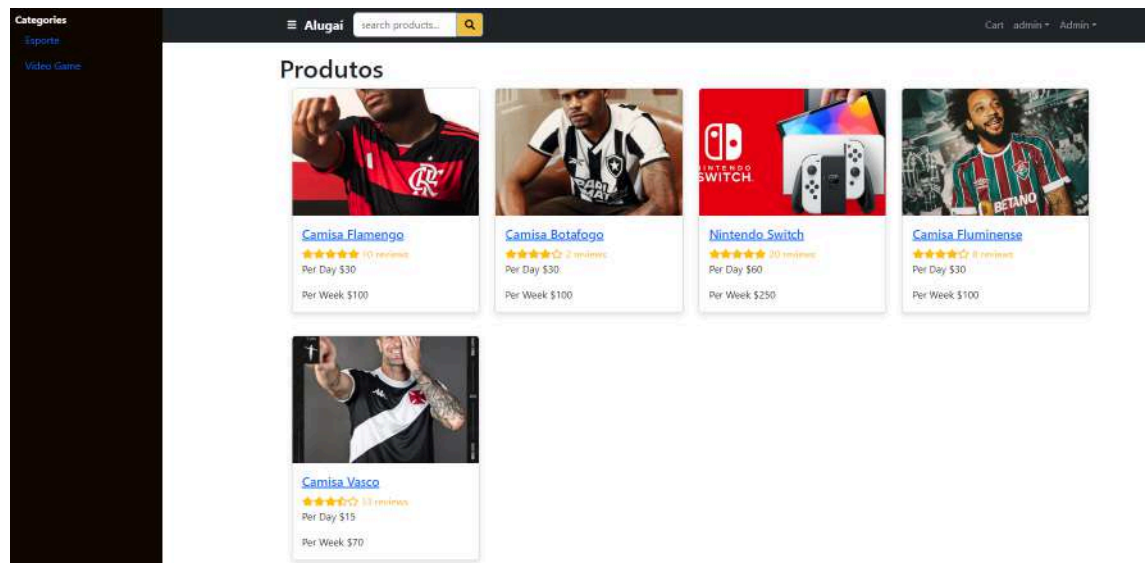
Regras de Negócio	[RN1] Os campos 'Nome', 'Slug', 'Preço por Dia', 'Preço por Semana', 'Imagem', 'Categoria', 'Marca', 'Quantidade em Estoque', 'Descrição', 'Avaliação', 'Número de Avaliações' são obrigatórios. [RN2] O preço por dia e o preço por semana devem ser valores válidos e maiores que zero. [RN3] É necessário fazer o upload de uma imagem do produto. [RN4] A categoria do produto deve ser selecionada a partir de uma lista pré-definida.
--------------------------	---

Descrição de Caso de Uso UC03 – Busca de Produto



Produtos



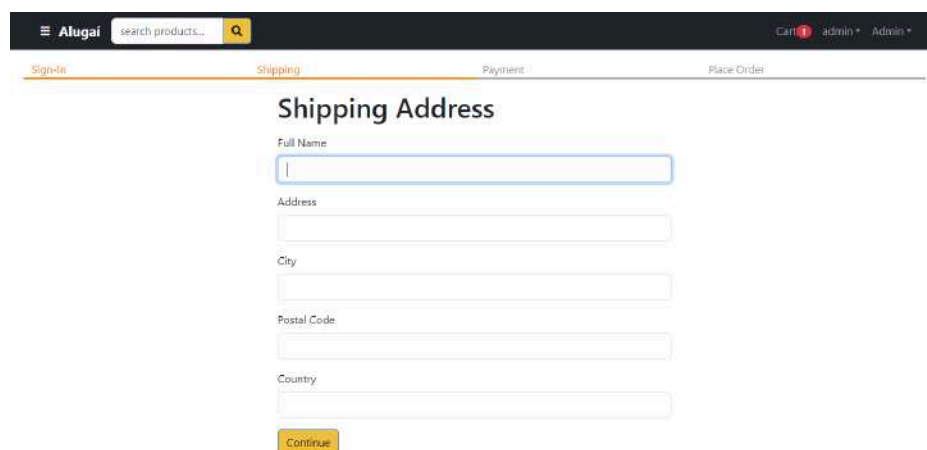
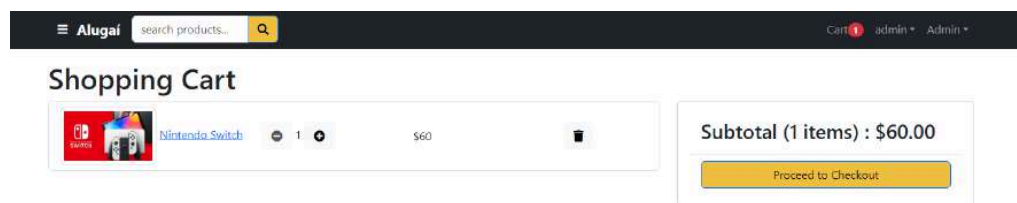
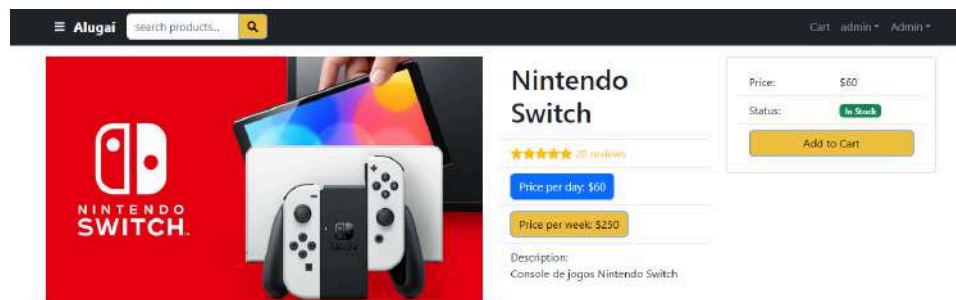


Descrição de Caso de Uso UC03 - Busca de Produto

Nome:	Busca de Produto
Objetivo:	Permitir que os usuários encontrem produtos disponíveis na plataforma.
Atores:	Locatário e Administrador
Pré-condição:	
Pós-condição (cenário de sucesso):	O usuário visualiza os produtos disponíveis de acordo com os critérios de busca.
Pós-condição (cenário de insucesso):	Mensagem de produto não encontrado é vista pelo usuário.
Trigger:	O usuário digita a palavra chave e aperta o botão de busca O usuário acessa o menu lateral e busca por categoria

Fluxo Principal:	1. O sistema apresenta uma interface de busca contendo um campo para 'Palavra-chave'. 2. O usuário preenche o campo de busca com a palavra-chave desejada. 3. O usuário aciona o botão de busca. 4. O sistema realiza a busca de produtos de acordo com a palavra-chave especificada. 5. O sistema exibe os resultados da busca ao usuário, mostrando os produtos disponíveis que correspondem à palavra-chave. 6. O usuário navega pelos resultados e visualiza informações detalhadas de cada produto. 7. O usuário seleciona um produto para obter mais informações ou proceder com o aluguel.
Fluxos Alternativos:	[A1] Busca por Categoria: 1. O usuário seleciona uma categoria específica na barra de busca 2. O sistema exibe uma lista de produtos pertencentes à categoria selecionada. [A2] Nenhum resultado encontrado: 1. O sistema não encontra produtos que correspondam à busca realizada. 2. O sistema exibe uma mensagem indicando que nenhum resultado foi encontrado.
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	[RN1] O campo 'Palavra-chave' permite a busca por termos específicos relacionados ao nome do produto. [RN2] Os resultados da busca são exibidos de forma clara e organizada, facilitando a visualização e seleção pelo usuário.

Descrição de Caso de Uso UC04 – Registro de Aluguel



Alugai

search products...

Q

Cart 1adminAdmin

Sign-InShippingPaymentPlace Order

Shipping Address

Full Name

admin

Address

rua

City

Rio de Janeiro

Postal Code

00000

Country

Brasil

Continue

Alugai

search products...

Q

Cart 1adminAdmin

Sign-InShippingPaymentPlace Order

Payment Method

☐ Credito

☒ Pix

Continue

Alugai

search products...

Q

Cart 1adminAdmin

Sign-InShippingPaymentPlace Order

Preview Order

Shipping

Name: admin
Address: rua,Rio de Janeiro, 00000,Brasil

Edit

Payment

Method: Pix

Edit

Items

 Nintendo Switch

1

\$60

Edit

Order Summary

Items	\$60.00
Shipping	\$10.00
Tax	\$9.00
Order Total	\$79.00

Place Order

Alugai

search products...

Q

Cart admin Admin

Order 6694046038401377a22cb0c4

Shipping

Name: admin
Address: rua,Rio de Janeiro, 00000,Brasil

Not Delivered

Payment

Method: Pix

Not Paid

Return

Name: admin

Not Returned

Items

	Nintendo Switch	1	\$60
---	-----------------	---	------

Order Summary

Items	\$60.00
Shipping	\$10.00
Tax	\$9.00
Order Total	\$79.00

Pay

Descrição de Caso de Uso UC04 - Registro de Aluguel

Nome:	Registro de Aluguel
Objetivo:	Permitir que um locatário registre o aluguel de produtos na plataforma.
Atores:	Locatário, Sistema
Pré-condição:	1. O locatário está autenticado na plataforma. 2. O carrinho de compras contém itens para aluguel.
Pós-condição (cenário de sucesso):	1. O aluguel é registrado com sucesso na plataforma. 2. O sistema atualiza o estado do pedido para refletir o aluguel. 3. O locatário é redirecionado para a página de detalhes do pedido.
Pós-condição (cenário de insucesso):	1. O aluguel não é registrado devido a problemas técnicos. 2. O sistema exibe uma mensagem de erro ao locatário.
Trigger:	O locatário confirma a finalização do aluguel na página de revisão do pedido.
Fluxo Principal:	1. O sistema exibe uma página de revisão do pedido com detalhes do aluguel. 2. O locatário confirma os detalhes do aluguel. 3. O sistema solicita informações adicionais, como endereço de entrega e método de pagamento. 4. O locatário fornece as informações solicitadas. 5. O sistema valida as informações fornecidas pelo locatário. 6. O sistema registra o aluguel na plataforma e reserva os produtos para o locatário.

	7. O sistema exibe uma mensagem de confirmação e redireciona o locatário para a página de detalhes do pedido.
Fluxos Alternativos:	[A1] Locatário cancela o registro do aluguel: 1. O sistema cancela a operação e retorna à página anterior. [A2] Informações fornecidas pelo locatário são inválidas ou incompletas: 1. O sistema exibe uma mensagem de erro indicando quais informações precisam ser corrigidas ou fornecidas. 2. O sistema retorna ao passo 3 do Fluxo Principal.
Fluxos de Exceção	[E1] Falha na validação das informações: 1. O sistema identifica uma falha na validação das informações fornecidas pelo locatário. 2. O sistema exibe uma mensagem de erro detalhando o problema. 3. O sistema cancela a operação e retorna à página anterior.
Pontos de Extensão:	
Regras de Negócio	[RN1] Todas as informações fornecidas pelo locatário devem ser válidas e completas para o registro do aluguel. [RN2] O sistema deve garantir que o produto esteja disponível para aluguel durante o período especificado pelo locatário. [RN3] Após o registro do aluguel, o sistema deve reservar o produto para o locatário e iniciar o processo de comunicação entre as partes envolvidas.

Descrição de Caso de Uso UC05 – Registro de Pagamento

Alugai

search products...

Q

Cart 1 admin Admin

Shopping Cart



Nintendo Switch

1

\$60

Subtotal (1 items) : \$60.00

Proceed to Checkout

Alugai

search products...

Q

Cart 1 admin Admin

Sign-In Shipping Payment Place Order

Shipping Address

Full Name

admin

Address

rua

City

Rio de Janeiro

Postal Code

00000

Country

Brasil

Continue

Alugai

search products...

Q

Cart 1 admin Admin

Sign-In Shipping Payment Place Order

Payment Method

☐ Credit

☒ Pix

Continue

Alugai

search products...

Q

Cart 1 admin Admin

Sign-InShippingPaymentPlace Order

Preview Order

Shipping

Name: admin
Address: rua,Rio de Janeiro, 00000,Brasil

[Edit](#)

Payment

Method: Pix

[Edit](#)

Items

	Nintendo Switch	1	\$60
---	-----------------	---	------

[Edit](#)

Order Summary

Items	\$60.00
Shipping	\$10.00
Tax	\$9.00
Order Total	\$79.00

[Place Order](#)

Alugai

search products...

Q

Cart admin Admin

Order 6694046038401377a22cb0c4

Shipping

Name: admin
Address: rua,Rio de Janeiro, 00000,Brasil

Not Delivered

Payment

Method: Pix

Not Paid

Return

Name: admin

Not Returned

Items

	Nintendo Switch	1	\$60
---	-----------------	---	------

Order Summary

Items	\$60.00
Shipping	\$10.00
Tax	\$9.00
Order Total	\$79.00

[Pay](#)

localhost:3000 diz

Order Paid

OK

Alugai

search products...

Q

CartadminAdmin

Order 6694046038401377a22cb0c4

Shipping

Name: admin

Address: rua,Rio de Janeiro, 00000,Brasil

Not Delivered

Payment

Method: Pix

Paid at 14/07/2024, 14:07:59

Return

Name: admin

Not Returned

Items



Nintendo Switch

1

\$60

Order Summary

Items

\$60.00

Shipping

\$10.00

Tax

\$9.00

Order Total

\$79.00

Deliver Order

Descrição de Caso de Uso UC05 - Registro de Pagamento

Nome:	Registro de Pagamento
Objetivo:	Permitir que o locatário efetue o pagamento do aluguel na plataforma.
Atores:	Locatário, Locador
Pré-condição:	1. O aluguel está registrado na plataforma. 2. O locatário está autenticado na plataforma. 3. O locatário forneceu informações de pagamento válidas.
Pós-condição (cenário de sucesso):	O pagamento é processado com sucesso e o locador é notificado sobre o recebimento do pagamento.
Pós-condição (cenário de insucesso):	O pagamento não é processado e o locatário é informado sobre a falha.
Trigger:	O locatário seleciona a opção de pagamento na página de detalhes do aluguel.
Fluxo Principal:	1. O sistema exibe uma tela de pagamento contendo o valor total a ser pago e as opções de pagamento disponíveis (Crédito, Pix). 2. O locatário seleciona o método de pagamento desejado. 3. O locatário fornece as informações necessárias para o pagamento. 4. O locatário confirma as informações de pagamento. 5. O sistema valida as informações fornecidas. 6. O sistema encaminha as informações de pagamento para o Sistema de Pagamento. 7. O Sistema de Pagamento processa o pagamento.

	8. O Sistema de Pagamento envia uma confirmação de pagamento para o sistema da plataforma. 9. O sistema registra o pagamento e atualiza o status do aluguel para "Pago". 10. O sistema envia uma notificação para o locador informando sobre o recebimento do pagamento.
Fluxos Alternativos:	[A1] Locatário cancela o processo de pagamento: 1. O sistema cancela a operação e retorna à página anterior. [A2] Informações fornecidas pelo locatário são inválidas ou incompletas: 1. O sistema exibe uma mensagem de erro indicando quais informações precisam ser corrigidas ou fornecidas. 2. O sistema retorna ao passo 3 do Fluxo Principal.
Fluxos de Exceção	[E1] Falha no processamento do pagamento: 1. O Sistema de Pagamento identifica uma falha no processamento do pagamento. 2. O Sistema de Pagamento envia uma notificação de falha para o sistema da plataforma. 3. O sistema exibe uma mensagem de erro informando ao locatário sobre a falha no processamento do pagamento. 4. O sistema cancela a operação e retorna à página anterior.
Pontos de Extensão:	
Regras de Negócio	[RN1] Todas as informações de pagamento fornecidas pelo locatário devem ser válidas e completas para o processamento do pagamento. [RN2] O sistema deve utilizar um método seguro e confiável para processar os pagamentos. [RN3] Após o processamento do pagamento, o sistema deve atualizar o status do aluguel para "Pago" e notificar o locador sobre o recebimento do pagamento.

Descrição de Caso de Uso UC06 – Devolução de Pedido

Alugai

search products...

Q

Cart admin Admin

Order 6694046038401377a22cb0c4

Shipping

Name: admin

Address: rua,Rio de Janeiro, 00000,Brasil

Not Delivered

Payment

Method: Pix

Paid at 14/07/2024, 14:07:59

Return

Name: admin

Not Returned

Items

	Nintendo Switch	1	\$60
---	-----------------	---	------

Order Summary

Items	\$60.00
Shipping	\$10.00
Tax	\$9.00
Order Total	\$79.00

Deliver Order

localhost:3000 diz

Order Delivered

OK

Alugai

search products...

CartadminAdmin

Returns

ID	USER	DATE	TOTAL	PAID	DELIVERED	RETURNED	DETAILS
668ebf1d17566df951d95685	DELETED USER	2024-07-10	\$113.50	No	No	No	Details
668ec1e817566df951d9568d	DELETED USER	2024-07-10	\$33.00	No	No	No	Details
668ec5bd470bee36a4c82e2	DELETED USER	2024-07-10	\$33.00	2024-07-10	No	No	Details
668edf1217967faf75883e3d	DELETED USER	2024-07-10	\$33.00	2024-07-10	No	No	Details
668ee05e17967faf75883e55	DELETED USER	2024-07-10	\$207.00	2024-07-10	No	No	Details
668ee8f7577a314d09abb348	DELETED USER	2024-07-10	\$33.00	No	No	No	Details
668eebbbad85bfd439e351de	DELETED USER	2024-07-10	\$79.00	No	No	No	Details
668eebc7ad85bfd439e351ea	DELETED USER	2024-07-10	\$113.50	No	No	No	Details
668eebd5ad85bfd439e351fb	DELETED USER	2024-07-10	\$207.00	2024-07-10	No	No	Details
668ef0d2ad85bfd439e3527a	DELETED USER	2024-07-10	\$207.00	2024-07-10	No	No	Details
668f30cead85bfd439e352f5	DELETED USER	2024-07-11	\$113.50	No	No	No	Details
668f9dccb85bfa5aa8a801	cr7	2024-07-11	\$207.00	2024-07-11	2024-07-12	No	Details
669139cdf38ef19ad7b4b892	admin	2024-07-12	\$125.00	2024-07-12	2024-07-12	No	Details
66916d22c7eb7dc414c27021	Gabriel Barbosa	2024-07-12	\$575.00	2024-07-12	2024-07-12	No	Details

Alugai

search products...

CartadminAdmin

Order 6694046038401377a22cb0c4

Shipping

Name: admin

Address: rua,Rio de Janeiro, 00000,Brasil

Delivered at 14/07/2024, 14:11:41

Payment

Method: Pix

Paid at 14/07/2024, 14:07:59

Return

Name: admin

Not Returned

Items



Nintendo Switch

1

\$60

Order Summary

Items\$60.00

Shipping\$10.00

Tax\$9.00

Order Total\$79.00

Return Order

localhost:3000 diz

Order Returned

OK

Alugai

search products...

Cart admin Admin

Order 6694046038401377a22cb0c4

Shipping

Name: admin

Address: rua.Rio de Janeiro, 00000,Brasil

Delivered at 14/07/2024, 14:11:41

Payment

Method: Pix

Paid at 14/07/2024, 14:07:59

Return

Name: admin

Returned at 14/07/2024, 14:15:02

Items



Nintendo Switch

1

\$60

Order Summary

Items	\$60.00
Shipping	\$10.00
Tax	\$9.00
Order Total	\$79.00

Descrição de Caso de Uso UC06 - Devolução de Pedido

Nome:	Devolução de Pedido
Objetivo:	Permitir que o usuário solicite a devolução de um pedido na plataforma.
Atores:	Locatário, Locador
Pré-condição:	1. O aluguel está em andamento. 2. O locatário está autenticado na plataforma. 3. O locatário já efetuou o pagamento. 4. O locatário já recebeu o produto
Pós-condição (cenário de sucesso):	A devolução é registrada com sucesso na plataforma e o locador é notificado sobre a solicitação.
Pós-condição (cenário de insucesso):	A devolução não é registrada e o locatário é informado sobre a falha.
Trigger:	O locatário seleciona a opção de solicitar devolução na página de detalhes do aluguel.
Fluxo Principal:	1. O sistema exibe uma tela contendo os detalhes do aluguel 2. O locatário confirma a solicitação de devolução. 3. O sistema registra a solicitação de devolução na plataforma. 4. O sistema envia uma notificação para o locador informando sobre a solicitação de devolução.

Fluxos Alternativos:	[A1] Locatário cancela a solicitação de devolução: 1. O sistema cancela a operação e retorna à página anterior.
Fluxos de Exceção	[E1] Falha no processamento da devolução: 1. O sistema identifica uma falha no processamento da devolução. 2. O sistema exibe uma mensagem de erro informando ao locatário sobre a falha. 3. O sistema cancela a operação e retorna à página anterior.
Pontos de Extensão:	
Regras de Negócio	[RN1] Após o registro da solicitação de devolução, o sistema deve notificar imediatamente o locador sobre a solicitação para que ele possa responder prontamente.

Descrição de Caso de Uso UC07 – Cadastramento de Smart Lockers Livres

Em acordo com o orientador, não foi implementada a utilização de smart lockers. Foi priorizado outros casos de uso por falta de tempo disponível nas disciplinas de Projeto Final para estudo mais detalhado de como ocorreria essa integração.

Descrição de Caso de Uso UC07 -Cadastramento de Smart Locker Livres

Nome:	Cadastramento de Smart Lockers Livres
Objetivo:	Permitir que um locador cadastre um conjunto de Smart Lockers como uma opção de entrega/retirada de itens na plataforma.
Atores:	Locador, Tempo
Pré-condição:	O locador de smart lockers está autenticado na plataforma. O aluguel pelo locador dos smart lockers foi confirmado.
Pós-condição (cenário de sucesso):	O conjunto de smart lockers é registrado para o aluguel
Pós-condição (cenário de insucesso):	A solicitação não é registrada e o locador é informado sobre a falha.
Trigger:	
Fluxo Principal:	1. Leitura do arquivo de origem com a totalidade de smart lockers livres 2. Verificação se existe alocação de algum smart locker livre[E1][RN1] 3. Registro na base de dados dos smart lockers livres
Fluxos Alternativos:	
Fluxos de Exceção	[E1] Existe algum smart locker dado como livre que encontra-se alocado por um usuário da plataforma 1. Essa situação é registrado em um log 2. Esse log vai ser analisado 3. A carga dos smart lockers é abortada.
Pontos de Extensão:	
Regras de Negócio	[RN1] Só é permitida a carga se a totalidade de smart lockers não tiver algum equipamento para algum usuário da plataforma

Descrição de Caso de Uso UC08 – Cadastrar Usuário

Alugai

search products...

Q

Cart Sign In

Produtos



[Camisa Flamengo](#)
★★★★★ 10 reviews
Per Day \$30
Per Week \$100



[Camisa Botafogo](#)
★★★★☆ 2 reviews
Per Day \$30
Per Week \$100



[Nintendo Switch](#)
★★★★★ 20 reviews
Per Day \$60
Per Week \$250



[Camisa Fluminense](#)
★★★★★ 8 reviews
Per Day \$30
Per Week \$100



[Camisa Vasco](#)
★★★★☆ 33 reviews
Per Day \$15
Per Week \$70

Alugai

search products...

Q

Cart Sign In

Email

Password

Sign In

New customer? [Create your account](#)

Alugai

search products...

Q

Cart Sign In

Sign Up

Name

Email

Password

Confirm Password

Sign Up

Already have an account? [Sign In](#)

Descrição de Caso de Uso UC08 - Cadastrar Usuário

Nome:	Cadastrar Usuário
Objetivo:	Permitir o cadastro de um novo usuário no sistema.
Atores:	Locatário
Pré-condição:	O usuário não deve estar logado no sistema.
Pós-condição (cenário de sucesso):	Usuário cadastrado no sistema.
Pós-condição (cenário de insucesso):	Usuário não cadastrado.
Trigger:	A opção 'Sign In' na tela principal é acionada.
Fluxo Principal:	1. O usuário seleciona a opção de "Sign In" na tela inicial 2. O usuário seleciona a opção de "Create you account" 3. O sistema apresenta um formulário de cadastro contendo os campos: • Nome • Email • Senha • Confirmar Senha 4. O usuário preenche os campos Nome, Email, Senha e Confirmar Senha 5. O usuário seleciona a opção 'Sign Up' para enviar o formulário. 6. O sistema valida os campos. 7. Se a validação for bem-sucedida, o sistema cria a conta do usuário e realiza o login automático. 8. O sistema redireciona o usuário para a página inicial ou para a URL de redirecionamento especificada.
Fluxos Alternativos:	[A1] Locatário cancela a solicitação de devolução: 1. O sistema cancela a operação e retorna à página anterior.

Fluxos de Exceção	[E1] Falha no processamento da devolução: 1. O sistema identifica uma falha no processamento da devolução. 2. O sistema exibe uma mensagem de erro informando ao locatário sobre a falha. 3. O sistema cancela a operação e retorna à página anterior.
Pontos de Extensão:	
Regras de Negócio	[RN1] O locatário deve fornecer informações adicionais sobre a solicitação de devolução para ajudar o locador a entender o motivo da devolução. [RN2] O sistema deve registrar a solicitação de devolução de forma clara e precisa para que o locador possa tomar as medidas necessárias. [RN3] Após o registro da solicitação de devolução, o sistema deve notificar imediatamente o locador sobre a solicitação para que ele possa responder prontamente.

Descrição de Caso de Uso UC09 – Adicionar e Remover Itens do Carrinho

Alugáí

search products...

Q

Cart Sign In

Produtos



[Camisa Flamengo](#)
★★★★★ 10 reviews
Per Day \$30
Per Week \$100



[Camisa Botafogo](#)
★★★★★ 2 reviews
Per Day \$30
Per Week \$100



[Nintendo Switch](#)
★★★★★ 20 reviews
Per Day \$60
Per Week \$250



[Camisa Fluminense](#)
★★★★★ 8 reviews
Per Day \$30
Per Week \$100



[Camisa Vasco](#)
★★★★★ 33 reviews
Per Day \$15
Per Week \$70

Alugáí

search products...

Q

Cart Sign In



Camisa Fluminense

★★★★★ 8 reviews

Price per day: \$30

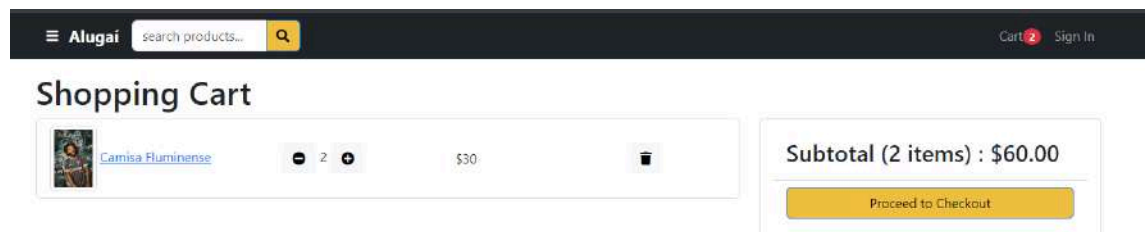
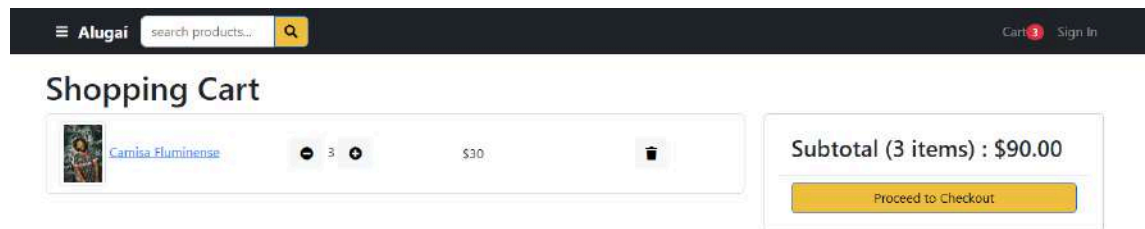
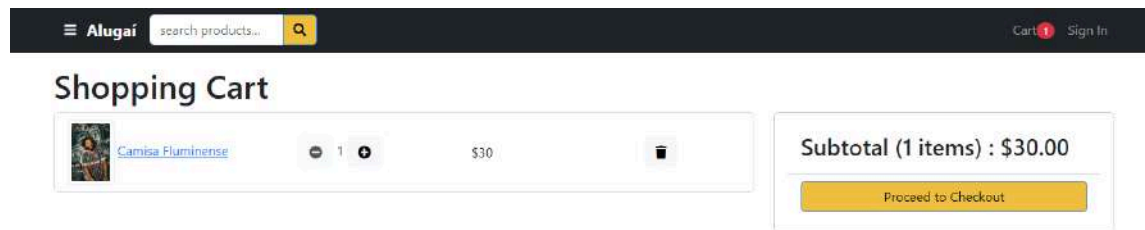
Price per week: \$100

Description:
Camisa do Fluminense ano 2024

Price: \$30

Status: In Stock

Add to Cart

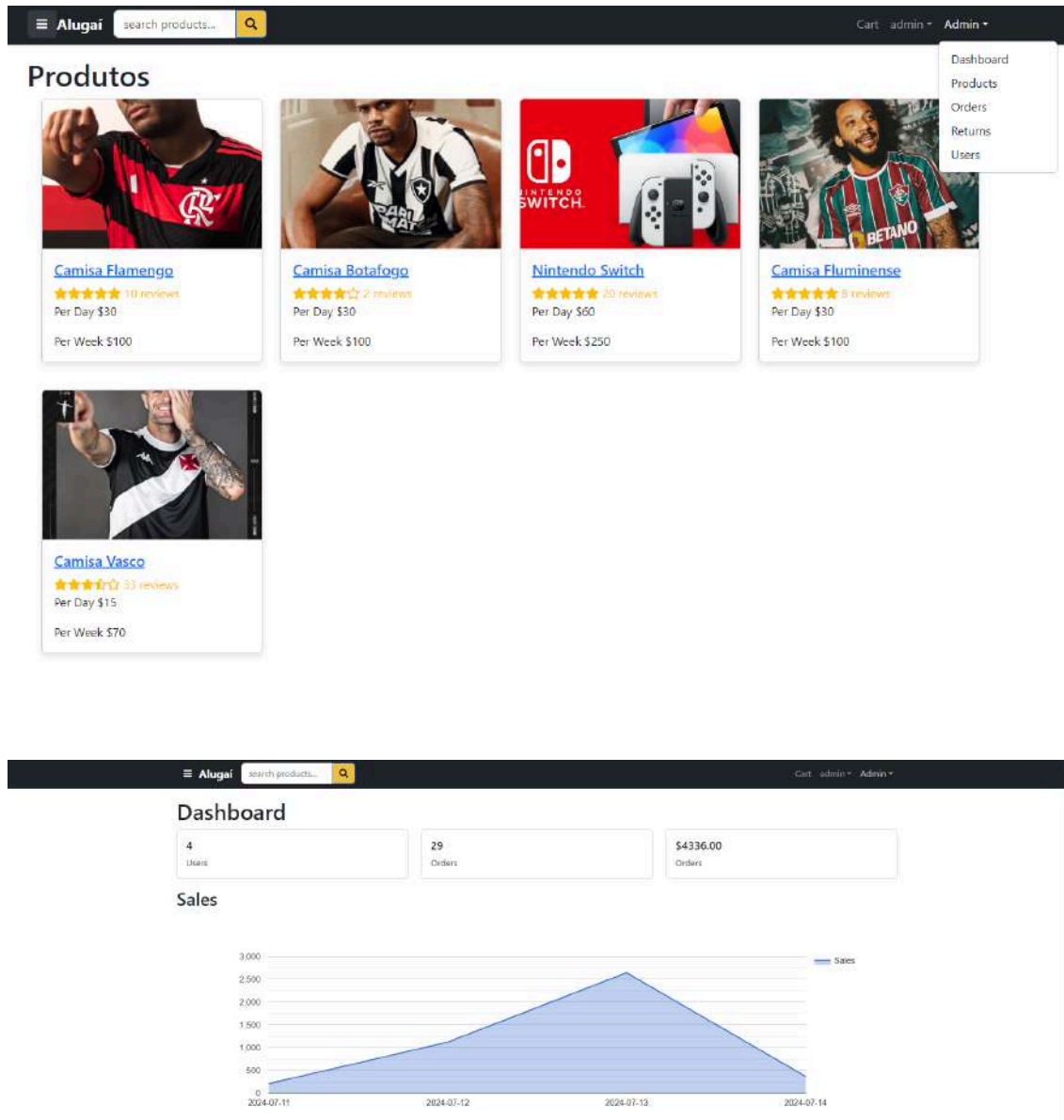


Descrição de Caso de Uso UC08 - Adicionar e Remover Itens do Carrinho

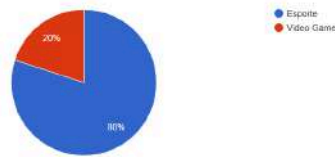
Nome:	Adicionar e Remover Itens do Carrinho
Objetivo:	Permitir que o usuário adicione e remova itens do carrinho de compras.
Atores:	Locatário
Pré-condição:	O usuário deve estar navegando no site, acessar a página de detalhes do produto e o produto deve estar disponível.
Pós-condição (cenário de sucesso):	Item adicionado ao carrinho. Item removido do carrinho. Quantidade de itens no carrinho atualizada.
Pós-condição (cenário de insucesso):	Nenhuma alteração no carrinho.
Trigger:	O usuário clica em "Add to Cart" na página do produto.

	O usuário clica em "+" ou "-" para alterar a quantidade de um item no carrinho. O usuário clica no ícone de lixeira para remover um item do carrinho.
Fluxo Principal:	1. O usuário navega até a página de detalhes do produto. 2. O sistema apresenta as informações do produto, incluindo opções de preço por dia e por semana. 3. O usuário seleciona a opção de preço desejada. 4. O usuário clica no botão "Add to Cart". 5. O sistema adiciona o item ao carrinho e redireciona o usuário para a página do carrinho. 6. O sistema exibe a opção de confirmação.
Fluxos Alternativos:	[A1] Remover Itens do Carrinho: 1. O usuário navega até a página do carrinho de compras. 2. O sistema apresenta a lista de itens no carrinho. 3. O usuário clica no ícone de lixeira ao lado do item que deseja remover. 4. O sistema remove o item do carrinho. 5. O sistema atualiza a lista de itens no carrinho e recalcula o subtotal. [A2] Atualizar Quantidade de Itens no Carrinho: 1. O usuário navega até a página do carrinho de compras. 2. O sistema apresenta a lista de itens no carrinho. 3. O usuário clica nos botões "+" ou "-" para alterar a quantidade de um item. 4. O sistema verifica a disponibilidade do item no estoque. 5. Se o item estiver disponível, o sistema atualiza a quantidade do item no carrinho e recalcula o subtotal. 6. O sistema exibe a quantidade atualizada e o novo subtotal.
Fluxos de Exceção	[E1] Ocorre um erro ao tentar adicionar um item ao carrinho: 1. O sistema apresenta uma mensagem "Erro ao tentar adicionar ao carrinho". 2. O sistema retorna para o passo 4 do Fluxo Principal para Adicionar Itens ao carrinho.
Pontos de Extensão:	
Regras de Negócio	[RN1] O sistema deve verificar a disponibilidade de estoque antes de adicionar ou atualizar a quantidade de um item no carrinho. [RN2] O sistema deve recalcular o subtotal do carrinho sempre que um item for adicionado, removido ou sua quantidade for alterada. [RN3] O sistema deve permitir apenas a adição de itens com uma opção de preço selecionada.

Descrição de Caso de Uso UC010 – Dashboard do Administrador(locador)



Categories



All rights reserved

Descrição de Caso de Uso UC010 - Dashboard do Administrador

Nome:	Visualizar Dashboard do Administrador
Objetivo:	Permitir que o administrador visualize o resumo de usuários, pedidos e vendas, além de gráficos de vendas diárias e categorias de produtos.
Atores:	Administrador(locador)
Pré-condição:	O administrador deve estar logado no sistema.
Pós-condição (cenário de sucesso):	Resumo de usuários, pedidos e vendas é exibido. Gráficos de vendas diárias e categorias de produtos são exibidos.
Pós-condição (cenário de insucesso):	Uma mensagem de erro é exibida se os dados não puderem ser carregados.
Trigger:	O administrador navega o menu de admin e acessar a opção de dashboard
Fluxo Principal:	O administrador navega para a página de dashboard. O sistema exibe uma mensagem de carregamento enquanto os dados são obtidos. O sistema recebe e processa os dados de resumo de usuários, pedidos e vendas. O sistema exibe: • O número total de usuários. • O número total de pedidos. • O total de vendas. O sistema exibe um gráfico de vendas diárias. O sistema exibe um gráfico de categorias de produtos.
Fluxos Alternativos:	[A1] Ocorre um erro ao carregar os dados: 1. O sistema exibe uma mensagem de erro ao administrador.
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	[RN1] O sistema deve autenticar o administrador antes de permitir o acesso ao dashboard.

10. Testes Realizados

- **Objetivos do teste Cadastrar Produto**
 - **Validação de Dados:** Verificar se todos os campos obrigatórios (nome, descrição, preço, categoria, imagem, etc.) são preenchidos corretamente e se os dados seguem o formato esperado.
 - **Autorização:** Garantir que apenas usuários com privilégios de administrador podem cadastrar produtos.
 - **Persistência de Dados:** Assegurar que os dados do produto sejam salvos corretamente no banco de dados.
 - **Interface de Usuário:** Confirmar que a interface do usuário reflete o novo produto corretamente na listagem de produtos e nas buscas.

Banco de Dados de Produtos Antes do Cadastro

The screenshot shows the MongoDB Compass interface for the 'test.products' collection. The sidebar on the left lists databases: 'olugoi', 'sample_mflix', 'test', 'orders', 'products' (selected), and 'users'. The top bar displays statistics for the 'test.products' collection: STORAGE SIZE 84KB, LOGICAL DATA SIZE 169KB, TOTAL DOCUMENTS 5, and INDEXES TOTAL SIZE 104KB. Below the statistics, there are tabs for 'Find', 'Indexes', 'Schema', 'Anti-Patterns', 'Aggregation', and 'Search Indexes'. A search bar is present with the placeholder 'Type a query: { field: 'value' }'. The main area shows the 'QUERY RESULTS: 1-5 OF 5' and displays a single document in JSON format:

```
{
  "_id": "ObjectID('66919387f386f50a8794b8b21')",
  "name": "Camisa Flamengo",
  "slug": "Camisa Flamengo",
  "image": "https://www.cloudinary.com/dff5wckw/image/upload/v3728811106/j3u08va...",
  "brand": "Adidas",
  "category": "Esportes",
  "description": "Camisa do clube de regatas do Flamengo ano 2024",
  "priceDay": 30,
  "priceWeek": 168,
  "countInStock": 1000,
  "rating": 5,
  "numberReviews": 38,
  "createdAt": "2024-07-12T16:02:11.947+00:00",
  "updatedAt": "2024-07-12T14:12:16.136+00:00",
  "__v": 0
}
```


Tela de Cadastro de Produto

≡ Alugai

search products...

Q

Cart admin Admin

Product

669484dd38401377a22cb6a1

Name

PlayStation 5

Slug

PlayStation 5

Price Per Day

30

Price Per Week

100

Image File

https://res.cloudinary.com/dfi5wndme/image/upload/v1721009373/prbezrxnp

Upload File

Escolher arquivo ps5.jpg

Category

Video Game

Brand

Sony

Count In Stock

50

30

Price Per Week

100

Image File

https://res.cloudinary.com/dfi5wndme/image/upload/v1721009373/prbezrxnp

Upload File

Escolher arquivo ps5.jpg

Category

Video Game

Brand

Sony

Count In Stock

50

Description

Console de jogos PlayStation 5

Rating

4.5

Number of Reviews

40

Update

All rights reserved

Banco de Dados de Produtos Após o Cadastro de Produto

test.products

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 2.26KB TOTAL DOCU

[Find](#) [Indexes](#) [Schema Anti-Patterns](#) 0

[Generate queries from natural language in Compass](#)

[Filter](#) Type a query: { field: 'value' }

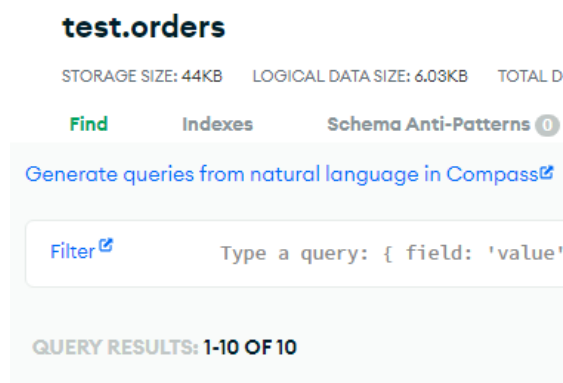
QUERY RESULTS: 1-6 OF 6

```
_id: ObjectId('669484dd38401377a22cb6a1')
name: "PlayStation 5"
slug: "PlayStation 5"
image: "https://res.cloudinary.com/dfi5wndme/image/upload/v1721009373/prbezrxn..."
brand: "Sony"
category: "Video Game"
description: "Console de jogos PlayStation 5"
priceDay: 30
priceWeek: 100
countInStock: 50
rating: 4.5
numReviews: 40
createdAt: 2024-07-15T02:09:33.411+00:00
updatedAt: 2024-07-15T02:10:19.225+00:00
__v: 0
```

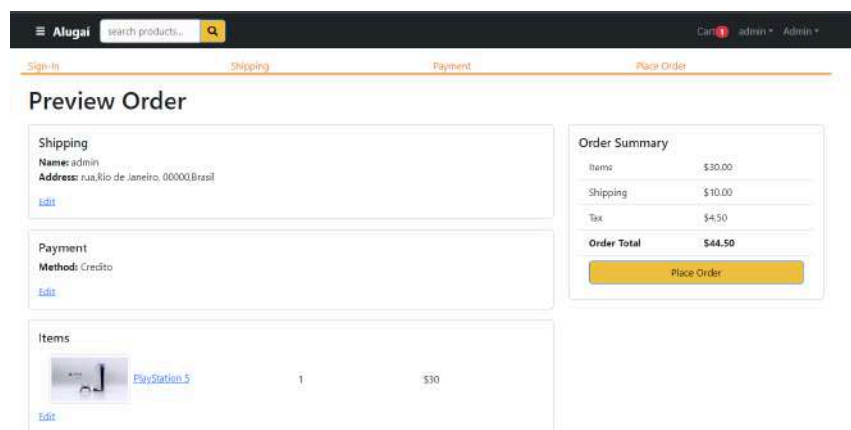
- **Objetivos do teste Registro de Aluguel**

- **Fluxo de Aluguel:** Testar o processo completo de aluguel de produtos, garantindo que os itens podem ser alugados e que o sistema mantém o estado dos itens alugados.
- **Persistência de Dados:** Garantir que os dados de aluguel são registrados corretamente no banco de dados.
- **Notificações:** Certificar que os usuários recebem confirmações de aluguel.

Banco de Dados Antes do Registro de Aluguel



Tela de Registro de Aluguel



Alugai

search products...

Cart admin Admin

Order 669487c538401377a22cb93f

Shipping

Name: admin

Address: rua,Rio de Janeiro, 00000,Brasil

Not Delivered

Payment

Method: Credito

Not Paid

Return

Name: admin

Not Returned

Items

	PlayStation 5	1	\$30
---	---------------	---	------

Order Summary

Items	\$30.00
Shipping	\$10.00
Tax	\$4.50
Order Total	\$44.50

Pay Order

Banco de Dados de Ordens Após o Registro de Aluguel

test.orders

STORAGE SIZE: 44KB LOGICAL DATA SIZE: 6.03KB TOTAL DOCUMENTS: 11

Find Indexes Schema Anti-Patterns 0

Generate queries from natural language in Compass

Filter Type a query: { field: 'value' }

QUERY RESULTS: 1-11 OF 11

```

{
  "_id": ObjectId('669487c538401377a22cb93f'),
  "orderItems": Array (1)
  "shippingAddress": Object
    "paymentMethod": "Credito"
    "itemsPrice": 30
    "shippingPrice": 10
    "taxPrice": 4.5
    "totalPrice": 44.5
  "user": ObjectId('668fcd6fbdf835140b387eaf')
  "isPaid": false
  "isDelivered": false
  "isReturned": false
  "createdAt": 2024-07-15T02:21:57.311+00:00
  "updatedAt": 2024-07-15T02:21:57.311+00:00
  "__v": 0
}

```

- **Objetivos do teste Registro de Pagamento**
 - **Atualização de Status:** Garantir que o status dos pedidos é atualizado corretamente após o pagamento ser confirmado.
 - **Persistência de Dados:** Confirmar que os registros de pagamento são armazenados corretamente no banco de dados para fins de auditoria e consulta futura.

Banco de Dados Antes do Pagamento

```

{
  _id: ObjectId('669487c538401377a22cb93f')
  orderItems: Array (1)
  shippingAddress: Object
    paymentMethod: "Credito"
    itemsPrice: 30
    shippingPrice: 10
    taxPrice: 4.5
    totalPrice: 44.5
  user: ObjectId('668fcd6fbdf835140b387eaf')
  isPaid: false
  isDelivered: false
  isReturned: false
  createdAt: 2024-07-15T02:21:57.311+00:00
  updatedAt: 2024-07-15T02:21:57.311+00:00
  __v: 0
}

```

Tela de Registro de Pagamento

Alugai

search products...

q

Cart admin Admin

Order 669487c538401377a22cb93f

Shipping

Name: admin

Address: rua, Rio de Janeiro, 00000, Brazil

Not Delivered

Payment

Method: Credito

Paid at 14/07/2024 23:30:31

Return

Name: admin

Not Returned

Items



Plexiglass 5

1

\$30

Order Summary

Items	\$30.00
Shipping	\$10.00
Tax	\$4.50
Order Total	\$44.50

Deliver Order

Banco de Dados Após o Registro de Pagamento

```
_id: ObjectId('669487c538401377a22cb93f')
▶ orderItems : Array (1)
▶ shippingAddress : Object
  paymentMethod : "Credito"
  itemsPrice : 30
  shippingPrice : 10
  taxPrice : 4.5
  totalPrice : 44.5
  user : ObjectId('668fcd6fbdf835140b387eaf')
  isPaid : true
  isDelivered : false
  isReturned : false
  createdAt : 2024-07-15T02:21:57.311+00:00
  updatedAt : 2024-07-15T02:30:31.103+00:00
  __v : 0
  paidAt : 2024-07-15T02:30:31.102+00:00
```

- **Objetivos do teste Registro de Entrega**
 - **Fluxo de Entrega:** Testar o processo de entrega de produtos, garantindo que o status de entrega dos pedidos é atualizado corretamente.
 - **Notificações:** Certificar que os usuários recebem atualizações sobre o status de entrega dos seus pedidos.
 - **Persistência de Dados:** Garantir que os registros de entrega são armazenados corretamente no banco de dados.

Banco de Dados Antes do Registro de Entrega

```
_id: ObjectId('669487c538401377a22cb93f')
▶ orderItems : Array (1)
▶ shippingAddress : Object
  paymentMethod : "Credito"
  itemsPrice : 30
  shippingPrice : 10
  taxPrice : 4.5
  totalPrice : 44.5
  user : ObjectId('668fcd6fbdf835140b387eaf')
  isPaid : true
  isDelivered : false
  isReturned : false
  createdAt : 2024-07-15T02:21:57.311+00:00
  updatedAt : 2024-07-15T02:30:31.103+00:00
  __v : 0
  paidAt : 2024-07-15T02:30:31.102+00:00
```

Tela de Registro de Entrega

Alugai

search products...

Cart admin Admin

Order 669487c538401377a22cb93f

Shipping

Name: admin

Address: rua,Rio de Janeiro, 00000,Brasil

Delivered at 14/07/2024, 23:38:14

Payment

Method: Credito

Paid at 14/07/2024, 23:30:31

Return

Name: admin

Not Returned

Items



Playstation 5

1

\$30

Order Summary

Items	\$30.00
Shipping	\$10.00
Tax	\$4.50
Order Total	\$44.50

Return Order

Banco de Dados Após o Registro de Entrega

```
_id: ObjectId('669487c538401377a22cb93f')
orderItems : Array (1)
shippingAddress : Object
  paymentMethod : "Credito"
  itemsPrice : 30
  shippingPrice : 10
  taxPrice : 4.5
  totalPrice : 44.5
user : ObjectId('668fcd6fbdf835140b387eaf')
isPaid : true
isDelivered : true
isReturned : false
createdAt : 2024-07-15T02:21:57.311+00:00
updatedAt : 2024-07-15T02:38:14.791+00:00
__v : 0
paidAt : 2024-07-15T02:30:31.102+00:00
deliveredAt : 2024-07-15T02:38:14.790+00:00
returnDate : 2024-07-15T02:38:14.790+00:00
```


- **Objetivos do teste Registro de Devolução**

- **Fluxo de Devolução:** Testar o processo de devolução de produtos, garantindo que os itens devolvidos são processados corretamente.
- **Persistência de Dados:** Garantir que os registros de devolução são armazenados corretamente no banco de dados.
- **Notificações:** Certificar que os usuários são notificados sobre a confirmação da devolução.

Banco de Dados Antes do Registro de Devolução

```
{
  _id: ObjectId('669487c538401377a22cb93f')
  orderItems : Array (1)
  shippingAddress : Object
    paymentMethod : "Credito"
    itemsPrice : 30
    shippingPrice : 10
    taxPrice : 4.5
    totalPrice : 44.5
  user : ObjectId('668fcd6fbdf835140b387eaf')
  isPaid : true
  isDelivered : true
  isReturned : false
  createdAt : 2024-07-15T02:21:57.311+00:00
  updatedAt : 2024-07-15T02:38:14.791+00:00
  __v : 0
  paidAt : 2024-07-15T02:30:31.102+00:00
  deliveredAt : 2024-07-15T02:38:14.790+00:00
  returnDate : 2024-07-15T02:38:14.790+00:00
}
```

Tela de Registro de Devolução

[Cart](#)
[admin](#)
[Admin](#)

Order 669487c538401377a22cb93f

Shipping

Name: admin
Address: rua Rio de Janeiro, 000008Brasil

Delivered at 14/07/2024, 23:38:14

Payment

Method: Credito

Paid at 14/07/2024, 23:30:31

Return

Name: admin
Returned at 15/07/2024, 00:09:10

Items

	PlayStation 5	1	\$30
--	---------------	---	------

Order Summary

Items	\$30.00
Shipping	\$10.00
Tax	\$4.50
Order Total	\$44.50

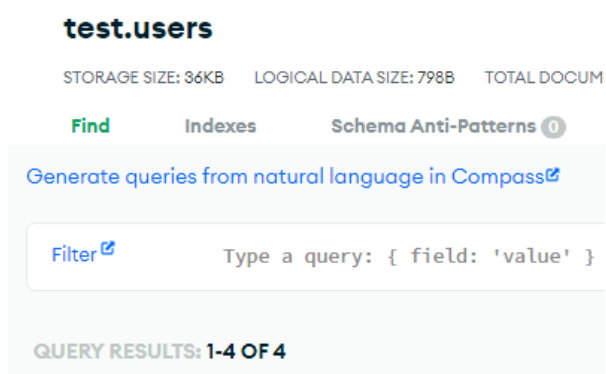
Banco de Dados Após Registro de Devolução

```
_id: ObjectId('669487c538401377a22cb93f')
▶ orderItems : Array (1)
▶ shippingAddress : Object
  paymentMethod : "Credito"
  itemsPrice : 30
  shippingPrice : 10
  taxPrice : 4.5
  totalPrice : 44.5
user : ObjectId('668fcd6fbdf835140b387eaf')
isPaid : true
isDelivered : true
isReturned : true
createdAt : 2024-07-15T02:21:57.311+00:00
updatedAt : 2024-07-15T03:09:10.569+00:00
__v : 0
paidAt : 2024-07-15T02:30:31.102+00:00
deliveredAt : 2024-07-15T02:38:14.790+00:00
returnDate : 2024-07-15T02:38:14.790+00:00
returnedAt : 2024-07-15T03:09:10.568+00:00
```

- **Objetivos do teste Cadastro de Usuário**

- **Validação de Dados:** Verificar se os dados fornecidos pelos usuários no momento do cadastro são validados corretamente (e.g., senha e confirmação de senha são iguais).
- **Segurança:** Garantir que as senhas são armazenadas de forma segura.
- **Persistência de Dados:** Assegurar que os dados do novo usuário são armazenados corretamente no banco de dados.
- **Autenticação:** Verificar se o novo usuário pode realizar login com as credenciais cadastradas e se tem acesso às funcionalidades apropriadas para o seu perfil.

Banco de Dados Antes do Cadastro de Usuário



Tela de Cadastro de Usuário

The screenshot shows the 'Sign Up' form in the Alugai application. The form is titled 'Sign Up' and has a dark header bar at the top with the Alugai logo and a search bar. The form fields are:

- Name:** Novo Usuario
- Email:** novousuario@email.com
- Password:** Masked with dots (.....)
- Confirm Password:** Masked with dots (.....)

 Below the fields is a yellow 'Sign Up' button. At the bottom, there is a link 'Sign-In' for users who already have an account.

Banco de Dados Após Cadastro do Usuário

test.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 1008B TOTAL DOCUMENTS: 5

[Find](#) [Indexes](#) [Schema Anti-Patterns](#) 0

[Generate queries from natural language in Compass](#)

[Filter](#) Type a query: { field: 'value' }

QUERY RESULTS: 1-5 OF 5

```
{
  "_id": ObjectId('6694959438401377a22cb957'),
  "name": "Novo Usuario",
  "email": "novousuario@email.com",
  "password": "$2a$10$d6u/uBfGAlR8czorz0PoH.tE6ALbJV7iRsoLnv13AH/LR0rCJB8ZW",
  "isAdmin": false,
  "createdAt": 2024-07-15T03:20:52.767+00:00,
  "updatedAt": 2024-07-15T03:20:52.767+00:00,
  "__v": 0
}
```

11. Considerações Finais

O desenvolvimento deste aplicativo de aluguel de itens evidenciou a importância de uma abordagem integrada e inovadora para atender às demandas contemporâneas por conveniência, acessibilidade e interação social.

A economia compartilhada, impulsionada pela transformação digital, mostrou-se um terreno fértil para soluções que promovem a sustentabilidade e colaboração.

O uso de tecnologias modernas e a implementação de um sistema robusto de reputação são pontos-chave que agregam valor à plataforma, proporcionando segurança e confiança aos usuários.

Este projeto não só consolidou os conhecimentos adquiridos ao longo do curso, mas também me ajudou a me desenvolver e a buscar sempre novos conhecimentos para enfrentar desafios reais no campo do desenvolvimento de software.

12. Trabalhos Futuros

Após a implementação das funcionalidades descritas nos casos de uso, o projeto pode ser expandido de várias maneiras. Algumas das novas funcionalidades que poderiam ser adicionadas incluem:

Utilização de Smart Lockers: Implementar o uso de smart lockers para facilitar a entrega e devolução de produtos alugados, oferecendo mais conveniência aos usuários.

Sistema de Penalização por Atraso: Desenvolver um sistema de penalizações para locatários que atrasarem a devolução dos produtos, incentivando a pontualidade e aumentando a eficiência do serviço.

Integração com Diferentes Métodos de Pagamento: Adicionar suporte para múltiplos métodos de pagamento, como carteiras digitais (e.g., PayPal, Google Pay, Apple Pay), Pix, transferência bancária, e pagamento por criptomoedas, ampliando as opções disponíveis para os usuários.

Avisos e Alertas por Email: Implementar um sistema de notificações por email para manter os usuários informados sobre o status de seus alugueis, vencimentos de prazo, e outras comunicações importantes.

Análise do locador/locatário: Implementar um sistema de análise do locador, mostrando sua experiência, quanto tempo está na plataforma, sua taxa de resposta, percentual de comentários que ele responde, tempo médio de resposta.

Chat de Comunicação entre Locatário e Locador: Desenvolver um sistema de chat para facilitar a comunicação direta entre locatários e locadores, permitindo resolver dúvidas e negociar detalhes de forma eficiente.

Além dessas novas funcionalidades, o projeto também pode ser aprimorado através da melhoria e continuação das funcionalidades existentes:

Implementação de um sistema mais robusto de cadastro com verificação de CPF e reconhecimento facial.

Variedade de opções de entrega e retirada de itens: Adicionar a possibilidade de retirada no endereço fornecido pelo locador, adicionar a integração com aplicativos existentes de entregas e a implementação de smart lockers em locais públicos para a retirada.

Variedade de Periodicidades para Aluguel: Permitir uma maior variedade de opções de periodicidade para aluguel, como diárias, semanais, mensais, e anuais, oferecendo mais flexibilidade aos usuários.

Comentários e Análise dos Produtos: Implementar um sistema de avaliações e comentários para que os usuários possam compartilhar suas experiências e opiniões sobre os produtos alugados, ajudando outros usuários a tomar decisões informadas.

Lista Definida de Categorias para Cadastro de Produtos: Criar uma lista predefinida de categorias para o cadastro de produtos, tornando o processo mais organizado e facilitando a busca por itens específicos na plataforma.

Menu para Editar Perfil do Usuário: Adicionar um menu para que os usuários possam editar suas informações de perfil, incluindo dados pessoais, preferências, e métodos de pagamento, melhorando a usabilidade e personalização do sistema.

Essas expansões e melhorias têm o potencial de aumentar significativamente a funcionalidade e a usabilidade da plataforma, proporcionando uma experiência mais rica e satisfatória para todos os usuários envolvidos.

13. Referências bibliográficas

- [1] How Airbnb designs for trust | Joe Gebbia. Disponível em
<<https://www.youtube.com/watch?v=16cM-RFid9U>> Acesso em 24 de out. 2023
- [2] Estudo aponta que 48% dos brasileiros buscam renda extra. Disponível em
<<https://valor.globo.com/patrocinado/dino/noticia/2022/07/22/estudo-aponta-que-48-dos-brasileiros-buscam-renda-extra.ghtml>> Acesso em 24 de out. 2023
- [3] The Current and Future State of the Sharing Economy. Disponível em
<<https://www.brookings.edu/articles/the-current-and-future-state-of-the-sharing-economy/#:~:text=The%20sharing%20economy%20is%20%E2%80%9Cthe,access%20to%20good%20and%20services%E2%80%9D>> Acesso em 24 de out. 2023
- [4] How Airbnb Turned Steve Jobs' Three-Click Rule Into Design Brilliance. Disponível em
<<https://skift.com/2017/02/14/how-airbnb-turned-steve-jobs-three-click-rule-into-design-brilliance/>> Acesso em 24 de out. 2023
- [5] Applying UML and patterns - Craig Larman - 2a edição. Disponível em:<<https://personal.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf>> Acesso em: 15 de julho de 2024
- [6] Introduction to Node.js. Disponível em:
<<https://nodejs.org/pt/learn/getting-started/introduction-to-nodejs>> Acesso em: 12 de dezembro de 2023
- [7] Roteamento com Express.js. Disponível em:
<<https://expressjs.com/pt-br/guide/routing.html>> Acesso em: 10 de janeiro de 2024
- [8] Node.js and Express. Disponível em:
<https://fullstackopen.com/en/part3/node_js_and_express> Acesso em: 10 de janeiro de 2024
- [9] Saving data to MongoDB. Disponível em:
<https://fullstackopen.com/en/part3/saving_data_to_mongo_db#fetching-object>

[ts-from-the-database](#) > Acesso em: 18 de março de 2024

[10] Mongoose.js Guide. Disponível em:
<<https://mongoosejs.com/docs/guide.html>> Acesso em: 18 de março de 2024

[11] Express Tutorial Part 3: Using a Database (with Mongoose). Disponível em:
<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/mongoose> Acesso em: 18 de março de 2024

[12] Introduction to React. Disponível em:
<https://fullstackopen.com/en/part1/introduction_to_react> Acesso em: 10 de janeiro de 2024

[13] Component state, event handlers. Disponível em:
<https://fullstackopen.com/en/part1/component_state_event_handlers#event-handling> Acesso em: 18 de março de 2024

[14] Redux: Um tutorial prático e simples! Disponível em:
<https://blog.geekhunter.com.br/redux-um-tutorial-pratico-e-simples/#O_que_e_o_Redux> Acesso em: 7 de abril de 2024

[15] Redux Essentials, Part 1: Redux Overview and Concepts. Disponível em:
<<https://redux.js.org/tutorials/essentials/part-1-overview-concepts>> Acesso em: 7 de abril de 2024

[16] Introdução ao Axios. Disponível em:
<<https://axios-http.com/ptbr/docs/intro>> Acesso em: 8 de abril de 2024

[17] Como usar o Axios com o React: o guia definitivo (2021). Disponível em:
<<https://www.freecodecamp.org/portuguese/news/como-usar-o-axios-com-o-react-o-guia-definitivo-2021/>> Acesso em: 8 de abril de 2024

[18] Why React-Bootstrap. Disponível em:
<<https://react-bootstrap.netlify.app/docs/getting-started/why-react-bootstrap>> Acesso em: 12 de abril de 2024

[19] O que é JSON Web Tokens? Disponível em:
<https://www.alura.com.br/artigos/o-que-e-json-web-tokens?utm_term=&utm_

[campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=164068847699&hsa_ad=703853174125&hsa_src=g&hsa_tgt=dsa-2273097816642&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&qad_source=1&gclid=EAlaIQobChMI m6HEvZushwMVQ19IAB2m9gouEAAYASAAEgKoVPD_BwE](https://www.google.com/search?q=campaign=%5BSearch%5D+%5BPerformance%5D+-+Dynamic+Search+Ads+-+Artigos+e+Conte%C3%BAdos&utm_source=adwords&utm_medium=ppc&hsa_acc=7964138385&hsa_cam=11384329873&hsa_grp=164068847699&hsa_ad=703853174125&hsa_src=g&hsa_tgt=dsa-2273097816642&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&qad_source=1&gclid=EAlaIQobChMI m6HEvZushwMVQ19IAB2m9gouEAAYASAAEgKoVPD_BwE)> Acesso em: 16 de abril de 2024

[20] Introduction to JSON Web Tokens. Disponível em:
<<https://jwt.io/introduction>> Acesso em: 16 de abril de 2024

[21] Password Hashing in Node.js Using bcryptjs Library. Disponível em:
<https://dev.to/mr_walkr/password-hashing-in-nodejs-using-bcryptjs-library-3j56> Acesso em: 1 de maio de 2024

[22] React Google Charts. Disponível em:
<<https://www.react-google-charts.com/examples/pie-chart>> Acesso em: 16 de junho de 2024

[23] React Google Charts. Disponível em:
<<https://www.react-google-charts.com/examples/line-chart>> Acesso em: 16 de junho de 2024

[24] Introduction to Cloudinary for Node.js. Disponível em:
<https://cloudinary.com/documentation/node_integration> Acesso em: 23 de junho de 2024

[25] 10 Best Web Development Stacks. Disponível em:
<<https://dev.to/snevy1/10-best-web-development-stacks-409i>> Acesso em: 12 de dezembro de 2023

[26] Stack Overflow 2023 Developer Survey. Disponível em:
<<https://survey.stackoverflow.co/2023/#developer-profile-developer-roles>> Acesso em: 12 de dezembro de 2023

[27] Top 10 Technology Stack in Software Development For 2024. Disponível em:
<<https://medium.com/@shubhijain/top-10-technology-stack-in-software-development-for-2024-6e0478e12f56>> Acesso em: 12 de dezembro de 2023