



LUIZ FELIPE MONTUANI E SILVA

**RELATÓRIO DE ESTÁGIO:
DESENVOLVIMENTO MOBILE NA IOASYS
DESENVOLVIMENTO DE SOFTWARE LTDA**

**LAVRAS – MG
2023**

LUIZ FELIPE MONTUANI E SILVA

RELATÓRIO DE ESTÁGIO:
DESENVOLVIMENTO MOBILE NA IOASYS DESENVOLVIMENTO DE SOFTWARE
LTDA

Relatório de Estágio Supervisionado
apresentado à Universidade Federal de Lavras
como parte das exigências do curso de Ciência
da Computação, para obtenção do título de
Bacharel.

Prof^ª. Dra. Renata Teles Moreira
Orientadora

LAVRAS - MG
2023

LUIZ FELIPE MONTUANI E SILVA

RELATÓRIO DE ESTÁGIO:
DESENVOLVIMENTO MOBILE NA IOASYS DESENVOLVIMENTO DE SOFTWARE
LTDA

Relatório de Estágio Supervisionado
apresentado à Universidade Federal de Lavras
como parte das exigências do curso de Ciência
da Computação, para obtenção do título de
Bacharel.

Aprovado em 27 de julho de 2023

Profa. Dra. Renata Teles Moreira	DCC/UFLA
Prof. Dr. Bruno de Abreu Silva	DCC/UFLA
Prof. Dr. Paulo Afonso Parreira Junior	DCC/UFLA


Prof^ª. Dra. Renata Teles Moreira
Orientadora

LAVRAS - MG
2023

AGRADECIMENTOS

Agradeço primeiramente a Deus, pela ajuda em todas as situações. A minha família que muito me apoiou em todos os momentos da minha vida. A Universidade Federal de Lavras, em especial ao Departamento de Ciência da computação, por todo conhecimento que obtive durante o curso. Aos professores que ministraram excelentes aulas, não somente sobre a ementa da disciplina, mas também sobre a vida.

A empresa Ioasys pela oportunidade no mercado de trabalho.

A equipe de corrida UFLA Runners pela união e companheirismo.

A mim, por todo o esforço e dedicação durante esse período tão rico e importante para minha formação pessoal e profissional.

Muito obrigado!

RESUMO

Com o presente trabalho pretendeu-se evidenciar a importância da realização do estágio de forma ativa, participativa e prática, cujo objetivo principal é aplicar os conhecimentos adquiridos em sala de aula, nos contextos de trabalho e inovação de tecnologias que proporcionam novas maneiras de pensar e construir produtos que agradem e conquistem os potenciais consumidores. O documento discorre sobre as principais atividades realizadas pelo autor durante o período de estágio em desenvolvimento de software, realizado na empresa Ioasys Desenvolvimento de Software LTDA., a qual é uma empresa brasileira de inovação tecnológica especializada na criação e desenvolvimento de aplicativos e demais componentes do universo digital, a partir de sua missão e propósitos discorrendo ainda detalhadamente sobre as atividades desenvolvidas, neste período e o bom êxito dos resultados alcançados. Por fim, a partir dos dados e resultados apresentados, demonstra-se a importância de empregar os conteúdos e aprendizados contextualizando-os com as novas formas de criar e reinventar os produtos, através da análise, observação e coletas de dados referentes aos gostos e preferências das pessoas em vista de produzir a melhor propaganda para satisfazer seus desejos e necessidades.

Palavras-chave: Estágio. Flutter. Dart. Metodologias Ágeis.

LISTA DE FIGURAS

Figura 1 - Bloco de código na linguagem <i>Dart</i>	12
Figura 2 - Exemplo de teste de unidade utilizando a linguagem <i>Dart</i>	19
Figura 3 - Página de listagem produtos	22
Figura 4 - Página de cadastro de um produto (Parte 1)	23
Figura 5 - Página de cadastro de um produto (Parte 2)	23
Figura 6 - Página de edição de um produto (Parte 1)	24
Figura 7 - Página de edição de um produto (Parte 2)	25
Figura 8 - Menu para exclusão de um produto.....	26
Figura 9 - Seleção de produtos	27
Figura 10 - Reserva Fácil - Tela de login	29
Figura 11 - Minha CVC - Tela inicial	30
Figura 12 - Minha CVC - Perfil.....	31
Figura 13 - Minha CVC - Central de Ajuda	32
Figura 14 - Minha CVC - Feedback	33
Figura 15 - Minha CVC - Seção “Mais Ajuda”.....	34
Figura 16 - Minha CVC - Abertura de Chamado	35
Figura 17 – Vertical de Carros - Seleção de local de retirada e devolução.....	36
Figura 18 - Vertical de Carros - Seleção de data e horário	37
Figura 19 - Vertical de Carros – Listagem de resultados	38
Figura 20 - Vertical de Carros – Edição de busca	39
Figura 21 - Vertical de Carros – Filtros.....	40
Figura 22 - Vertical de Carros – Detalhes do produto.....	41
Figura 23 - Vertical de Carros – Checkout.....	42
Figura 24 - Vertical de Carros – Resultados (Parte 1).....	44
Figura 25 - Vertical de Carros – Resultados (Parte 2).....	44

Sumário

1	INTRODUÇÃO	8
2	SOBRE A ORGANIZAÇÃO	9
2.1	Processo de Desenvolvimento da loasys	9
2.2	Produtos	10
3	REVISÃO DA LITERATURA	11
3.1	Front-end	11
3.2	Dart	11
3.3	Flutter	12
3.4	Clean Architecture	13
3.5	SOLID	14
3.6	Metodologias ágeis	15
3.6.1	Scrum	16
3.6.2	Kanban	17
3.7	Testes de unidade	18
4	ATIVIDADES DESENVOLVIDAS	20
4.1	Atividades gerais	20
4.1.1	Versionamento de código	20
4.1.2	Revisão de código	21
4.1.3	Mentoria	21
4.2	Backoffice	21
4.2.1	Cadastro de produto	22
4.2.2	Edição de produto	24
4.2.3	Exclusão de produto	25
4.3	Reserva Fácil	27

4.3.1	<i>Correção de bugs</i>	27
4.3.2	<i>Mapeamento de erros</i>	28
4.3.3	<i>Testes de Widgets</i>	28
4.4	<i>Minha CVC e Submarino Viagens</i>	29
4.4.1	<i>Central de ajuda</i>	31
4.4.1.1	<i>Abertura de chamado</i>	34
4.4.2	<i>Vertical de carros</i>	35
4.4.2.1	<i>Edição da busca</i>	38
4.4.2.2	<i>Tela de filtros</i>	39
4.4.2.3	<i>Tela de detalhes</i>	40
5	<i>CONSIDERAÇÕES FINAIS</i>	43
	<i>REFERÊNCIAS</i>	46

1 INTRODUÇÃO

O presente documento descreve as atividades realizadas pelo autor durante o período de estágio em desenvolvimento de software, realizado na empresa Ioasys Desenvolvimento de Software LTDA., a qual é uma empresa brasileira de inovação tecnológica especializada na criação e desenvolvimento de aplicativos e demais componentes do universo digital.

São apresentados o projeto trabalhado, as tecnologias e os processos de desenvolvimento utilizados e a cultura empresarial, bem como está destacada a importância do estágio, que proporcionou muita interação pessoal e aprendizado técnico, no desenvolvimento pessoal e profissional do autor.

Como objetivos principais do estágio realizado o autor almejou a aquisição de conhecimentos técnicos específicos adquiridos no curso para aplicação deles no desenvolvimento das atividades do estágio, com o intuito de alcançar a excelência e a consolidação dos conteúdos teóricos e práticos aprendidos em sala de aula.

O curso proporcionou ao autor os conhecimentos, as aprendizagens essenciais e as ferramentas necessárias para a realização do estágio, o que foi de suma importância para o desenvolvimento acadêmico, profissional e pessoal do autor, uma vez que mostrou de forma crítica reflexiva uma visão mais ampla do mercado de trabalho e das formas do autor se relacionar com as pessoas e com o próprio mercado.

O projeto o qual o autor estava inserido é a construção de aplicações para a empresa CVC Brasil Operadora e Agência de Viagens S.A., o qual é apenas alguns dos diversos projetos que a Ioasys possui com a CVC.

A alocação principal do autor foi no desenvolvimento *front-end* utilizando um kit de desenvolvimento de interface de usuário (UI toolkit e framework) chamado *Flutter*, criado pela empresa Google em 2015, o qual possibilita a criação de aplicativos compilados nativamente para os sistemas web, *Android*, *iOS* e *desktop*.

Além deste capítulo introdutório, este relatório de estágio está organizado como segue. O Capítulo 2 apresenta a história e estrutura da empresa em que o autor realizou o estágio. O Capítulo 3 descreve as ferramentas e conceitos utilizados. O Capítulo 4 apresenta as atividades realizadas. Por fim, o Capítulo 5 apresenta as considerações finais acerca das atividades desempenhadas durante o estágio.

2 SOBRE A ORGANIZAÇÃO

A Ioasys é uma empresa de tecnologia que se dedica ao desenvolvimento de soluções para a gestão empresarial. Com uma equipe altamente capacitada e tecnologias avançadas, a Ioasys oferece uma ampla gama de produtos e serviços para atender às necessidades de empresas de todos os tamanhos e segmentos.

Os produtos da Ioasys incluem softwares de gestão de recursos humanos, financeiro, vendas e suprimentos, bem como soluções de gestão de projetos e colaboração. Com uma interface intuitiva e fácil de usar, esses produtos permitem aos usuários gerenciar suas operações de forma eficiente e efetiva.

Além de seus produtos, a Ioasys também oferece serviços de consultoria e suporte técnico para ajudar as empresas a implementar e aproveitar ao máximo as soluções oferecidas. Com sua ampla experiência e conhecimento de tecnologia, a Ioasys ajuda as empresas a alcançar seus objetivos de negócios e a maximizar seu desempenho.

A sede da Ioasys está localizada em Belo Horizonte e suas filiais em São Paulo e Lavras. O autor estagiou de forma remota. A equipe que o autor atuou contava com 20 membros, sendo 15 desenvolvedores de software, 3 designers e 2 gerentes de produto.

2.1 Processo de Desenvolvimento da Ioasys

A Ioasys é uma empresa de desenvolvimento de software que segue um processo de desenvolvimento ágil para entregar soluções de alta qualidade aos seus clientes. O processo de desenvolvimento da Ioasys é baseado nos princípios do Manifesto Ágil e utiliza metodologias como Scrum e Kanban.

1. Planejamento: Nesta fase, a equipe da Ioasys trabalha em conjunto com o cliente para entender os requisitos do projeto e definir as metas a serem alcançadas. São identificadas as funcionalidades prioritárias e estabelecido um cronograma inicial.

2. Análise: Nessa etapa, os analistas da Ioasys realizam uma análise mais detalhada dos requisitos levantados, identificando as necessidades do usuário e definindo as histórias de usuário. As histórias de usuário são pequenas descrições das funcionalidades desejadas, escritas na perspectiva do usuário.

3. Planejamento da Sprint: Com base nas histórias de usuário definidas, a equipe da Ioasys realiza o planejamento da sprint. Uma sprint é um período fixo (geralmente de 2 a 4 semanas) durante o qual a equipe desenvolve e entrega incrementos do software. Nessa fase,

são definidas as tarefas específicas a serem realizadas durante a sprint e estimadas as horas necessárias para cada uma delas.

4. Desenvolvimento: Durante a sprint, os desenvolvedores da Ioasys trabalham na implementação das funcionalidades definidas nas histórias de usuário. Eles utilizam práticas de desenvolvimento ágil, como programação em pares, revisões de código e testes automatizados, para garantir a qualidade do software.

5. Revisão: Ao final de cada sprint, a equipe realiza uma revisão do trabalho realizado, demonstrando as funcionalidades implementadas aos stakeholders e coletando feedback. Esse feedback é utilizado para melhorar o produto e ajustar as prioridades para as próximas sprints.

6. Retrospectiva: Também ao final de cada sprint, a equipe realiza uma retrospectiva para avaliar o próprio processo de trabalho. São identificados pontos fortes e áreas de melhoria, e são definidas ações para aprimorar a eficiência e a qualidade do desenvolvimento.

7. Entrega: Após várias sprints, o produto final é entregue ao cliente. Dependendo do projeto, pode haver uma fase de testes e validação antes da entrega final.

É importante ressaltar que o processo de desenvolvimento da Ioasys é flexível e adaptável às necessidades específicas de cada projeto. A empresa busca constantemente aprimorar suas práticas e incorporar novas tecnologias e metodologias para oferecer soluções inovadoras e de alto valor aos seus clientes.

2.2 Produtos

Durante o estágio o autor atuou exclusivamente em produtos da cliente CVC¹, esta é uma empresa do ramo do turismo e contratou a Ioasys para construção de diversos projetos. Dentre eles estão os projetos Backoffice Mobile, Reserva Fácil, Minha CVC e Submarino Viagens, nos quais o autor trabalhou durante o período do estágio e foram detalhados no capítulo 4 desse documento.

¹ <https://www.cvc.com.br/institucional/sobre-a-cvc-nossa-historia>

3 REVISÃO DA LITERATURA

Este capítulo trata da fundamentação teórica dos aspectos abordados durante as atividades do estágio, onde são apresentados os conceitos de *front-end*, Dart, Flutter, Clean Architecture, SOLID, Metodologias ágeis e testes de unidade no contexto da Ciência da Computação.

3.1 *Front-end*

Front-end é a parte da aplicação que o usuário final visualiza e interage. Ele é responsável por apresentar a interface gráfica do usuário (GUI) e controlar a interação do usuário com a aplicação. O *front-end* é construído com tecnologias como HTML, CSS e *JavaScript* e pode ser implementado com frameworks como *React*, *Angular* ou *Vue* (SOUTO, 2023).

O *front-end* é crucial para a experiência do usuário final, pois é responsável por fornecer uma interface fácil de usar e intuitiva. Ele também é responsável por garantir a funcionalidade e a responsividade da aplicação, permitindo que ela se adapte a diferentes tamanhos de tela e dispositivos.

Além disso, o *front-end* também é responsável por realizar validações de entrada de dados e garantir a segurança da aplicação, juntamente com as proteções no *back-end*, evitando ataques como injeção de código ou roubo de dados.

Logo, o *front-end* é uma parte fundamental da aplicação que afeta diretamente a experiência do usuário final e é crucial para o sucesso de qualquer projeto de software.

3.2 Dart

*Dart*² é uma linguagem de programação de alto nível desenvolvida pela Google. É uma linguagem estaticamente tipada, o que significa que os tipos de dados são determinados em tempo de compilação, o que garante a segurança do código. *Dart* é uma linguagem moderna e fácil de aprender, com sintaxe semelhante a outras linguagens populares como Java e *JavaScript*.

A linguagem *Dart* é amplamente utilizada para o desenvolvimento de aplicativos *web*, móveis e *desktop*. É uma opção popular para desenvolvedores que buscam uma alternativa para as linguagens mais antigas e desejam aproveitar as vantagens de uma linguagem mais moderna

² <https://dart.dev/>

e flexível. Além disso, a linguagem *Dart* é compatível com o ambiente de desenvolvimento de aplicativos da Google, o *Flutter*, que permite a criação de aplicativos móveis para *iOS* e *Android* a partir de uma única base de código.

A linguagem *Dart* possui uma ampla biblioteca padrão, que inclui recursos avançados como programação assíncrona, manipulação de *strings*, coleções de dados e muito mais. Além disso, existem muitas bibliotecas e pacotes disponíveis para a linguagem *Dart*, que fornecem recursos adicionais e tornam o desenvolvimento de aplicativos ainda mais fácil.

Logo, a linguagem de programação *Dart* é moderna, poderosa e oferece muitos recursos avançados, além de ser compatível com vários ambientes de desenvolvimento. Na Figura 1 é possível verificar um laço de repetição escrito na linguagem *Dart*, onde será mostrado no terminal a *string* da linha 3, a cada 500 milissegundos até o valor da variável “*i*” ser igual a 4.

Figura 1 - Bloco de código na linguagem *Dart*

A screenshot of a code editor window with a dark background and light-colored text. The code is written in Dart and consists of seven lines. Line 1: `1 Future<void> main() async {`. Line 2: `2 for (var i = 0; i < 5; i++) {`. Line 3: `3 print('Hello from Dart ($i)!');`. Line 4: `4 await Future.delayed(Duration(milliseconds: 500));`. Line 5: `5 }`. Line 6: `6 }`. Line 7: `7`. The code is color-coded: `Future` is blue, `void` is blue, `main` is blue, `async` is blue, `for` is blue, `var` is blue, `i` is blue, `=` is blue, `0` is blue, `<` is blue, `5` is blue, `>` is blue, `;` is blue, `++` is blue, `{` is blue, `print` is blue, `'Hello from Dart ($i)!'` is red, `;` is blue, `await` is blue, `Future` is blue, `.` is blue, `delayed` is blue, `(Duration(milliseconds: 500))` is blue, `;` is blue, `}` is blue, `}` is blue.

Fonte: Do Autor (2022)

Durante o estágio foi utilizada a linguagem *Dart* para o desenvolvimento dos produtos em que o autor trabalhou, citados na seção 2.2 desse documento.

3.3 Flutter

*Flutter*³ é um framework de desenvolvimento de aplicativos móveis, desenvolvido pelo Google, que permite a criação de aplicativos para *Android*, *iOS* e *web* de forma rápida e eficiente. Ele utiliza a linguagem de programação *Dart* e tem como principal característica a possibilidade de criar interfaces de usuário customizadas e animadas com facilidade.

O *Flutter* possui uma ampla biblioteca de *widgets*, que são componentes visuais como botões, textos e caixas de diálogo, entre outros, que podem ser usados para criar a interface do

³ <https://flutter.dev/>

aplicativo. Além disso, o framework permite a criação de aplicativos de alta qualidade e com boa performance, mesmo em dispositivos com hardware limitado.

Outra vantagem do *Flutter* é a possibilidade de testar e depurar a aplicação rapidamente, sem a necessidade de fazer *upload* para uma loja de aplicativos. Isso permite que os desenvolvedores possam experimentar novas ideias e corrigir erros de forma mais rápida e eficiente.

O *Flutter* também possui uma comunidade ativa e crescente, o que significa que há muitos recursos disponíveis, incluindo documentação, fóruns de discussão e pacotes de terceiros que podem ser usados para acelerar o desenvolvimento.

Em resumo, o *Flutter* é uma plataforma de desenvolvimento de aplicativos móveis moderna e poderosa que permite a criação de aplicativos atraentes e de alta qualidade para diversas plataformas. Com sua biblioteca de *widgets*, possibilidade de testes rápidos e uma comunidade ativa, o *Flutter* é uma excelente escolha para desenvolvedores que procuram criar aplicativos móveis de alta qualidade.

Durante o estágio o *Flutter* foi utilizado juntamente com a linguagem *Dart*, citada na seção anterior, para o desenvolvimento dos produtos em que o autor trabalhou.

3.4 Clean Architecture

A *Clean Architecture* é uma abordagem de arquitetura de software que se concentra na separação das responsabilidades do código, com o objetivo de torná-lo mais fácil de manter, modificar e escalar. Ela foi proposta por Robert Martin e se baseia em alguns princípios simples (MARTIN, 2019):

- Separar preocupações: A *Clean Architecture* separa o código em camadas, cada uma delas com responsabilidades bem definidas, para que as alterações em uma camada não afetem as outras.
- Usar dependências invertidas: A *Clean Architecture* insere dependências de forma que as camadas internas do código dependam apenas das camadas externas, e não o contrário. Isso garante que a lógica interna do software não seja afetada por mudanças na camada externa.
- Garantir a testabilidade: A *Clean Architecture* permite que as camadas internas do código sejam facilmente testadas, independentemente das camadas externas, o que ajuda a garantir a qualidade do software.

- Foco no negócio: A *Clean Architecture* coloca o domínio do negócio (a lógica de negócios da aplicação) no centro do design da arquitetura, o que garante que o software esteja centrado nas necessidades do negócio.

Portanto, a *Clean Architecture* é uma abordagem de arquitetura de software que visa tornar o código mais fácil de manter, modificar e escalar, garantindo a qualidade, testabilidade e foco no negócio. Ao seguir esses princípios, é possível construir software mais robusto, escalável e sustentável.

Durante o estágio a abordagem *Clean Architecture* foi utilizada como modelo de arquitetura de todos os softwares em que o estagiário atuou.

3.5 SOLID

SOLID é um acrônimo que representa cinco princípios de design de software importantes, desenvolvidos por Robert Martin (MARTIN, 2022):

- Princípio da responsabilidade única (*Single Responsibility Principle*): Este princípio afirma que cada classe ou módulo deve ter apenas uma responsabilidade, e todas as suas operações e dados devem estar relacionados a essa responsabilidade. Isso ajuda a manter o código limpo, organizado e fácil de modificar.
- Princípio da abertura/fechamento (*Open/Closed Principle*): Este princípio afirma que as classes ou módulos devem ser projetados para serem estendidos, mas não modificados. Isso significa que você pode adicionar novas funcionalidades sem precisar mudar o código existente.
- Princípio da substituição de Liskov (*Liskov Substitution Principle*): Este princípio afirma que objetos de uma classe base devem ser substituíveis por objetos de suas subclasses sem quebrar a aplicação. Isso ajuda a garantir a consistência do código.
- Princípio da segregação de interface (*Interface Segregation Principle*): Este princípio afirma que as interfaces devem ser pequenas e específicas, para que classes não tenham que implementar métodos que não precisam. Isso ajuda a manter o código mais limpo e fácil de manter.
- Princípio de inversão de dependência (*Dependency Inversion Principle*): Este princípio afirma que as classes devem depender de abstrações, e não de detalhes

de implementação. Isso ajuda a tornar o código mais flexível e fácil de modificar.

Em suma, os princípios SOLID ajudam a garantir que o código seja fácil de manter, modificar e escalar, tornando-o mais sustentável e escalável. Ao seguir esses princípios, o desenvolvedor pode construir software mais robusto e confiável.

Durante o estágio a abordagem SOLID foi utilizada para garantir a qualidade, fácil manutenibilidade e desacoplamento dos softwares em que o autor trabalhou.

3.6 Metodologias ágeis

Metodologias Ágeis são abordagens para gerenciamento de projetos que se concentram na flexibilidade, colaboração e entrega contínua de valor. Elas foram desenvolvidas como uma alternativa aos métodos de gerenciamento de projetos tradicionais, que muitas vezes são burocráticos e menos eficientes.

As metodologias ágeis valorizam a comunicação e colaboração entre os membros da equipe, bem como a flexibilidade para responder a mudanças no projeto. Elas também priorizam a entrega de valor ao cliente ao invés de seguir um plano rigoroso (BECK et al., 2001).

Algumas das metodologias ágeis mais populares incluem:

- Scrum: uma abordagem iterativa e incremental para o gerenciamento de projetos que se concentra em colaboração e entrega contínua de valor (SCHWABER; SUTHERLAND, 2020).
- Kanban: uma metodologia que se concentra na visualização do fluxo de trabalho e na limitação do número de tarefas em andamento ao mesmo tempo (VACANTI; COLEMAN, 2020).
- *Extreme Programming (XP)*: uma metodologia ágil que se concentra em práticas de programação efetivas, como programação em pares e testes automatizados (WELLS, 2022).
- *Lean*: uma abordagem ágil baseada na filosofia *Lean Manufacturing*, que se concentra em eliminar desperdício e maximizar o valor entregue ao cliente (BALARDIM, 2019).

Metodologias ágeis são amplamente utilizadas no desenvolvimento de software, pois permitem que as equipes trabalhem de maneira mais eficiente, respondendo rapidamente a mudanças e entregando valor ao cliente de maneira contínua. Ao contrário das metodologias

tradicionais, as metodologias ágeis são flexíveis e permitem que as equipes se adaptem a mudanças, garantindo um resultado de sucesso. Neste documento iremos nos aprofundar nas metodologias Scrum e Kanban, as quais foram utilizadas durante o período de estágio.

3.6.1 Scrum

Scrum é uma metodologia ágil de gerenciamento de projetos de software que se concentra na entrega de valor e na colaboração constante entre equipes. Ele é amplamente utilizado para gerenciar projetos complexos e altamente incertos, como projetos de software. Conforme descrito no guia do Scrum, "Scrum é uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptativas para problemas complexos". (SCHWABER; SUTHERLAND, 2020, p. 3, tradução nossa)

A metodologia Scrum se baseia em ciclos curtos de trabalho, conhecidos como *sprints*, que duram geralmente de uma a quatro semanas. Cada *sprint* começa com uma reunião de planejamento, na qual a equipe determina o que será entregue no final do *sprint*. Em seguida, a equipe trabalha em colaboração para completar as tarefas e alcançar seus objetivos, realizando uma reunião diária para alinhamento das atividades em desenvolvimento. Ao final de cada *sprint*, há uma revisão com o cliente para avaliar o progresso e planejar a próxima *sprint* e uma reunião de retrospectiva onde o Time Scrum analisa os principais pontos positivos e negativos da *sprint*, com o intuito de não cometerem os mesmos erros e aprimorarem o fluxo de trabalho. Cada etapa do fluxo citado acima, *sprint*, reunião diária, revisão e retrospectiva, é um evento do *framework* Scrum.

Outros aspectos importantes de Scrum incluem:

- Time Scrum: uma equipe multifuncional composta pelo Product Owner, o Time de Desenvolvimento e o Scrum Master.
 - Product Owner: “O Product Owner é o responsável por maximizar o valor do produto e do trabalho do Time de Desenvolvimento” (SCHWABER; SUTHERLAND, 2020, p. 5, tradução nossa);
 - Time de Desenvolvimento: “O Time de Desenvolvimento consiste de profissionais que realizam o trabalho de entregar uma versão usável que

potencialmente incrementa o produto “Pronto” ao final de cada Sprint.” (SCHWABER; SUTHERLAND, 2020, p. 6, tradução nossa);

- Scrum Master: “O Scrum Master é responsável por garantir que o Scrum seja entendido e aplicado.” (SCHWABER; SUTHERLAND, 2020, p. 7, tradução nossa);
- Artefatos do Scrum: “Os artefatos do Scrum representam o trabalho ou o valor para o fornecimento de transparência e oportunidades para inspeção e adaptação.” (SCHWABER; SUTHERLAND, 2020, p. 13, tradução nossa);
 - Backlog do Produto: é uma lista ordenada de tudo que é necessário para a conclusão do produto;
 - Backlog da *sprint*: conjunto de itens do Backlog do Produto que serão desenvolvidos durante a *sprint*;
 - Incremento: entregável finalizado ao final de cada *sprint*, ou seja, uma parcela funcional do produto final.

O Scrum permite que as equipes trabalhem de maneira colaborativa e ágil, adaptando-se a mudanças e entregando valor constantemente. Isso é importante em projetos de software, onde as mudanças são comuns e o tempo é crítico. Ao seguir a metodologia Scrum, equipes podem garantir que o projeto será concluído de maneira eficiente e eficaz, entregando valor ao cliente.

3.6.2 Kanban

Kanban é uma metodologia de gerenciamento de projetos que se concentra em visualizar o fluxo de trabalho e limitar o número de tarefas em andamento. Ela foi originalmente desenvolvida na indústria automotiva japonesa e tem sido aplicada com sucesso a vários outros setores, incluindo software. (VACANTI; COLEMAN, 2020, p.4, tradução nossa).

A metodologia Kanban se baseia em um quadro de tarefas com colunas que representam o estado atual de cada tarefa. Por exemplo, as colunas comuns incluem "Para Fazer", "Em Andamento" e "Concluído". Cada tarefa é representada por um cartão que é movido de uma coluna para outra conforme avança.

Outros aspectos importantes de Kanban incluem:

- Limite de trabalho em andamento: limitar o número de tarefas que podem estar em andamento ao mesmo tempo, garantindo que a equipe não se sobrecarregue.
- Priorização: determinar a ordem das tarefas baseadas nas necessidades do projeto e nas dependências entre as tarefas.
- Feedback contínuo: reunir regularmente para avaliar o progresso e identificar oportunidades de melhoria.
- Flexibilidade: permitir mudanças no processo conforme as necessidades mudam.

Kanban é uma metodologia ágil e flexível que permite que as equipes trabalhem de maneira mais eficiente, visualizando o fluxo de trabalho e adaptando-se a mudanças conforme elas surgem. Isso é importante em projetos de software, onde mudanças são comuns e o tempo é crítico. Ao seguir a metodologia Kanban, equipes podem garantir que o projeto será concluído de maneira eficiente e eficaz, entregando valor ao cliente.

3.7 Testes de unidade

Testes de unidade são testes automatizados que verificam se uma pequena parte do código de um software funciona como esperado. Eles são projetados para testar uma unidade de código específica, como uma função, classe ou método, e garantir que ela execute sua tarefa corretamente, independentemente do contexto em que é chamada.

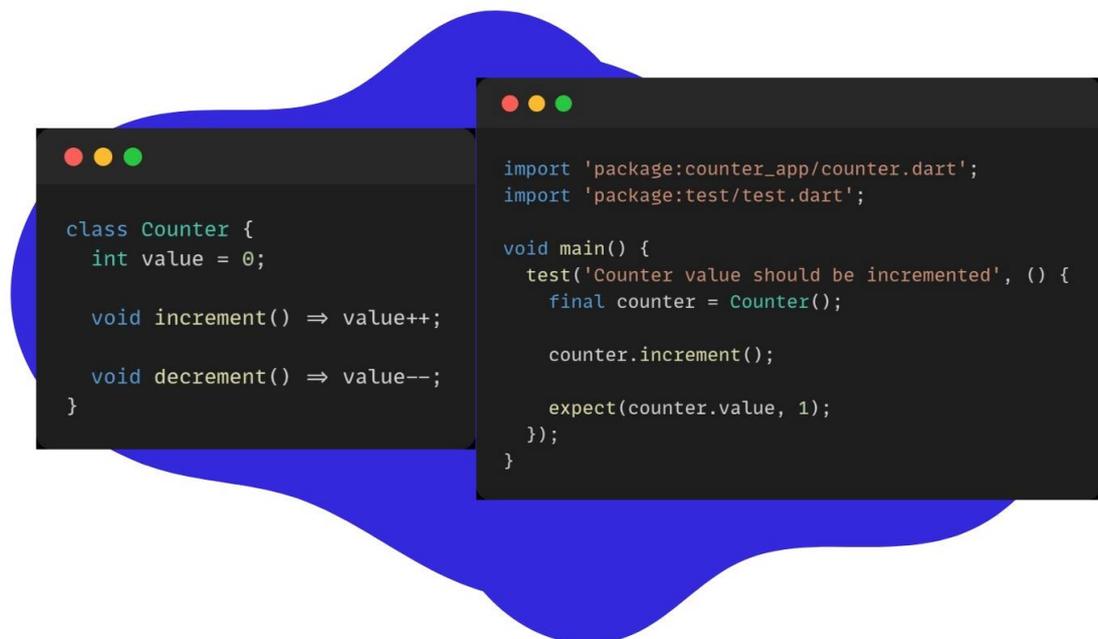
Os testes de unidade são importantes porque:

- Melhoram a qualidade do código: Os testes de unidade garantem que as unidades de código funcionem corretamente, o que ajuda a prevenir erros e *bugs* no software.
- Facilitam o desenvolvimento de software: Ao testar pequenas partes do código de forma isolada, os desenvolvedores podem identificar e corrigir problemas de forma mais eficiente, o que ajuda a acelerar o processo de desenvolvimento.
- Aumentam a confiança no código: Ao realizar testes de unidade, os desenvolvedores podem ter certeza de que as unidades de código funcionam como esperado, o que aumenta a confiança no software.

- Facilitam a manutenção do código: Quando o código é testado e verificado com testes de unidade, fica mais fácil para os desenvolvedores fazerem alterações e adicionar novos recursos, pois eles podem ter certeza de que o código existente ainda funciona corretamente.

Logo, os testes de unidade são uma parte importante do processo de desenvolvimento de software, pois ajudam a garantir a qualidade, confiança e manutenibilidade do código. Eles são normalmente realizados como parte da integração contínua, que é um processo de entrega de software que combina automação de testes, construção e implantação de software.

Figura 2 - Exemplo de teste de unidade utilizando a linguagem *Dart*

The image shows two overlapping code editor windows with a dark background and light-colored text. The left window displays the implementation of a Counter class, and the right window displays a unit test for the Counter class. Both windows have a blue shadow behind them.

```
class Counter {  
  int value = 0;  
  
  void increment() => value++;  
  
  void decrement() => value--;  
}
```

```
import 'package:counter_app/counter.dart';  
import 'package:test/test.dart';  
  
void main() {  
  test('Counter value should be incremented', () {  
    final counter = Counter();  
  
    counter.increment();  
  
    expect(counter.value, 1);  
  });  
}
```

Fonte: Do Autor (2022)

Testes de unidade foram utilizados durante o estágio para garantir a qualidade do produto final.

4 ATIVIDADES DESENVOLVIDAS

Este capítulo descreve algumas das principais atividades realizadas durante o período de estágio. Na seção 4.1 serão descritas as atividades comumente realizadas em todos os projetos trabalhados durante o estágio, já na seção 4.2 serão expostas as atividades desenvolvidas no sistema Backoffice, o qual é utilizado para inserção, edição e remoção de produtos do *marketplace* contido no aplicativo da CVC. Na sequência, na seção 4.3, serão tratadas as atividades realizadas no aplicativo chamado Reserva Fácil, o qual é uma aplicação destinada ao público B2B, ou seja, os agentes de viagem das franquias da CVC. Por fim, na seção 4.4, será apresentado o que foi implementado no aplicativo principal da empresa CVC.

4.1 Atividades gerais

Esta seção descreve as atividades que foram realizadas em todos os projetos trabalhados durante o estágio. Estas atividades foram essenciais para um bom fluxo de trabalho, bem como para que o código fosse revisado e tivesse uma ótima qualidade.

4.1.1 Versionamento de código

Os códigos dos projetos trabalhados ficavam armazenados em repositórios utilizando o sistema de versionamento chamado Git⁴, os quais são enviados para repositórios remotos para fácil acesso de todos os membros da equipe. Além do Git, também eram utilizadas as aplicações Bitbucket⁵ e GitLab⁶.

O controle de versão seguia um fluxo de desenvolvimento bem estruturado para que se pudesse gerar versões das aplicações a qualquer momento desejado. O projeto mantinha uma *branch* principal chamada de “*master*”, a qual continha a última versão estável da aplicação. Paralelamente havia uma *branch* chamada de “*dev*”, a qual era usada para atribuir todas as atividades realizadas, como um “espelho” da *branch* “*master*”. Além disso, para iniciar o desenvolvimento de uma tarefa, primeiramente era criada uma *branch* com o nome iniciado por “*feat*”, em seguida do código da tarefa criada no software JIRA⁷ e o título da mesma.

⁴ <https://git-scm.com/>

⁵ <https://bitbucket.org/>

⁶ <https://about.gitlab.com/>

⁷ <https://www.atlassian.com/br/software/jira>

4.1.2 Revisão de código

A revisão de código era realizada por pares, ou seja, quando um desenvolvedor finalizava sua tarefa, ele criava um “*merge/pull request*”. Esse processo significa que o código fonte com a tarefa desenvolvida foi enviado para o software de hospedagem online e o desenvolvedor solicitou que a *branch* contendo o código seja mesclada com a *branch* “*dev*”, porém, para que isso seja possível, é necessário o aval de outros dois desenvolvedores.

As revisões de código eram realizadas para que a qualidade do software não fosse afetada, ou seja, caso um erro fosse encontrado ele poderia ser corrigido antes que o código fosse mesclado com a *branch* “*master*”.

4.1.3 Mentoria

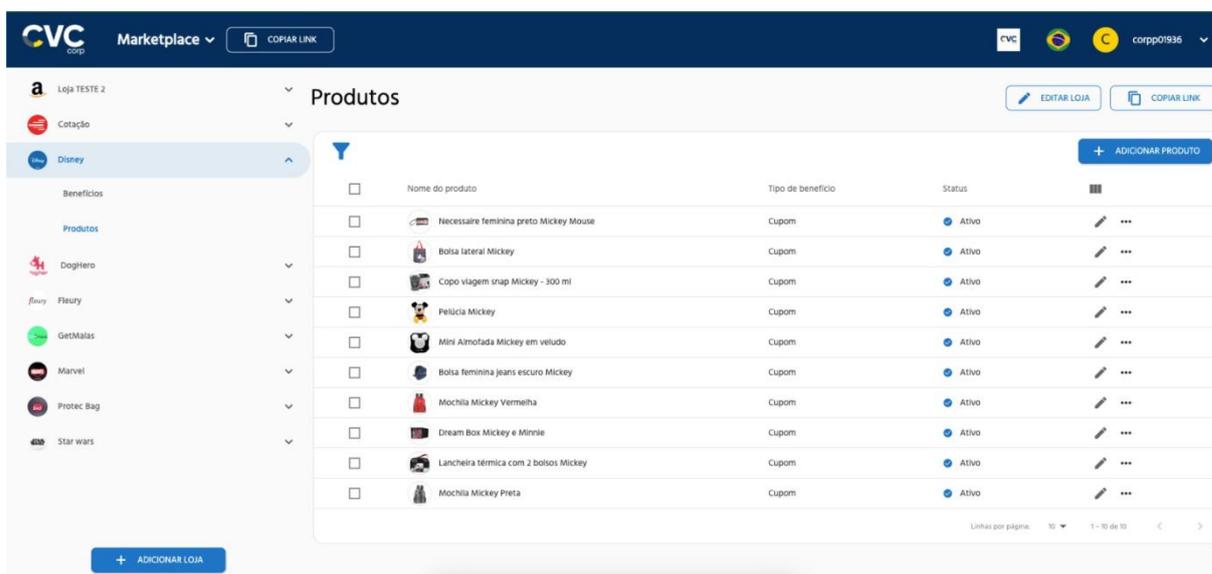
Antes de ser alocado em algum projeto dentro da empresa, o estagiário realizou duas semanas de *onboarding*, período em que os estagiários tiveram várias oficinas para conhecer a estrutura da empresa e os softwares utilizados no dia a dia de trabalho.

Após esse período, o estagiário foi alocado no projeto chamado Backoffice, do cliente CVC. Neste momento também foi designado um mentor, outro desenvolvedor com maior experiência, para auxiliar o estagiário no entendimento das atividades e da estrutura do projeto. O processo de mentoria durou seis meses e teve avaliações bimestrais, tanto do mentor quanto do mentorado.

4.2 Backoffice

Como citado na seção anterior, o primeiro projeto no qual o estagiário trabalhou, no período de agosto/2021 a dezembro/2021, foi o Backoffice Mobile que se trata de uma aplicação web, construída com o framework Flutter para o *front-end* e node para o *back-end*. O software se trata de uma interface para facilitar a criação, edição e remoção de lojas, produtos e cupons de desconto no *marketplace* contido no aplicativo CVC Viagens, por colaboradores do cliente que não possuem expertise para realizar esses procedimentos diretamente pelo console do projeto no Firebase.

Figura 3 - Página de listagem produtos



Fonte: Backoffice Mobile (2022)

Nesse projeto, o estagiário trabalhou especificamente com o cadastro, edição e remoção de produtos. Essas funcionalidades são de suma importância para o cliente, pois é dessa forma que o time de marketing da CVC manipula os produtos que serão apresentados no *marketplace* do aplicativo.

Para que os fluxos dessas funções fossem contemplados, o autor criou toda a estrutura necessária, ou seja, todas as camadas descritas na seção *Clean architecture*.

4.2.1 Cadastro de produto

Ao acessar o sistema Backoffice, em sua página inicial do módulo chamado “Marketplace”, o mesmo possui um menu lateral com a listagem das lojas e onde, também, é possível acessar as listagens de benefícios e produtos.

Para que fosse possível a criação de um produto primeiramente o autor criou a tela demonstrada nas Figura 4 e Figura 5. Os campos do formulário, “Nome do produto”, “Checkoute url”, “Descrição”, “Especificação”, “Cupom”, “Categoria”, “Tags” e “Imagens”, desta tela contemplam todos os dados necessários para a realização da requisição ao *back-end* para o cadastro de um produto.

Figura 4 - Página de cadastro de um produto (Parte 1)

The screenshot displays the 'Cadastrar produto' (Register product) page in the CVC Marketplace. The interface features a dark blue header with the CVC logo, 'Marketplace', and a 'COPIAR LINK' button. On the right side of the header, there are icons for CVC, a Brazilian flag, a yellow circle with a 'C', and the text 'corp01936'. The left sidebar lists various stores, with 'Disney' currently selected. The main content area is titled 'Cadastrar produto' and contains several input fields: 'Nome do produto', 'Checkout url', 'Descrição', and 'Especificação'. Below these is a 'Cupom' dropdown menu set to 'Selecionar'. At the bottom of the form is a 'Categorias e Tags' dropdown menu set to 'Categoria'. A '+ ADICIONAR LOJA' button is located at the bottom left of the sidebar area.

Fonte: Backoffice Mobile - CVC (2022)

Figura 5 - Página de cadastro de um produto (Parte 2)

This screenshot shows the second part of the product registration process. The 'Categorias e Tags' dropdown is set to 'Categoria'. Below it, there is a red 'Obrigatório' (Required) label and a 'Tag' dropdown menu. The 'Imagens' (Images) section features a dashed border around a central area containing a camera icon and the text 'Você pode arrastar e soltar arquivos aqui para adicioná-los.' (You can drag and drop files here to add them.) Below this text is the word 'ou' (or) and a '+ ADICIONAR' button. At the bottom of the page, there are two buttons: 'CANCELAR' and 'CADASTRAR PRODUTO'.

Fonte: Backoffice Mobile - CVC (2022)

Para que o fluxo dessa função fosse contemplado, o autor criou toda a estrutura necessária, ou seja, o *use case*, na camada *domain*, que recebe a *entity* criada e a envia para a

camada *data* onde a *entity* é transformada em *model* pelo *repository*. Após isso, a *model* é enviada para a camada *infra*, onde é realizada uma requisição do tipo POST ao *back-end*, que utiliza um *json* construído a partir da *model* citada anteriormente.

4.2.2 Edição de produto

A funcionalidade de edição de produto utiliza a mesma tela desenvolvida pelo autor para o cadastro de um produto, porém, nesse caso, os campos do formulário já vêm preenchidos com os dados do produto que se deseja editar.

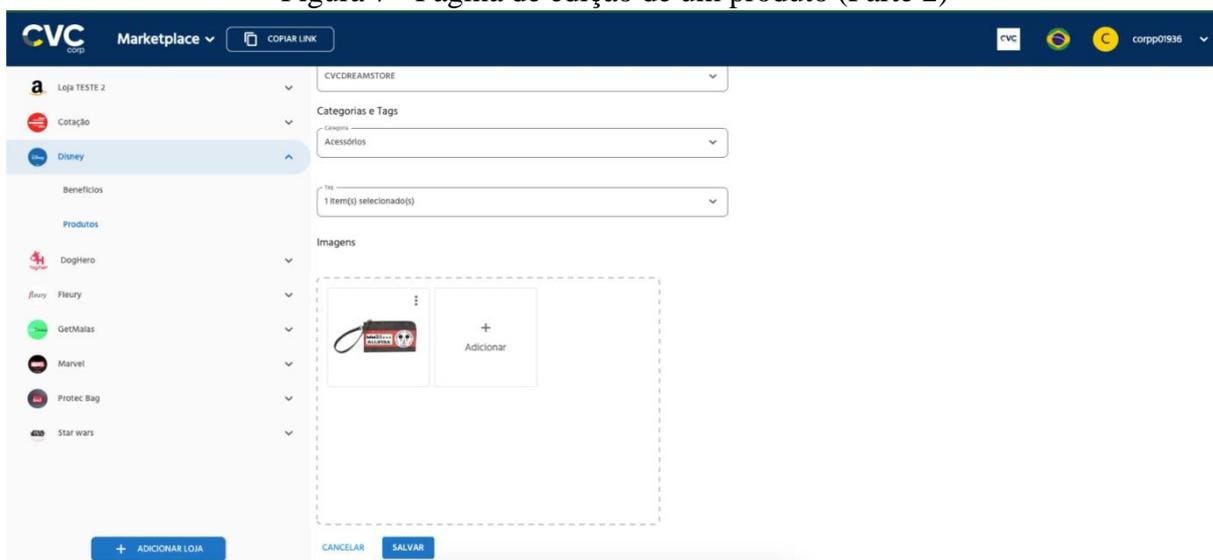
Conforme a Seção 4.2.1, a funcionalidade de edição de produto utiliza o mesmo tipo de *entity model*, visto que se trata de um objeto de mesmo tipo. As diferenças no fluxo ocorrem na camada *infra* onde é realizada uma requisição do tipo PUT e é utilizado o identificador do produto como *query string* na URL da requisição.

Figura 6 - Página de edição de um produto (Parte 1)

The screenshot displays the 'Cadastrar produto' (Add product) form in the CVC Marketplace. The interface includes a sidebar on the left with a list of store categories: Loja TESTE 2, Cotação, Disney, Benefícios, Produtos, Doghero, Fleury, GetMalas, Marvel, Protec Bag, and Star wars. The main form area is titled 'Cadastrar produto' and contains several sections: 'Informações do produto' with a 'Nome do produto' field containing 'Necessaire feminina preto Mickey Mouse' and a 'Check-out url' field containing 'https://www.lojadream.com/necessaire-feminina-preto-mickey-mouse/p'; a 'Descrição' field with a paragraph of text; an 'Especificações' field listing attributes: 'Altura (cm): 13,00', 'Largura (cm): 20,00', 'Comprimento (cm): 5,00', 'Peso (kg): 0,10', and 'Material: Couro Sintetico com Camurça'; a 'Cupom' field with a dropdown menu showing 'CVC DREAMSTORE'; and a 'Categorias e Tags' section with a dropdown menu showing 'Acessórios'. A 'COPIAR LINK' button is located in the top right corner of the header area.

Fonte: Backoffice Mobile - CVC (2022)

Figura 7 - Página de edição de um produto (Parte 2)

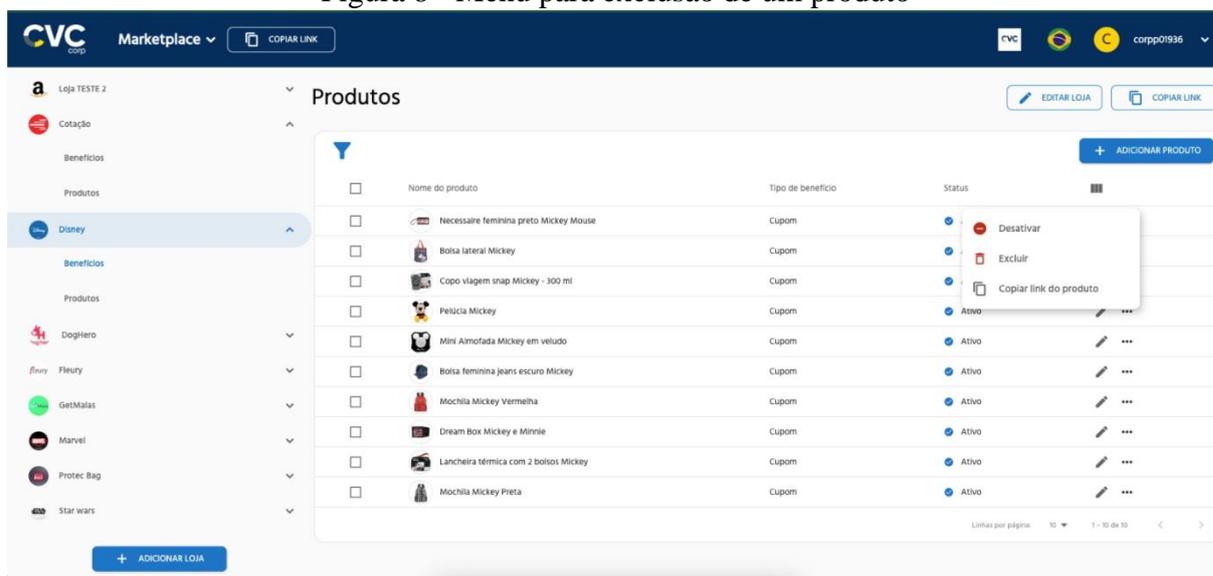


Fonte: Backoffice Mobile - CVC (2022)

4.2.3 Exclusão de produto

Há duas formas de excluir um produto na aplicação. A primeira delas, caso queira excluir um produto individualmente, basta clicar no ícone de “três pontos” na célula de um produto e clicar no botão excluir como demonstrado na Figura 8.

Figura 8 - Menu para exclusão de um produto

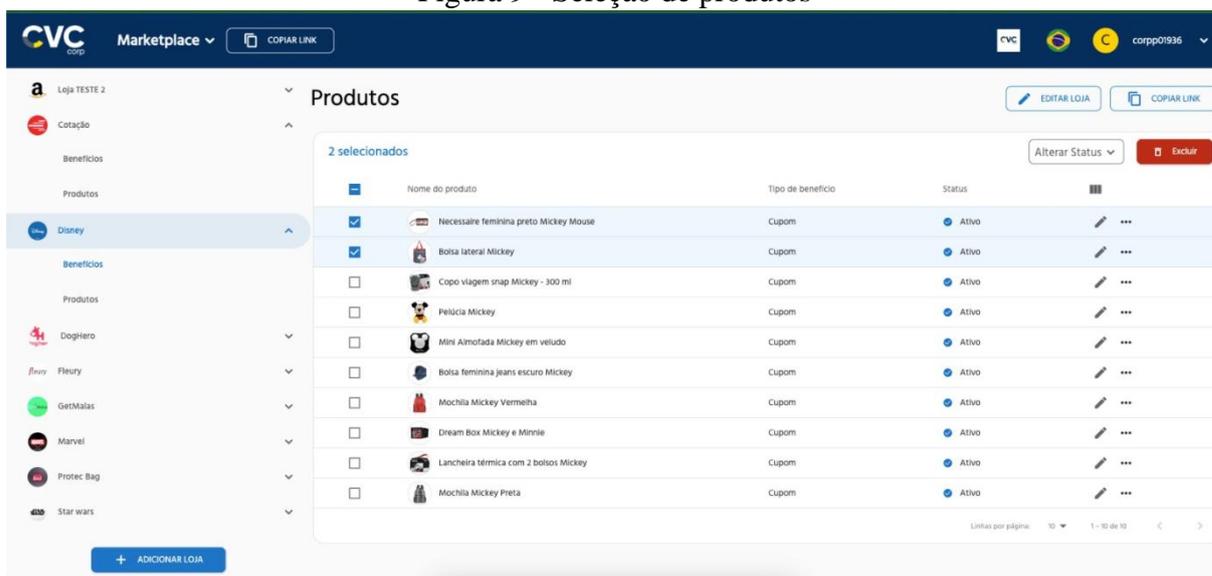


Fonte: Backoffice Mobile - CVC (2022)

O autor desenvolveu esse menu que contém o botão excluir e ao clicá-lo ele realiza uma requisição *http* ao *back-end* do tipo DELETE enviando o identificador do produto que se deseja excluir.

A segunda maneira de exclusão é quando se clica nas caixas de seleção dispostas no canto esquerdo da tabela, e após isso aparece o botão excluir no canto superior direito da tabela. O clique no botão chama uma função, passando por parâmetro uma lista do tipo *string* com os identificadores dos produtos que se deseja excluir, para que seja feita uma requisição do tipo PATCH.

Figura 9 - Seleção de produtos



Fonte: Backoffice Mobile - CVC (2022)

4.3 Reserva Fácil

Após a mudança na estrutura das equipes de trabalho, o autor foi designado para o projeto Reserva Fácil, o qual consistiu na manutenção do aplicativo de mesmo nome voltado aos agentes de viagem das franquias da CVC.

Pelo aplicativo os agentes de viagem conseguem realizar a venda de produtos como passagens aéreas, reservas em hotéis e aluguel de carros de modo fácil, pois não precisam estar na agência de viagens em que trabalham.

Nesse projeto o autor trabalhou principalmente na correção de *bugs*, mapeamento de erros e na realização de testes de Widgets, no período de janeiro/2022 a fevereiro/2022.

4.3.1 Correção de bugs

A primeira tarefa do autor no projeto Reserva Fácil foi mapear e corrigir *bugs* encontrados no aplicativo. Essa foi uma tarefa muito importante tanto para garantir a qualidade do software quanto para o autor entender melhor o projeto que foi inserido, uma vez que precisou percorrer todos os fluxos do aplicativo para análise, a fim de encontrar *bugs* e corrigi-los.

4.3.2 Mapeamento de erros

Outra tarefa que o autor desenvolveu foi o mapeamento de erros no aplicativo e a categorização deles em duas classes *front-end* e *back-end*, a fim de criar tarefas destinadas a cada erro encontrado.

No final foi construído um documento para apresentação no evento *Daily* para a ciência dos membros do time e para discutir possíveis soluções para os erros encontrados para garantir a qualidade do produto e, conseqüentemente, a experiência do usuário.

4.3.3 Testes de Widgets

Por fim, o autor desenvolveu a tarefa de implementar testes de *Widgets* da camada *presentation* do módulo de login. Essa foi uma tarefa muito importante para o projeto, pois serviu de base para as próximas implementações de testes de *Widgets* nos outros módulos do aplicativo, uma vez que o projeto ainda não contemplava a prática de implementação desse tipo de testes.

Visto que o termo teste de *Widgets* ainda não foi citado nesse documento é importante descrever sua funcionalidade para entendimento do leitor. Segundo a documentação do Flutter⁸, esse tipo de teste tem por objetivo verificar se a interface do usuário do *widget* parece e interage conforme o esperado. O teste de um widget envolve várias classes e requer um ambiente de teste que forneça o contexto de ciclo de vida do widget apropriado.

Esses testes contemplaram todos os fluxos possíveis no módulo de login:

- Login com usuário e senha corretos;
- Login com usuário e/ou senha incorretos;
- Alteração de senha;
- Login com outro usuário.

⁸ <https://docs.flutter.dev/testing/widget-tests>

Figura 10 - Reserva Fácil - Tela de login



Fonte: Reserva Fácil - CVC (2022)

4.4 Minha CVC e Submarino Viagens

O último projeto em que o autor foi membro da equipe de desenvolvimento durante o período de estágio foi o desenvolvimento e sustentação dos aplicativos Minha CVC e Submarino Viagens. A atuação do estagiário nesse projeto foi de março/2021 a setembro/2021.

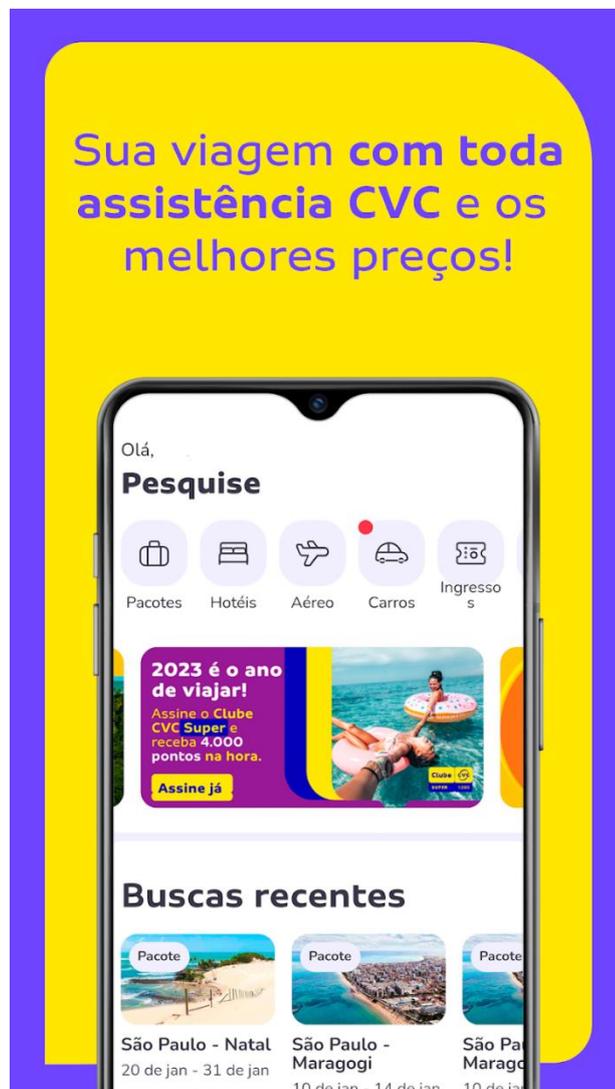
Como foi utilizado o *framework* Flutter e, com isso, o projeto necessita de apenas um código fonte para diversas plataformas, os aplicativos estavam disponíveis para Android e iOS. O modelo *whitelabel* foi utilizado neste projeto, pois dessa forma o time tinha total controle para gerar os ícones e o tema de cada produto, uma vez que são marcas diferentes e cada aplicativo possui sua identidade visual.

Esse projeto marcou a entrada da CVC no âmbito do mercado de venda de produtos por meio de dispositivos móveis, um passo muito importante, visto que durante a pandemia de coronavírus o setor digital se desenvolveu de forma estrondosa.

Principalmente pela necessidade de transformação digital dos empreendimentos para que os mesmos pudessem permanecer ativos e não sofrer tanto, ou até mesmo crescer, durante o período pandêmico onde muitos seguiam as orientações para ficarem em casa.

No aplicativo os usuários podem comprar passagens aéreas, realizar reservas em hotéis, alugar carros e, também, contam com um *marketplace* onde possuem descontos em parceiros da CVC. Na Figura 9 é mostrada a tela inicial do aplicativo Minha CVC.

Figura 11 - Minha CVC – Tela inicial



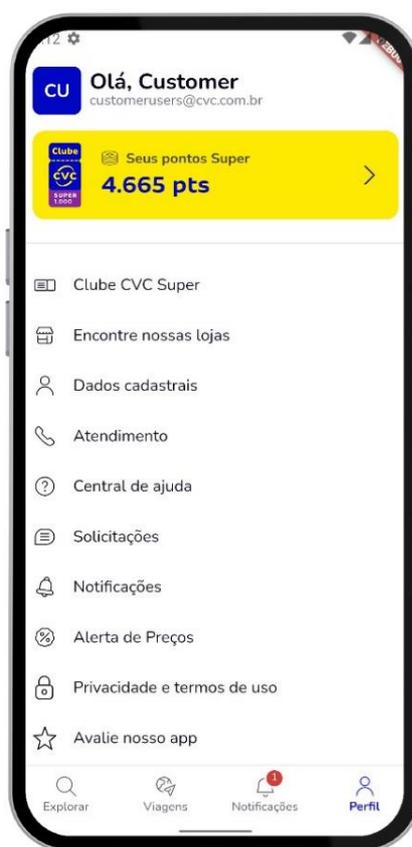
Fonte: Google Play Store (2022)

4.4.1 Central de ajuda

Após o estudo sobre o novo projeto ao qual foi inserido, o autor desenvolveu o módulo chamado “Central de ajuda”, que possui a finalidade de permitir ao usuário encontrar respostas para suas dúvidas, entrar em contato com os canais de ajuda e vendas da CVC e abrir um chamado para a equipe de suporte.

Para entrar no módulo “Central de Ajuda”, o usuário deve entrar na área de perfil pelo botão no canto inferior direito da tela e clicar em “Central de ajuda”, conforme é mostrado na Figura 12.

Figura 12 - Minha CVC – Perfil

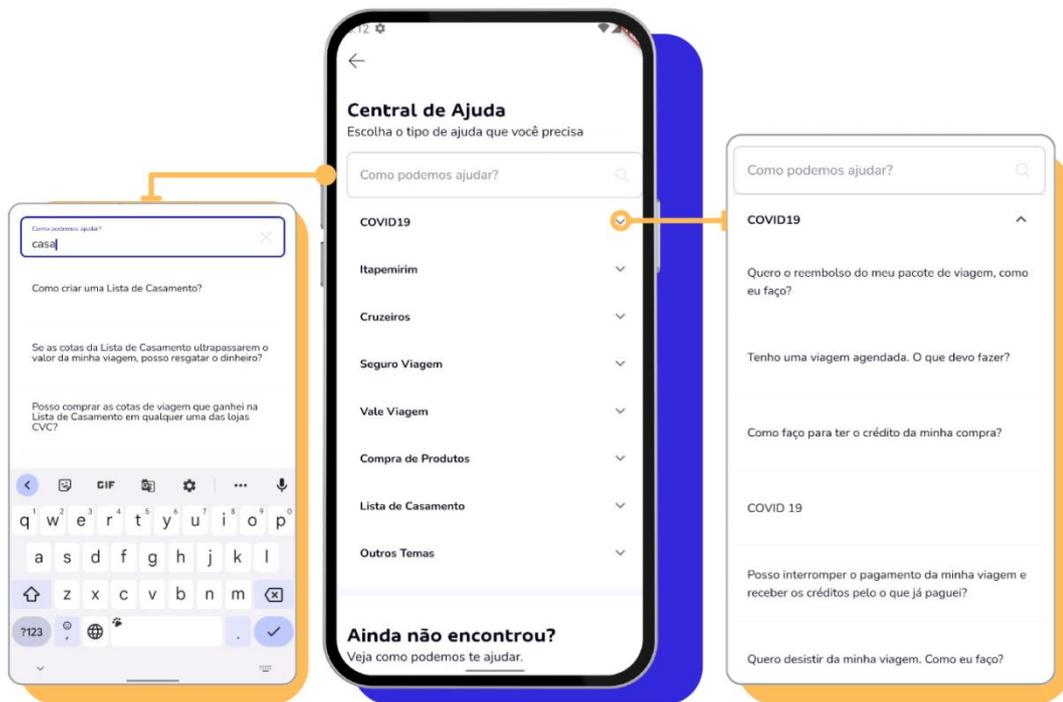


Fonte: Minha CVC (2022)

Ao entrar no módulo “Central de ajuda” o usuário terá uma lista expansível com temas definidos pela equipe de suporte por meio de métricas coletadas de dúvidas enviadas anteriormente. Os temas, perguntas e respostas são enviados pelo *back-end* e formatados pelo *front-end* para melhor visualização do usuário.

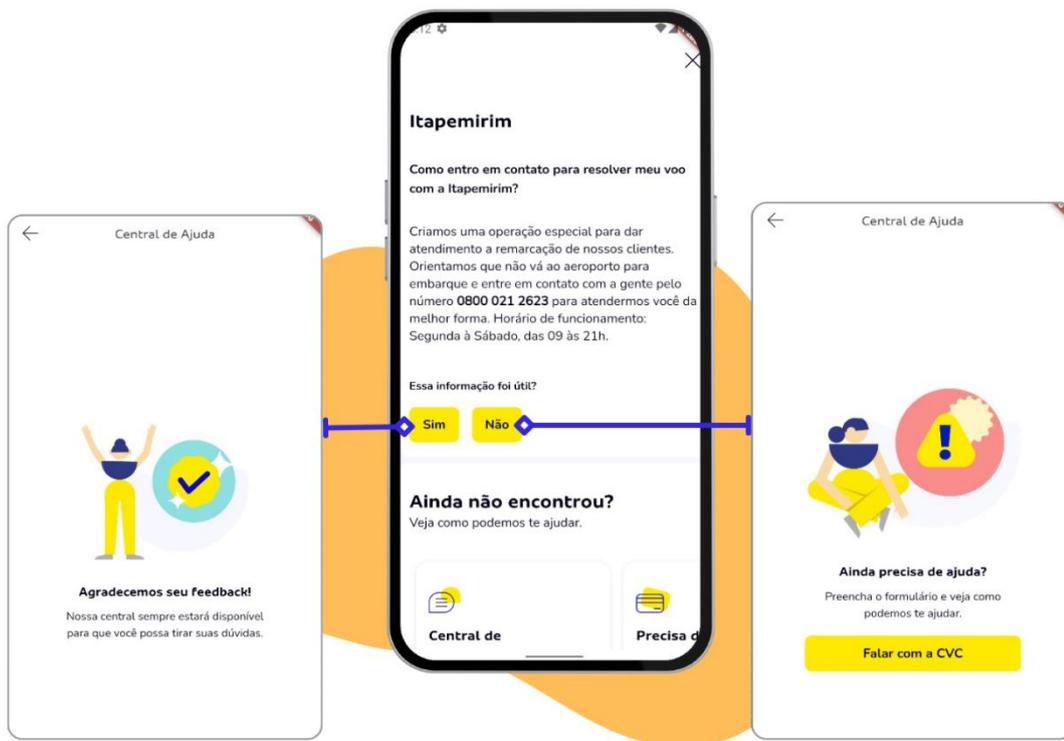
Conforme mostrado na Figura 13, há várias maneiras de se encontrar uma resposta para a dúvida do cliente, onde o mesmo pode pesquisar no campo de texto ou expandir algum tema para encontrar uma pergunta semelhante à sua.

Figura 13 - Minha CVC - Central de Ajuda



Fonte: Minha CVC (2022)

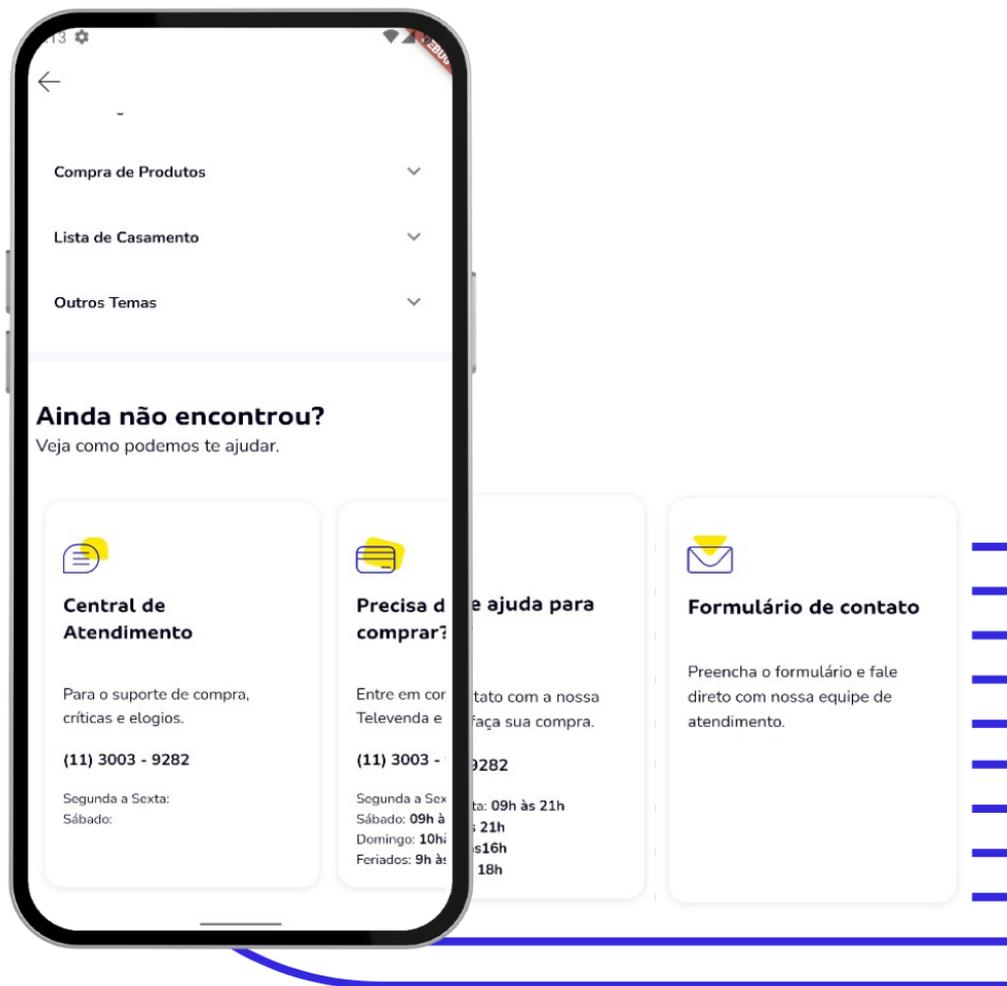
Figura 14 - Minha CVC –Feedback



Fonte: Minha CVC (2022)

Além do FAQ é possível entrar em contato com os canais de comunicação da CVC clicando em algum dos cartões da seção “Mais Ajuda”, conforme demonstrado na Figura 15.

Figura 15 - Minha CVC - Seção “Mais Ajuda”



Fonte: Minha CVC (2022)

Caso o usuário clique nos cartões de atendimento por meio de telefone, ele será redirecionado para o aplicativo de ligações do seu dispositivo já com o número na tela para realizar a ligação. Já, se o usuário clicar no cartão “Formulário de contato” ele será redirecionado para o módulo de abertura de chamado.

4.4.1.1 Abertura de chamado

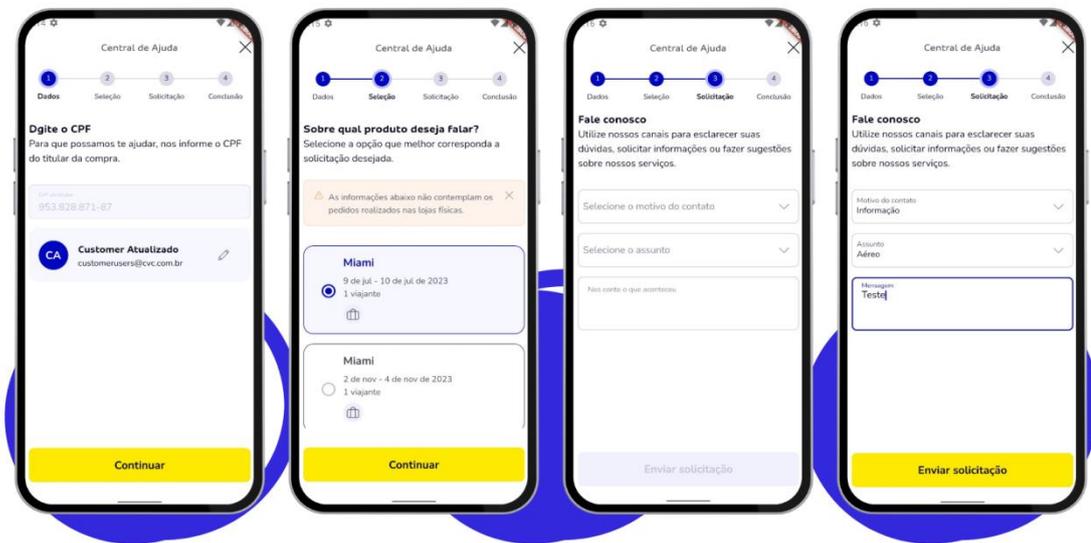
O autor desenvolveu o fluxo de abertura de chamado, o qual auxilia o cliente na resolução do problema, por meio de uma solicitação feita diretamente ao time de suporte da CVC para sanar dúvidas, fazer reclamações e, até mesmo, elogios à empresa.

Na tela inicial do módulo, o *front-end* já traz as informações de CPF, nome e e-mail do usuário. Nesta página há uma listagem das reservas realizadas pelo cliente de forma *online* e uma opção “Outros”, caso o problema não seja relacionado a alguma reserva do usuário ou, ele não tenha reserva.

Após a escolha de uma opção, o usuário deve preencher três campos para finalizar a abertura do chamado: um com o motivo, outro com o assunto e o último com uma descrição do problema a ser resolvido.

Por fim, vale ressaltar que se houver algum problema na requisição para abertura do chamado, seja por erro de conexão com a internet ou falha no *back-end*, os dados inseridos ficam salvos na *store* do módulo para posterior tentativa do usuário. Na Figura 16 é mostrado todo o fluxo citado nessa seção para a realização da abertura de chamado.

Figura 16 - Micha CVC - Abertura de Chamado



Fonte: Minha CVC (2022)

4.4.2 Vertical de carros

Outra tarefa desenvolvida pelo autor foi a construção do módulo nativo de aluguel de carros. Apesar desse módulo já existir, o fluxo era feito por meio de páginas da web, o que

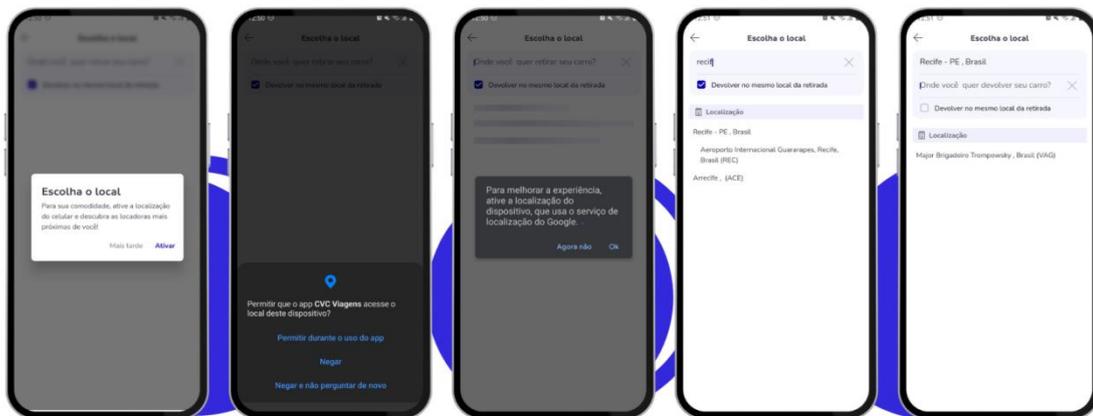
prejudicava a experiência do usuário, pois o carregamento era mais lento e não era possível tratar todos os cenários de erros.

Logo, com a finalidade de melhorar a experiência do usuário foi solicitado ao time que o autor integrava para que fosse desenvolvido uma vertical nativa - Flutter - para melhor fluidez e maior controle do fluxo por meio do time de desenvolvimento mobile.

Primeiramente, para realizar uma pesquisa, o usuário pode selecionar uma localização próxima que possua locadora de carros parceira da CVC, num raio de 40km, caso ele tenha permitido o acesso à sua localização, pois o sistema utiliza do serviço de geolocalização do dispositivo para encontrar locadoras no raio de distância citado. Outro modo é pesquisar manualmente no campo de texto a localização de origem e/ou de destino do usuário por meio de campos de texto.

Além dos fluxos citados, o usuário também pode selecionar se deseja retirar e devolver o carro alugado no mesmo local ou em outra locadora da marca.

Figura 17 – Vertical de Carros–Seleção de local de retirada e devolução



Fonte: Minha CVC (2022)

Após a seleção do local o usuário deve selecionar o período em que o carro será alugado na tela do calendário e os respectivos horários de retirada e devolução na tela seguinte. Para essa seleção foram usados os seletores de horário nativo para cada plataforma, nesse caso Android e iOS.

Figura 18 - Vertical de Carros - Seleção de data e horário



Fonte: Minha CVC (2022)

Depois de clicar no botão “Continuar” na tela de seleção de horário, o usuário será direcionado para a tela de carregamento da busca, a qual fará a requisição para o *back-end* com os dados preenchidos durante o fluxo exposto para que seja retornada uma lista de carros para o *front-end*, conforme mostrado na Figura 19.

Figura 19 - Vertical de Carros – Listagem de resultados



Fonte: Minha CVC (2022)

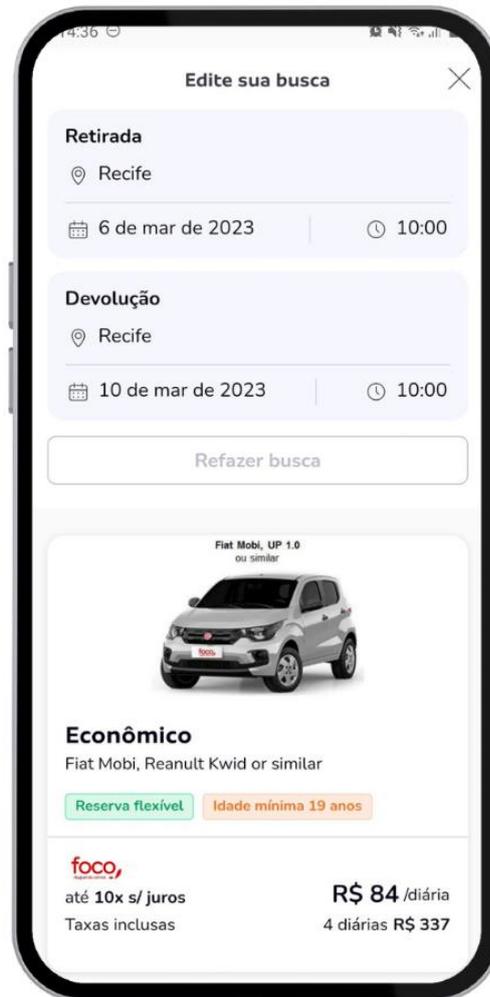
Na tela de resultado da busca, é possível editar a busca, ordenar a lista em ordem crescente e decrescente de preços, aplicar filtros rápidos - botões que estão à direita do botão de ordenação - e navegar para a página de detalhes de um veículo. Essas funcionalidades serão demonstradas nas seções seguintes.

4.4.2.1 Edição da busca

Para editar sua busca, o usuário deve clicar no botão “Editar” no canto superior esquerdo da tela de resultados. Após isso, aparecerá os campos conforme mostrado na Figura 20 para serem editados pelo cliente. Cada campo direciona o usuário para uma página, ou seja, o campo de localização para a tela de seleção do local, o campo de data para a tela de calendário e o

campo com as horas para a tela de seleção de horário. Todas essas telas já foram demonstradas nas Figuras 17 e 18.

Figura 20 - Vertical de Carros – Edição de busca



Fonte: Minha CVC (2022)

4.4.2.2 Tela de filtros

Na tela de filtros, o usuário pode customizar o resultado da busca por meio de diversos limitadores como:

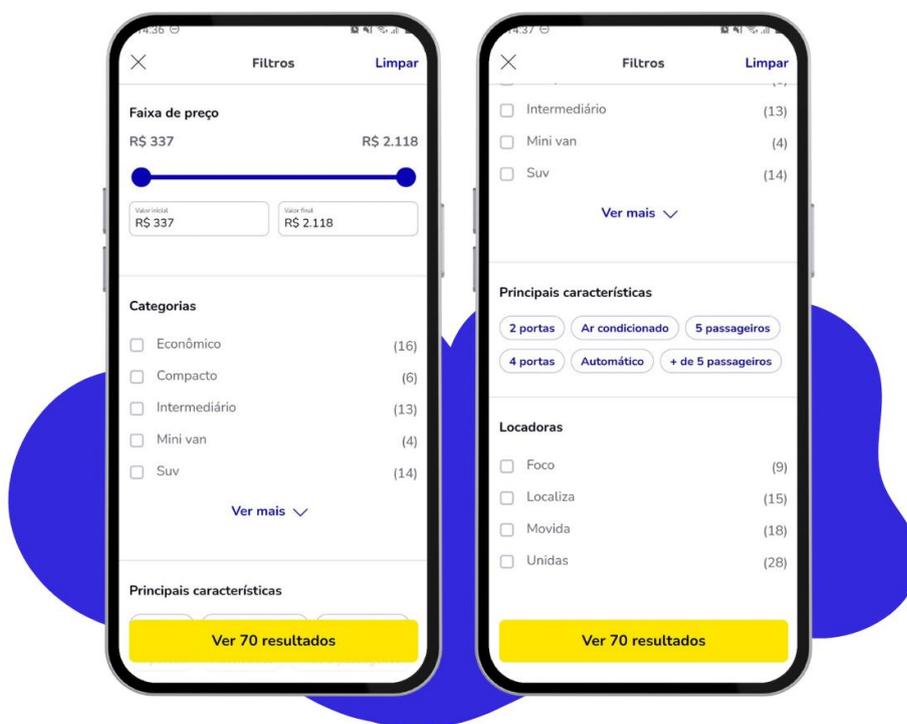
- Faixa de preço: onde o usuário pode arrastar os limites mínimos e máximos do delimitador de preço;
- Categorias: onde o usuário pode selecionar uma ou mais categorias da lista;
- Principais características: onde o usuário pode selecionar um ou mais *chips* com características dos veículos;

- Locadoras: onde o usuário pode limitar o resultado da busca a uma ou mais locadoras.

Após a seleção de um filtro pelo usuário será realizada uma chamada ao *back-end* que retornará a quantidade de carros disponíveis, conforme os delimitadores definidos pelo cliente. Esse valor retornado será utilizado para atualizar o botão localizado no inferior na página, com a nova quantidade de veículos, e quando esse botão for clicado, direcionará o usuário para a tela de resultados, porém com a lista atualizada de acordo com os filtros selecionados.

Todas essas seções da tela de filtros estão externadas na Figura 21.

Figura 21 - Vertical de Carros – Filtros



Fonte: Minha CVC (2022)

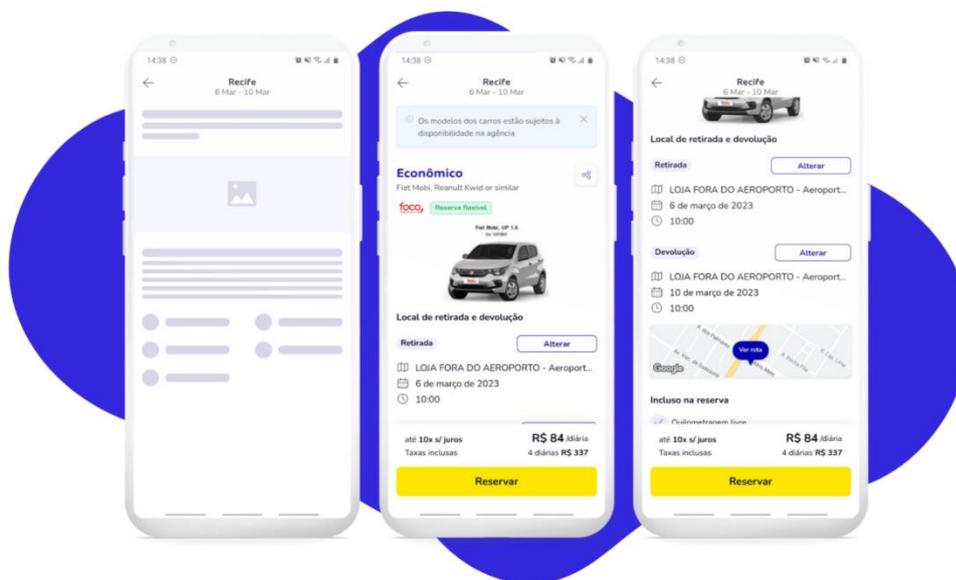
4.4.2.3 Tela de detalhes

Para melhor visualização do usuário, o autor implementou a tela de detalhes de um carro, na qual o cliente possui as informações de retirada e devolução, além da possibilidade de alteração deles. Ademais, a tela possui uma seção que consiste numa listagem das comodidades oferecidas pelo veículo escolhido. Em uma seção logo abaixo há uma lista com veículos

similares da mesma locadora para ofertar outras opções para o usuário e, por fim, as condições de pagamento e o preço total da locação do automóvel.

Vale ressaltar que a aplicação possui integração com o sistema para abrir o aplicativo padrão de visualização de mapas do dispositivo. Esse fluxo ocorre quando o usuário clica no mapa externado na Figura 22, onde o mesmo envia ao aplicativo o endereço ou as coordenadas da locadora, para auxiliar o cliente a localizá-la.

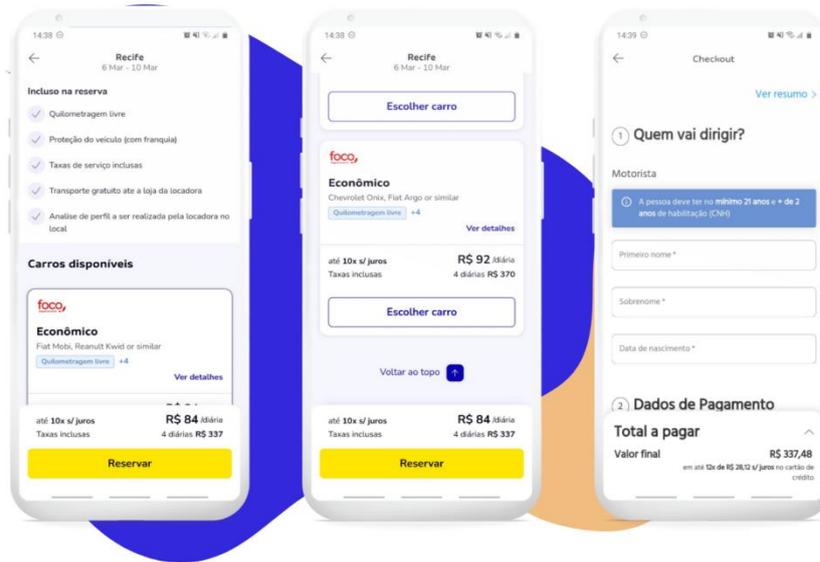
Figura 22 - Vertical de Carros – Detalhes do produto



Fonte: Minha CVC (2022)

Por último, ao clicar no botão “Reservar” o usuário é direcionado para a tela de *checkout*, a qual é uma página da web responsiva consumida pelo aplicativo. Essa tela pode ser visualizada na Figura 23.

Figura 23 - Vertical de Carros – Checkout



Fonte: Minha CVC (2022)

5 CONSIDERAÇÕES FINAIS

Em conclusão, o estágio realizado na empresa ioasys foi uma experiência enriquecedora que permitiu ao autor desenvolver habilidades técnicas e profissionais no desenvolvimento de aplicativos utilizando a tecnologia Flutter, além de adquirir conhecimentos valiosos sobre o setor de atuação da empresa. Ao longo do estágio, o autor teve a oportunidade de colaborar em projetos importantes e de trabalhar ao lado de profissionais altamente capacitados, o que proporcionou para ele uma visão mais ampla do mercado e das tendências do setor.

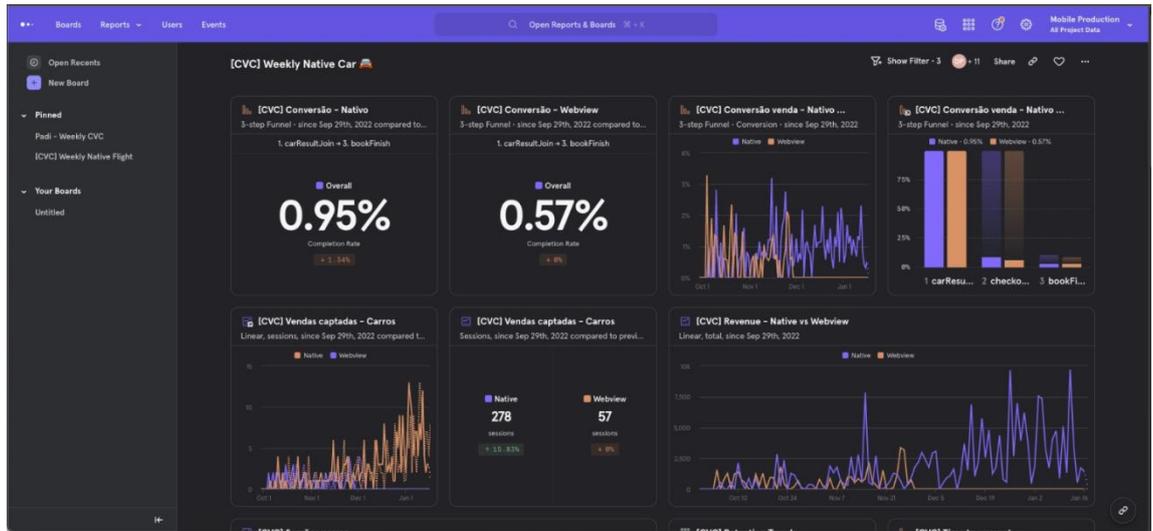
A empresa ioasys se destacou por ser uma organização inovadora, com uma cultura de trabalho colaborativa e aberta às novas ideias. Além disso, a empresa ofereceu um ambiente de trabalho acolhedor e incentivou o desenvolvimento profissional do autor ao longo do estágio.

Embora tenha enfrentado alguns desafios durante o estágio, o autor teve a oportunidade de aprender a lidar com situações adversas e de desenvolver sua resiliência. Ademais, a equipe da empresa esteve sempre disponível para oferecer suporte e orientação, o que contribuiu para que o autor se sentisse integrado e motivado durante todo o processo.

O Flutter é uma plataforma de desenvolvimento de aplicativos altamente flexível e eficiente, que oferece uma série de recursos avançados para criação de aplicativos para múltiplas plataformas. Ao longo do estágio, o autor pôde experimentar a tecnologia e constatar sua capacidade de agilidade e facilidade de uso.

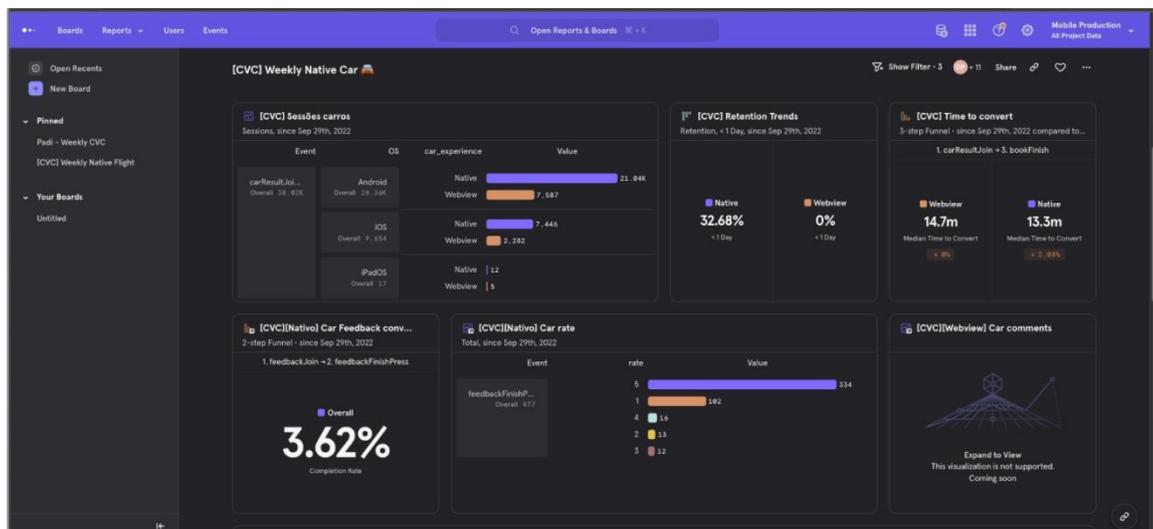
Isso é demonstrado através de alguns resultados obtidos das tarefas desenvolvidas pelo autor, juntamente com sua equipe. Por exemplo, na vertical de carros descrita na seção 4.4.2, houve um aumento significativo nos aluguéis de carros em comparação à versão anterior, pois a vertical nativa proporciona mais performance, mitigação de falhas e rapidez ao usuário, pontos que podem ser observados nas Figura 24 e Figura 25 em números, conforme abaixo, que demonstram quantitativamente que a prestação de serviços de aluguel de automóveis, aumentou significativamente na versão construída pelo autor, tendo um salto de aproximadamente 482% - diferença proporcional entre o número de aluguéis de carros entre as versões *webview* e nativa do aplicativo, circulada na Figura 24 - e qualitativamente em relação à otimização do tempo, que pode ser analisada na área delimitada da Figura 25, onde é mostrado que houve diminuição de 1.4 minutos no tempo para a realização do fluxo de aluguel de carro, estética do projeto e a forma como foi apresentado, a qual despertou o interesse dos usuários.

Figura 24 - Vertical de Carros – Resultados (Parte 1)



Fonte: Mixpanel CVC (2022)

Figura 25 - Vertical de Carros – Resultados (Parte 2)



Fonte: Mixpanel CVC (2022)

Portanto, o estágio na empresa ioasys foi uma excelente oportunidade para o autor se desenvolver profissionalmente, adquirir conhecimentos valiosos sobre o mercado de trabalho e proporcionou uma visão ampla sobre o desenvolvimento de aplicativos utilizando Flutter. Certamente as habilidades e conhecimentos adquiridos durante o estágio serão úteis na carreira futura do autor.

Reafirmando o que foi descrito no início desse trabalho em relação ao curso de Ciência da Computação, este forneceu ao autor os conhecimentos necessários, conteúdos de aprendizagem essenciais e ferramentas para a realização e conclusão do estágio. Isso foi fundamental para o crescimento acadêmico, profissional e pessoal do autor, fundamentando e alicerçando a teoria recebida em sala de aula, numa perspectiva mais ampla de forma crítica, reflexiva, prática e ponderada.

REFERÊNCIAS

BALARDIM, Eduardo. **Lean Manufacturing**: o que é, objetivos e princípios. O que é, Objetivos e Princípios. 2019. Disponível em: <https://fia.com.br/blog/lean-manufacturing/>. Acesso em: 26 dez. 2022.

BECK, Kent. et al. **Manifesto para Desenvolvimento Ágil de Software**. 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 26 dez. 2022.

DART. **Dart documentation**. Disponível em: <https://dart.dev/guides>. Acesso em: 25 dez. 2022.

FLUTTER. **Flutter documentation**. Disponível em: <https://docs.flutter.dev/>. Acesso em: 26 dez. 2022.

GIT. **Git**. Disponível em: <https://git-scm.com/>. Acesso em: 24 de dez. de 2022.

GITLAB. **Gitlab**. Disponível em: <https://docs.gitlab.com/>. Acesso em: 24 de dez. de 2022.

IOASYS. **ioasys**. Disponível em: <https://ioasys.com.br/sobre>. Acesso em: 26 dez. 2022.

MARTIN, Robert C. **Arquitetura limpa**: O guia do artesão para estrutura e design de software. [S. l.], Alta Books, 2019.

MARTIN, Robert C.. **Principles of OOD**. Disponível em: <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>. Acesso em: 26 dez. 2022.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide**: the definitive guide to scrum: the rules of the game. The Definitive Guide to Scrum: The Rules of the Game. 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>. Acesso em: 26 dez. 2022.

SOUTO, Mario. **Front-end, Back-end e Full Stack**. 2023. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>. Acesso em: 03 jan. 2023.

VACANTI, Daniel; COLEMAN, John. **THE KANBAN GUIDE**. 2020. Disponível em: <https://kanbanguides.org/wp-content/uploads/2021/01/Kanban-Guide-2020-12.pdf>. Acesso em: 28 dez. 2022.

VALENTE, Marco Tulio. **Engenharia de Software Moderna**. Princípios e Práticas para Desenvolvimento de Software com Produtividade, Editora: Independente, 395 páginas, 2020.

WELLS, Don. **Extreme Programming: a gentle introduction**. A gentle introduction. Disponível em: <http://www.extremeprogramming.org/>. Acesso em: 26 dez. 2022.