

Account Center: Plataforma Gerenciadora de Softwares Serviços

Alisson Alves Litvin, Giovane Galvão

Centro Universitário Campo Real
Rua Comendador Norberto, 1299 - Santa Cruz – Guarapuava – PR – Brasil
{engs-alissonlitvin@camporeal.edu.br, prof_giovanegalvao@camporeal.edu.br}

Abstract: The growing demand for complete solutions in the Software as a Service market, driven by the continuous expansion of this sector, generated the need for a tool that simplifies the use of these services. This artifact centralizes crucial data for operational functioning more effectively, connecting the company's needs directly to customer requirements. The Account Center was developed using agile methodologies, in this case Scrumban, which aims to reduce time and increase development flexibility. Along with this, it is worth highlighting that the structure chosen to set up the project also helped in these aspects, which are the use of an architecture monolithic and modular code standard, thereby creating a platform that enables financial administration between service providers and the customers who subscribe to them. This system was designed to create a technological ecosystem where managers can view and control variables related to these services in a highly efficient way.

Resumo: A crescente demanda por soluções completas no mercado de *Software as a Service*, impulsionada pela expansão contínua desse setor, gerou a necessidade de uma ferramenta que simplifica o uso desses serviços. Esse artefato centraliza dados cruciais para o funcionamento operacional de forma mais eficaz, conectando as necessidades da empresa diretamente às exigências dos clientes. O *Account Center* foi desenvolvido utilizando metodologias ágeis, neste caso o *Scrumban* que visa reduzir o tempo e aumentar a flexibilidade de desenvolvimento. Junto a isso, vale ressaltar que a estrutura escolhida para montar o projeto também auxiliou nesses quesitos que são a utilização de uma arquitetura monolítica e um padrão de código modular, com isso concebendo uma plataforma que viabiliza a administração financeira entre os provedores de serviços e os clientes que subscrevem-os. Este sistema foi concebido para criar um ecossistema tecnológico onde gestores podem visualizar e controlar variáveis relacionadas a esses serviços de maneira altamente eficiente.

LISTA DE ABREVIATURAS E SIGLAS

<i>SaaS</i>	<i>Software as a Service;</i>
<i>API</i>	<i>Application Programming Interface (Interface de Programação de Aplicação);</i>
<i>IP</i>	<i>Internet Protocol (Protocolo da Internet);</i>
<i>PHP</i>	<i>Hypertext Preprocessor(Pré-Processador de Hipertexto);</i>
<i>SPA</i>	<i>Single Page Applications(Aplicativos de Página Única);</i>
<i>BD</i>	<i>Banco de Dados;</i>
<i>SQL</i>	<i>Structured Query Language(Linguagem de Consulta Estruturada);</i>
<i>PaaS</i>	<i>Platform as a Service;</i>
<i>HTML</i>	<i>HyperText Markup Language(Linguagem de Marcação de HiperTexto);</i>
<i>CSS</i>	<i>Cascading Style Sheets(Folhas de Estilo em Cascata)</i>
<i>MIT</i>	<i>Massachusetts Institute of Technology(Instituto de Tecnologia de Massachusetts);</i>
<i>SPA</i>	<i>Single Page Application(Aplicativo de Página Inicial)</i>
<i>SSR</i>	<i>Server-side Rendered Application(Aplicativo Renderizado no lado do Servidor);</i>
<i>PWA</i>	<i>Progressive Web Application(Aplicativo Web App Progressivo)</i>
<i>TIC</i>	<i>Tecnologia da Informação e Comunicação(Information and Communication Technology);</i>
<i>CC</i>	<i>Cloud Computing(Computação em Nuvem);</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol(Protocolo de Transferência de Hipertexto);</i>
<i>QP</i>	<i>Questões de Pesquisa.</i>

1. Introdução

Com a constante expansão do mercado de *Software as a Service* (SaaS) (SUPERLOGICA, 2021), surge uma demanda por uma solução abrangente que possa supervisionar e organizar essa variedade de sistemas. Essa necessidade transcende a simples gestão de acessos, abrangendo também a criação de novas abordagens para a utilização desses produtos, estabelecendo um ambiente diversificado capaz de atender às distintas exigências dos usuários.

À medida que as organizações expandem suas operações, adotam uma variedade de SaaS para atender a diferentes necessidades internas, desde ferramentas de colaboração até soluções de análise de dados. Essa diversidade de serviços, muitas vezes, leva à fragmentação do controle, dificultando o monitoramento eficiente de custos, acessos e utilização (B2B STACK, 2023).

A ausência de uma solução centralizada para gerenciar assinaturas de SaaS resulta em processos manuais demorados, falta de visibilidade global sobre os custos de assinatura e possíveis violações de segurança. Além disso, a falta de um mecanismo para adaptar rapidamente as assinaturas às necessidades em constante mudança das empresas pode levar a gastos desnecessários e ineficiências operacionais (B2B STACK, 2023).

A motivação central que impulsionou o desenvolvimento do Account Center foi a identificação de uma lacuna substancial. Empresas que possuem numerosos SaaS enfrentam um desafio crucial: a ausência de um método unificado para administrar o funcionamento dessas instâncias. Isso engloba não apenas o controle de acessos, mas também a monitorização da utilização dos serviços, a análise do rendimento financeiro de cada serviço de maneira global e individualizada (ACCESSTAGE, 2022), bem como o rastreamento detalhado dos momentos e procedimentos de subscrição por parte dos clientes.

O cerne deste sistema reside na criação de uma plataforma que viabilize a administração financeira entre os SaaS e os clientes que efetuam a assinatura desses serviços, toda essa descrição pode ser resumida em ecossistema de SaaS ou ecossistema tecnológico, de forma mais ampla. Com isso em mente, foi concebida uma estrutura que concede aos gestores (ou administradores) a capacidade de visualizar e controlar essas variáveis, assegurando um funcionamento harmonioso de todas as operações.

O planejamento dessa estrutura, juntamente com o *design* da sua implementação e funcionamento, foi meticulosamente concebido para possibilitar um desenvolvimento ágil dentro de um período limitado. A adaptabilidade desempenha um papel fundamental, garantindo que o sistema seja ágil e flexível o suficiente para responder às mudanças dinâmicas do ambiente operacional.

1.1 Objetivos

O objetivo principal da pesquisa e da construção da plataforma Account Center é viabilizar a administração financeira entre os SaaS e os clientes que efetuam a assinatura desses serviços.

Como objetivos específicos a plataforma engloba:

- Criação de uma plataforma que mantenha usuários, produtos (SaaS), planos e assinaturas para gestão;
- Manter uma plataforma simples e de fácil uso, flexível e que atenda o objetivo de forma direta, sem muitas complicações em questões de armazenamento de registros e pagamentos;
- Administrar e Gerenciar SaaS para centralização de processos financeiros (ACCESSTAGE, 2022).

1.2 Organização do Trabalho

O projeto foi organizado em 3 principais etapas, sendo elas: Definições de Regras de Negócio, que é a etapa onde foi formulado como a plataforma agiria e de que forma a plataforma realizaria todas suas ações, Estruturação do Projeto, etapa onde foi definido todo tipo de metodologia/tecnologia que seria utilizada e o Desenvolvimento, que se trata da criação de todo o fluxo para no fim a ideia se tornar um produto, cada uma desempenhando um papel fundamental para realização do projeto.

2. Referencial Teórico

O cerne de uma plataforma de gestão de SaaS reside na aceleração e na conveniência do uso de diversas ferramentas, eliminando a necessidade de criar múltiplas contas ou procurar diferentes plataformas em locais distintos. Inspirada por empresas renomadas como Google, Microsoft e Blizzard, essa abordagem de negócios foi assimilada, transformando-se em uma plataforma que atende a diversas necessidades simultaneamente por intermédio de distintas SaaS (SUPERLOGICA, 2021).

Nesta seção serão apresentados os conceitos teóricos relacionados à temática deste trabalho, tópicos como: modelo de negócios SaaS, gerenciamento de assinaturas, controle de acesso e segurança e desenvolvimento de aplicações web serão apresentados em seus conceitos.

2.1. Modelos de Negócios SaaS e a Evolução das Assinaturas

SaaS é uma forma de disponibilizar *softwares* e soluções de tecnologia como um serviço. Com esse modelo, sua empresa não precisa instalar, manter e atualizar *hardwares* ou *softwares*, pois esses são executados nos servidores das empresas provedoras, que têm a responsabilidade de gerenciar o acesso e manter a estrutura de segurança de dados, conectividade e servidores necessários para o serviço (SALESFORCE, 2023).

Dado o contexto apresentado, pode-se dizer que as grandes vantagens de se trabalhar com assinaturas para SaaS é, além da centralização dos SaaS utilizados, também uma forma melhor de gerenciar os usuários e suas finanças por meio de uma única plataforma. Um dos problemas de se ter um sistema que centralize esses serviços é que os mesmos terão que ter alguma forma de se comunicar entre si para estarem alinhados para que todo o fluxo ocorra sem interrupções (SALESFORCE, 2023).

O impacto desse modelo de negócio seria muito grande a longo prazo muito pelo número de empresas que já possuem diversos SaaS e não tem um tipo de estrutura como essa, como também por se tratar de um novo método de negócio para as empresas.

2.2. Gerenciamento de Assinaturas em Ambientes Corporativos

O gerenciamento de assinaturas é importante tanto para ter uma segurança de que aquele serviço ainda está ou não em vigor para determinado(s) usuário(s), como também para se manter um histórico para eventuais relatórios ou pesquisas da própria empresa (MULTIEDRO, 2023).

A gestão das assinaturas até o momento possui apenas um problema evidente até o momento, que é ser única e exclusivamente de um usuário para um SaaS, algo que pode ser melhorado para que vários usuários utilizem da mesma assinatura, quanto ao número de registros, não será um grande problema por conta de que, quando uma assinatura expira, não se é criada outra, mas sim reativada quando o mesmo usuário assina o mesmo SaaS novamente, sendo assim os números de assinaturas na base de dados diminui consideravelmente (MULTIEDRO, 2023).

A boa gestão de assinaturas é uma vantagem competitiva frente aos seus concorrentes. Por mais que o produto seja bom, nenhum cliente vai aceitar cobranças indevidas, atrasos em ativações e renovações ou ainda um processo burocrático demais de pagamento (ASAAS, 2013), vantagens de um sistema que possui gestão de assinatura:

- **Automatização do controle de clientes por meio da plataforma de recorrência:** Automatizar o controle dos clientes. A integração direta com o pagamento permite que o processo de ativação, bloqueio ou cancelamento de assinaturas seja feito de forma automática.
- **Acompanhamento das segmentações dentro do gerenciador de assinaturas:** O acompanhamento das segmentações dentro do banco de assinantes. Se é oferecido diversos planos de assinatura pode saber exatamente quantos optaram pelo plano mais básico ou pelo mais completo, por exemplo. Isso permite que possa entrar em contato direto com cada um deles, enviando *e-mails* para avisar sobre promoções, upgrades e renovações.
- **Gestão de recorrência por meio de relatórios gerenciais:** Torna possível a geração de relatórios gerenciais com o balanço completo das operações em tempo real, facilitando muito o controle financeiro. Com a vantagem de que todas as informações ficam guardadas em segurança na nuvem e podem ser acessadas de qualquer lugar.

2.3. Controle de Acesso e Segurança em SaaS

O controle de acesso e a segurança desempenham um papel fundamental na garantia da integridade, confidencialidade e disponibilidade dos dados em soluções SaaS. Como os SaaS são *softwares* no qual os aplicativos são hospedados na nuvem e acessados por meio da internet, torna-se o controle de acesso e a segurança críticos, uma vez que os dados e aplicativos são acessados remotamente, muitas vezes por meio de dispositivos variados e redes não confiáveis (GETASTRA, 2023).

Aqui estão alguns aspectos importantes relacionados ao controle de acesso e segurança em SaaS:

1. **Autenticação:** Garantir que apenas usuários autorizados tenham acesso ao SaaS é fundamental. Isso envolve a implementação de medidas de autenticação robustas, como senhas fortes, autenticação de dois fatores (2FA), autenticação única (SSO) e integração com serviços de diretório.
2. **Autorização:** Uma vez autenticado, um usuário deve ter permissões apropriadas para acessar e modificar os dados. As permissões devem ser bem definidas e configuradas para garantir que os usuários só acessem as informações necessárias para suas funções.
3. **Criptografia:** A comunicação entre o cliente e o servidor, bem como os dados armazenados no SaaS, devem ser criptografados para proteger contra acessos não autorizados. O uso de protocolos seguros, como HTTPS, é essencial.
4. **Controle de versão e atualizações:** Garantir que o SaaS seja mantido atualizado com remendos de segurança e atualizações é fundamental para mitigar vulnerabilidades conhecidas.
5. **Políticas de segurança:** Estabelecer políticas de segurança claras e comunicá-las aos usuários é fundamental. Isso inclui diretrizes para o uso seguro do SaaS, bem como ações a serem tomadas em caso de violação de segurança.

Resumindo os SaaS já possuem suas próprias metodologias de segurança e acesso, sejam elas por *login* (cadastro e senha), chaves de API (gerada a partir de uma combinação do seu *hardware* e IP) ou *OAuth* (utilização de *tokens* de autorização), um padrão aberto de delegação de autorização, comumente utilizado para permitir que os usuários da Internet possam fazer login em sites de terceiros usando contas de um provedor de identidade (INTERGATE, 2023).

Na plataforma do Account Center, a metodologia de acesso e segurança se dá a partir da autenticação que ocorre via login e geração de token de acesso, quando esse usuário assina algum SaaS já é disparado uma requisição que faz com que esse mesmo usuário seja criado ou autenticado neste SaaS, a autenticação é responsabilidade do SaaS, o que o Account Center faz é emitir essa requisição e aguardar a resposta.

O acesso em larga escala não deverá ser um problema já que no Account Center não possui requisições grandes que requerem muito processamento para ocorrerem, uma vez que, tudo que o SaaS assinado faz, ocorre do lado do servidor do mesmo, sem nenhuma ligação com o servidor do Account Center.

2.4. Tecnologias de Desenvolvimento Web

A concepção do Account Center priorizou a agilidade e simplicidade, orientando a escolha de tecnologias que viabilizassem um desenvolvimento eficiente. Nesse sentido, foi decidido utilizar o *framework* VueJs, especificamente o Quasar, visando a agilidade na construção de *layouts* e *templates*.

A API foi construída exclusivamente em Laravel (PHP), enquanto o AXIOS foi adotado para possibilitar a perfeita integração com a API. Quanto ao gerenciamento de banco de dados, a opção recaiu sobre o MySQL, um sistema que emprega a linguagem SQL como interface.

Essa seleção de tecnologias converge para um processo ágil e simplificado de desenvolvimento, alinhando-se com os objetivos do projeto Account Center. A seguir, apresenta-se explicações mais detalhadas sobre cada uma dessas tecnologias.

VueJs: É um *framework* JavaScript para criar interfaces de usuário. Ele se baseia em HTML padrão, CSS e JavaScript e fornece um modelo de programação declarativo e baseado em componentes que ajuda você a desenvolver com eficiência interfaces de usuário, sejam elas simples ou complexas (VUEJS, 2023; VUEJS3COOKBOOK, 2020).

Quasar: É uma estrutura baseada em Vue.js de código aberto licenciada pelo MIT, que permite que você, como desenvolvedor web (QUASAR, 2023), crie rapidamente sites / aplicativos responsivos como, SPAs, SSR, PWAs, Aplicativos móveis (Android, iOS) através da Cordova ou Capacitor, Aplicativos de desktop multiplataforma.

Laravel: É um *framework* de aplicação web com sintaxe expressiva e elegante. Um *framework* web fornece uma estrutura e ponto de partida para criar seu aplicativo. O Laravel se esforça para fornecer uma experiência incrível para desenvolvedores, ao mesmo tempo em que fornece recursos poderosos, como injeção de dependência completa, uma camada de abstração de banco de dados expressiva, filas e trabalhos programados, testes de unidade e integração e muito mais (LARAVEL, 2023; DEVLARAVEL, 2023).

MySQL: É um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface. É atualmente um dos sistemas de gerenciamento de bancos de dados mais populares da *Oracle Corporation*, com mais de 10 milhões de instalações pelo mundo (MYSQL, 2023).

3. Estado da arte

Nesta seção, mostrará os resultados de um mapeamento sistemático da pesquisa executado com relação a temática do sistema AccountCenter. O objetivo da aplicação da técnica é apresentar trabalhos relacionados a esta pesquisa.

3.1 Metodologia de Pesquisa

O Mapeamento Sistemático (MS) é uma revisão ampla dos estudos primários existentes em um tópico de pesquisa específico que visa identificar a evidência disponível neste tópico. Assim, um MS é um estudo secundário que tem como objetivo identificar e classificar a pesquisa relacionada a um tópico amplo de pesquisa (KITCHENHAM, CHARTERS, 2007).

Os resultados de um MS ajudam a identificar lacunas nesta área, capazes de sugerir pesquisas futuras e prover um guia para posicionar adequadamente novas atividades de pesquisa (KITCHENHAM, CHARTERS, 2007). Assim, MSs visam prover uma visão geral de um tópico e identificar se há subtópicos nos quais mais estudos primários são necessários.

Este mapeamento sistemático foi baseado no método de Petersen (PETERSEN, 2008) que é composto por cinco etapas. A primeira delas é a definição das questões que devem ser respondidas ao final da pesquisa. A segunda etapa é a construção das *strings* de busca e de como será realizada a condução da pesquisa, incluindo nessa fase a definição das bases que serão consultadas. A terceira etapa realiza a análise dos artigos consultados. A quarta etapa classifica os artigos conforme as categorias escolhidas para o mapeamento, e por fim, a última fase tem a finalidade de realizar a extração dos dados.

Toda coleta de dados veio através das documentações das linguagens ou *frameworks* utilizados no projeto e/ou de artigos que analisam de forma econômica/empresarial o mercado de sistemas serviços como uma forma de inovação.

3.1.1 Questões de Pesquisa

As questões de pesquisa (QP) consideradas nesse mapeamento foram:

- QP1: Qual a necessidade encontrada no mercado para construção desse modelo de plataforma?
- QP2: Quais são os principais benefícios financeiros que uma empresa pode obter ao adotar uma plataforma de gerenciamento de *software* serviços, em termos de economia de custos e otimização de recursos?
- QP3: Qual a vantagem de centralizar seus SaaS em uma plataforma de gerenciamento?

3.1.2 Strings de Busca

O Quadro 1 apresenta as expressões para a condução desse mapeamento. A *string* implementada levando-se em consideração as palavras-chaves principais relacionadas ao tema e aquelas recorrentes, conforme leituras em artigos científicos.

Quadro 1 - *Strings* de busca

String	Expressão
1	((“software as a service”) AND (“benefits” OR “perspectives and challenges” OR “advantages”))

Fonte: O autor, 2023.

3.1.3 Análise dos Artigos

A busca não foi especificamente em um modelo de negócio que gerencia SaaS, pois um artigo com informações como essa não foram encontrados, então as buscas se limitaram aos próprios SaaS que retornou cerca de 58.200 artigos, porém a grande maioria eram artigos que não abordavam o tema ou só era mencionado algo e o assunto não era de fato relevante para essa pesquisa. Ao todo 5 artigos foram levados em consideração para a extração dos dados.

A pesquisa feita nos artigos, leva em consideração alguns critérios tanto para inclusão do artigo como sendo válido a pesquisa, como sendo inválido, são eles os citados abaixo:

Critérios de Inclusão:

- Artigos escritos em qualquer língua;
- Artigos publicados acima de 2010;
- Artigos com conteúdos condizentes com o tema;

Critérios de Exclusão:

- Artigos Duplicados;
- Artigos que ampliam ou desviam o conteúdo a temas paralelos ao da pesquisa;
- Artigos que não seja possível acessar.

3.1.4 Classificação e Extração de Dados

Após a análise dos critérios de inclusão e posterior exclusão feita através do google acadêmico, restaram 5 artigos considerados relevantes para a pesquisa.

Os artigos selecionados neste mapeamento tiveram seus atributos extraídos conforme o Quadro 2. Os atributos foram definidos com o objetivo de facilitar a visualização das respostas das questões planejadas no início do mapeamento.

Quadro 2 - Descrição dos atributos utilizados no mapeamento

Atributo	Descrição
Autor/Ano	Nome dos autores do trabalho e o ano em que foi publicado.
Finalidade do Trabalho	O(s) objetivo(s) da pesquisa desenvolvida no artigo.
Palavras-chave	Buscar no artigo se há a utilização de palavra-chave para facilitação do entendimento e leitura do mesmo.
Viés e Objetividade	Avaliação do artigo para saber se é imparcial e baseado em evidências sólidas, em vez de promover um determinado produto ou empresa.
Linguagem e Clareza	Verificação se o artigo está bem escrito, é claro e compreensível.

Relevância para o tema	Certificação de que o artigo se relaciona diretamente com o aspecto específico do tema que está sendo escrito.
------------------------	--

Fonte: O autor, 2023.

3.2 Resultados do Mapeamento Sistemático

Primeiramente em relação a arquitetura do Account Center, o artigo: *Software-as-a-service (SaaS): perspectives and challenges* (TSAI, BAI, HUANG, 2014), trás os vários aspectos da arquitetura de SaaS, com foco em personalização, *design* de banco de dados multi-inquilino e mecanismos de redundância e recuperação. Ele esboça quatro principais arquiteturas de SaaS e fornece *insights* sobre sua implementação:

1. Orientada a Banco de Dados;
2. Orientada a *Middleware*;
3. Orientada a Serviços (EasySaaS);
4. Baseada em PaaS.

No caso do Account Center a utilizada foi a primeira opção, porém se o projeto crescer e ter continuidade a terceira opção começará a ser implementada, visando a performance e segurança.

No âmbito de segurança de dados de SaaS, o artigo: *Opportunities and risks of software-as-a-service: Findings from a survey of IT executives* (BENLIAN, HESS, 2011), e o livro: *Data Security for Software as a Service* (JAYATHILAKA, et al, 2016) trazem estudos que indicam que, tanto para os que adotam quanto para os que não adotam o SaaS, as ameaças à segurança são o fator dominante que influencia a percepção de risco dos executivos de TI. Por outro lado, as vantagens de custo são o principal impulsionador que afeta a percepção de oportunidades do SaaS.

Além disso, os estudos revelam diferenças significativas nas percepções de riscos e oportunidades específicas do SaaS entre os que adotam e os que não adotam, incluindo riscos de desempenho e econômicos, melhorias de qualidade e acesso a recursos especializados.

As principais oportunidades do modelo SaaS são principalmente a sua escalabilidade e acessibilidade, por conta de sua alta volatilidade em relação ao desenvolvimento como também adaptação ao modelo de negócio, já seus riscos estão mais relacionados a sua segurança de dados e seus potenciais custos a longo prazo devido a maus planejamentos e custos inesperados devido a operação.

Precisa-se entender que a opção de uso de um sistema SaaS como também PaaS tem como suas oportunidades, como também seus riscos bem evidentes e estabelecidos, por isso no planejamento e estruturação do projeto, precisa-se conhecer muito bem o que será feito do produto final e quais serão os critérios em que esse SaaS atuará.

Em relação ao impacto empresarial de um Saas o artigo: *Determinants of software-as-a-service benefits and impact on firm performance* (LOUKISA, JANSSENB, MINTCHEV, 2018) relata que o uso do SaaS tem um impacto positivo na

inovação habilitada para TIC(Tecnologia da Informação e Comunicação) de uma empresa, especialmente quando há flexibilidade na infraestrutura de TIC da empresa, maturidade na gestão de processos e experiência anterior em terceirização de TIC (aprendizado organizacional). Já as capacidades técnicas, gerenciais e de relacionamento de TIC de uma empresa afetam positivamente o nível de sucesso da CC, o que, por sua vez, impacta positivamente no desempenho da empresa.

Por fim, o artigo: *The Impact of Software-as-a-Service on Software Ecosystems* (WALTER, KUDEL, MICHAEL, 2013) explora como a transição para o SaaS afeta as parcerias, relacionamentos e competição entre empresas no contexto dos ecossistemas de *software*. O estudo se concentra em uma única empresa líder de *software* que está fazendo a transição do modelo de *software* tradicional para o modelo SaaS.

O estudo também destaca a importância de criar nichos de mercado dentro do ecossistema, direcionar as demandas do mercado para os parceiros e criar oportunidades de negócios. Além disso, enfatiza a importância de habilitar parceiros existentes com especialização em nichos específicos e considera a criação de diferenciação nos serviços como uma estratégia importante.

Alguns exemplos desses impactos são:

- **Impacto nos Perfis dos Parceiros:** O estudo sugere uma mudança nos tipos de parceiros dentro dos ecossistemas de SaaS. Os parceiros de valor agregado (VARs), que são comuns nos ecossistemas de software tradicionais, estão sendo substituídos por Independent Software Vendors (ISVs), Original Equipment Manufacturers (OEMs) e Business Process Outsourcers (BPOs). Os parceiros de SaaS precisam de mais liquidez financeira e especialização em *marketing* e em domínios específicos.
- **Impacto nos Relacionamentos:** Os modelos de negócios de Integradores de Sistemas (SIs), que normalmente são baseados em projetos, podem não mais atender às demandas do mercado de SaaS, que exige soluções empacotadas entregues como serviço em relacionamentos de longo prazo. A confiança e relacionamentos mais profundos são essenciais para o sucesso das parcerias em ecossistemas de SaaS.
- **Impacto nos Ecossistemas:** No curto prazo, a competição entre os parceiros de SaaS pode não ser tão intensa, pois o mercado de SaaS ainda é jovem e oferece muitas oportunidades não exploradas. No entanto, no longo prazo, a competição pode aumentar, com a diferenciação ocorrendo em um nível de serviço e na marca da empresa, em vez de em recursos do produto.

3.2.1 Respostas para as Questões

Seção com as respostas para as questões definidas.

- A. **QP1:** Qual a necessidade encontrada no mercado para construção desse modelo de plataforma?

Com o cenário empresarial em constante evolução, as empresas enfrentam complexidades tecnológicas e demandas crescentes. Nesse contexto,

uma plataforma de gerenciamento de SaaS é essencial. Ela simplifica o gerenciamento de uma ampla variedade de aplicativos e ferramentas, proporcionando eficiência operacional.

A plataforma automatiza tarefas de rotina, como atualizações e *backups*, reduzindo custos operacionais e eliminando a necessidade de recursos humanos dedicados. Isso resulta em economia de tempo e recursos valiosos.

Mantendo a segurança dos sistemas e dados, a plataforma centraliza políticas de segurança, monitora ameaças em tempo real e garante atualizações de segurança. Isso protege os ativos críticos da empresa. O gerenciamento eficaz de SaaS otimiza o uso de licenças, evita redundâncias e identifica áreas de subutilização, gerando economias substanciais.

Além de *software*, a plataforma simplifica o rastreamento de licenças e outros ativos, como *hardware*. Isso melhora o controle e a eficiência na administração dos recursos da empresa.

A plataforma fornece informações detalhadas sobre o uso de software serviços, permitindo decisões informadas e ágeis. Isso ajuda as empresas a acompanhar as mudanças no ambiente de negócios. Mantendo sistemas confiáveis, a plataforma reduz problemas técnicos para clientes e usuários internos, melhorando a satisfação do cliente e a reputação da empresa.

B. QP2: Quais são os principais benefícios financeiros que uma empresa pode obter ao adotar uma plataforma de gerenciamento de software serviços, em termos de economia de custos e otimização de recursos?

A adoção de uma plataforma de gerenciamento de SaaS proporciona vantagens financeiras significativas para as empresas:

- **Redução de Custos Operacionais:** A automação de tarefas de rotina, como atualizações e *backups*, reduz custos operacionais, eliminando a necessidade de recursos humanos dedicados (IBMSAAS, 2020).
- **Otimização de Licenças:** Rastrear licenças de software evita compras desnecessárias e uso inadequado de ativos licenciados, economizando em taxas de licença (RADIX, 2023).
- **Gestão Eficiente de Ativos:** Além do *software*, a plataforma otimiza o uso de *hardware*, identificando ativos subutilizados ou necessitando de substituição (RADIX, 2023).
- **Conformidade:** Centralizar o gerenciamento de licenças evita multas por não conformidade com políticas de licenciamento (RADIX, 2023).
- **Eficiência de TI:** Recursos de TI são alocados com mais eficiência, evitando gastos em projetos não estratégicos (RADIX, 2023).
- **Agilidade:** Atualizar ou remover SaaS de forma ágil reduz custos relacionados à implementação demorada (RADIX, 2023).
- **Decisões Informadas:** Dados detalhados auxiliam na tomada de decisões bem fundamentadas sobre investimentos e reduções de gastos (RADIX, 2023).

- **Satisfação do Cliente:** Uma operação mais eficiente contribui para a satisfação do cliente, promovendo a retenção e a receita adicional (RADIX, 2023).

Resumindo uma plataforma de gerenciamento de SaaS aprimora a eficiência financeira das empresas, tornando-as mais competitivas e ágeis em um ambiente empresarial dinâmico (IBMSAAS, 2020).

C. QP3: Qual a vantagem de centralizar seus SaaS em uma plataforma de gerenciamento?

A centralização de sistemas serviços em uma plataforma de gerenciamento é vantajosa, pois proporciona eficiência, controle e otimização das operações empresariais. Essa abordagem simplifica a gestão, economiza tempo e recursos ao automatizar tarefas rotineiras, reduz custos ao otimizar licenças de *software* e aumenta a segurança ao aplicar políticas consistentes.

Além disso, ela facilita a tomada de decisões embasadas em dados detalhados sobre o uso de sistemas e serviços, permitindo que a empresa seja ágil em resposta às mudanças nas necessidades do negócio.

Também contribui para a gestão eficiente de ativos, incluindo *hardware*, melhorando o controle e a administração de recursos. Essa abordagem confiável resulta em menos problemas técnicos para clientes e usuários internos, melhorando a satisfação do cliente e a reputação da empresa.

Resumindo a centralização beneficia a empresa ao simplificar processos, economizar recursos, fortalecer a segurança, embasar decisões e aumentar a agilidade, tornando-a mais competitiva em um ambiente empresarial dinâmico.

4. Account Center: Plataforma Gerenciadora de Softwares Serviços.

Neste capítulo serão abordados quais foram as metodologias de desenvolvimento do Account Center, explicações e exemplos sobre seu funcionamento.

4.1 Metodologia de Desenvolvimento

O tipo de metodologia adotado para desenvolvimento foi a metodologia ágil, por conta de sua flexibilidade, foco na colaboração e capacidade de adaptação a mudanças. Neste caso, foi utilizada uma fusão de dois tipos de metodologia ágil, que são a *Scrum* que se baseia no uso de sprints, que em geral consistem em sessões de trabalho de duas semanas com entregáveis específicos a serem apresentados ao final e que também possui reuniões diárias de cerca de 15 minutos para interação entre os participantes em relação às demandas da sprint (SCRUM, 2023) e o *Kanban* que é aplicado por meio de quadros, uma forma de gestão visual de projetos que permite às equipes visualizar melhor a sua carga e fluxo de trabalho. Em um quadro Kanban, o trabalho é exibido como um quadro de projetos organizado por colunas. Tradicionalmente, cada coluna representa um estágio do trabalho (KANBAN, 2023). E com a aplicação dessa metodologia por meio da ferramenta Trello (TRELLO, 2023).

Foi optado por mesclar esses dois tipos de metodologia ágil por conta da flexibilidade, redução de sobrecarga e maior transparência, por se tratar de um estilo extremamente adaptável para o desenvolvimento. Sendo assim, todas as tarefas e processos foram separados em sprints e listados em cartões com o status em que se encontram, a fim de mapear e acompanhar o progresso de desenvolvimento.

A criação do Account Center envolveu uma abordagem organizada e dividida em três partes, cada uma desempenhando um papel fundamental na construção bem-sucedida do sistema. A seguir, destaca-se essas etapas:

1. Definição das Regras de Negócio:

A primeira fase deste projeto teve como foco a definição das regras de negócio. Nesse estágio inicial, foi realizada uma construção lógica abrangente, contemplando desde as regras que governam a estrutura geral do projeto até as diretrizes para a gestão de módulos e produtos. Um aspecto crucial foi o estabelecimento do fluxo de Assinaturas e seus Produtos, determinando como seriam mantidos e utilizados ao longo do sistema.

2. Estruturação do Projeto:

Para entender como a estruturação do projeto ocorreu primeiro é preciso entender alguns conceitos da área de desenvolvimento que corroboram para uma estruturação sólida, começando pela definição de *backend* e *frontend*, que respectivamente significam, primeiro a parte onde será escrito e mantido todas as regras de negócio e lógicas do sistema, como também a ponte de comunicação direta das rotas com o banco de dados que será consumido pelo *frontend*, já o *frontend* é a parte que se responsabiliza pelos aspectos visuais e interativos que o cliente acessa e consome.

Outras definições que podem ser importantes ressaltar são:

- **Arquitetura Monolítica:** Consiste normalmente em uma interface de usuário do lado do cliente, um banco de dados e uma aplicação do lado do servidor. Além de que todo esse fluxo fica armazenado em uma única base de código;
- **Scrumban:** É uma abordagem alinhada ao *Agile* para entrega de produtos, que é um híbrido de *Scrum* e *Kanban*. O *Scrumban* foi originalmente projetado como uma forma de transição do *Scrum* para o *Kanban*.

Foram tomadas decisões importantes sobre como iniciar o desenvolvimento. Isso incluiu a escolha da arquitetura, optando por uma abordagem monolítica devido à facilidade de implementação e agilidade no desenvolvimento. Quanto às tecnologias utilizadas, optou-se por usar PHP (Laravel) para o *backend*, devido à sua simplicidade e documentação abrangente. Para o *frontend*, JavaScript (Vue.js) e Quasar foram selecionados devido à sua versatilidade e baixa curva de aprendizado. Quanto à metodologia

de desenvolvimento, foi adotado o *Scrumban*, uma abordagem híbrida do *Scrum* e *Kanban*, visando alta agilidade e rapidez.

3. Desenvolvimento:

A fase de desenvolvimento foi a etapa mais extensa, na qual os códigos foram criados para dar vida ao sistema e sustentar suas funcionalidades. Alguns dos principais pontos de desenvolvimento incluíram:

- Instalação de bibliotecas e configuração do ambiente (.env);
- Criação da estrutura de pastas, organizada por módulos e visualizações tanto no *backend* quanto no *frontend*;
- Escolha e configuração da conexão com o banco de dados;
- Implementação da comunicação entre o *backend* e o *frontend* por meio do AXIOS;
- Desenvolvimento dos módulos e suas respectivas tabelas;
- Implementação da autenticação por *token*;
- Desenvolvimento das interfaces para os usuários;
- Integração com um *gateway* de pagamento.

Essas etapas detalhadas garantiram que o Account Center fosse construído de forma sólida e eficaz, com base em regras de negócios bem definidas e uma estrutura tecnológica adequada. A escolha de uma metodologia híbrida, que combina o melhor do *Scrum* e do *Kanban*, contribuiu para a agilidade e flexibilidade do projeto. O resultado é um sistema robusto e funcional que atende às necessidades da empresa de maneira eficiente.

4.2 Funcionamento da Plataforma

No mundo empresarial atual, a tecnologia desempenha um papel central em quase todos os aspectos das operações. Empresas de todos os tamanhos e setores dependem de uma variedade de SaaS para operar de forma eficiente e competitiva. No entanto, gerenciar esse ecossistema tecnológico pode ser uma tarefa desafiadora.

Tendo como base o que se foi exposto até então se chegar ao motivo da criação do *Account Center* que se trata em gerenciar ecossistemas de SaaS, porém não para empresas utilizadoras destes SaaS, mas sim para as que os disponibilizam, por isso do controle via assinaturas, para que as empresas provedoras, tenham esse controle sobre o acesso de seus SaaS, funciona como uma central de controle para todos os ativos tecnológicos de uma organização. Seu objetivo é oferecer uma visão unificada e abrangente de todo o ecossistema digital. Abaixo está listado algumas das características da plataforma:

1. **Centralização de Informações:** A plataforma serve como um repositório centralizado onde todas as informações relevantes são armazenadas. Isso inclui detalhes sobre os SaaS, controle de assinaturas e informações de contato de fornecedores;
2. **Monitoramento de Atividades:** Rastreia continuamente as atividades relacionadas às assinaturas. Isso inclui informações sobre quem está usando quais recursos, quais atualizações foram aplicadas e a validade das mesmas;

3. **Gestão de Assinaturas:** A plataforma ajuda a empresa a acompanhar todas as assinaturas dos SaaS. Ela pode notificar sobre datas de renovação, garantir que todas as licenças estejam em conformidade e evitar multas por não conformidade;
4. **Sistema de Permissões:** Esse recurso garante que mesmo pessoas da própria empresa provedora não acessem todas as informações que a plataforma contém, fazendo com que se tenha um melhor controle de quem acessa quais dados.

A seguir as principais telas do sistema e a descrição de suas funcionalidades:

VERSÃO ADMIN:

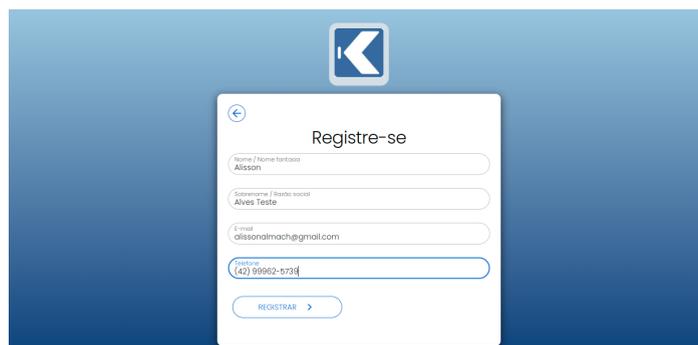
Login/Registrar: Telas iniciais, primeiro contato que o usuário terá (Figura 1 e 2).

Figura 1 - Tela de Login

A imagem mostra a tela de login de um sistema. No topo, há um ícone de uma seta para a esquerda dentro de um quadrado. Abaixo dele, o texto "Bem-vindo" é exibido. Seguem dois campos de entrada: "E-mail" com o valor "email@gmail.com" e "Senha" com caracteres ocultos por pontos e um ícone de olho para alternar a visibilidade. Abaixo dos campos, há um link "Esqueci minha senha". Na base da caixa de formulário, existem dois botões: "REGISTRAR-SE" e "LOGAR".

Fonte: O autor, 2023.

Figura 2 - Tela de Registro

A imagem mostra a tela de registro de um sistema. No topo, há um ícone de uma seta para a esquerda dentro de um quadrado. Abaixo dele, o texto "Registre-se" é exibido. Seguem quatro campos de entrada: "Nome / Nome fantasia" com o valor "Allison", "Sobrenome / Pseudo social" com o valor "Alves Teste", "E-mail" com o valor "alisonalmach@gmail.com" e "Telefone" com o valor "(42) 99962-5730". Na base da caixa de formulário, há um botão "REGISTRAR" com uma seta para a direita.

Fonte: O autor, 2023.

Produtos: Após logar, caso seja um usuário novo, o primeiro item do menu após a *dashboard* é a tela de produtos, que é onde será cadastrado os SaaS que serão assinados posteriormente (Figuras 3 e 4).

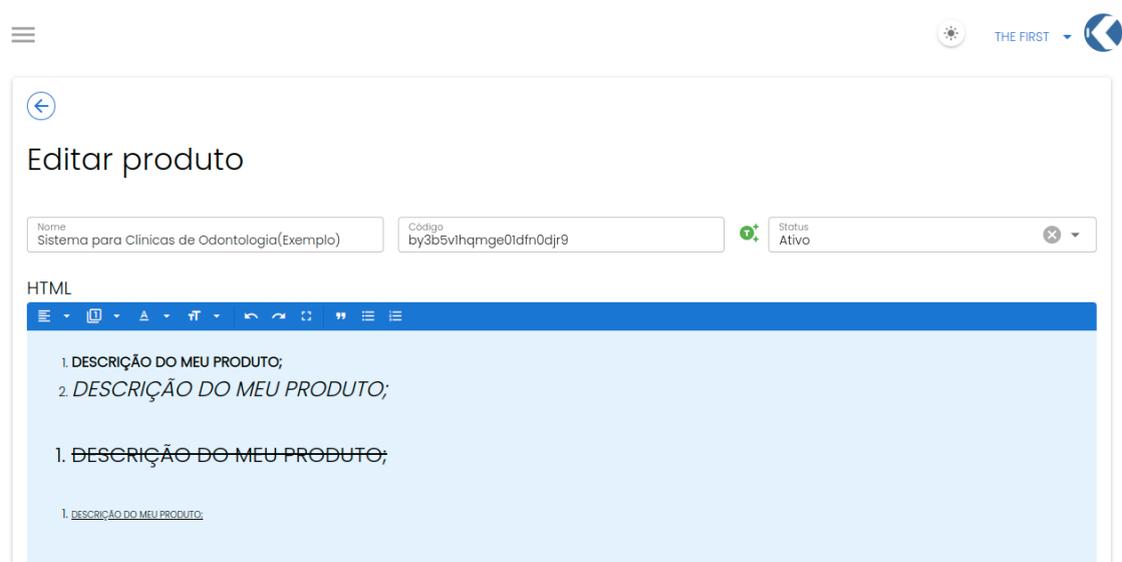
Figura 3 - Tela de listagem de Produtos



Fonte: O autor, 2023.

A listagem contém todos os produtos do usuário logado e também algumas ações como copiar o *link* (em verde) para registro de usuário (*link* que é para a página de registrar para o usuário se registrar e já ver somente os produtos desse usuário), *link* de acesso ao produto (segundo botão) e dois botões que fazem as ações de editar e excluir respectivamente.

Figura 4 - Tela de criação/edição de Produtos



Fonte: O autor, 2023.

Na Figura 4 é a tela de criar/editar produto, onde é preenchido as informações que vão dar origem ao produto, sendo elas o nome, código (gerado automaticamente), vindiID (id da conta para pagamento) e abaixo um editor de texto que funciona com base em HTML para preencher uma descrição mais personalizada para melhor visualização na hora da leitura para descrição do produto.

Planos: Este módulo é responsável por manter os planos dos produtos criados e é responsável pela organização desses planos na hora do usuário visualizar e assinar algum deles (Figuras 5 e 6).

Figura 5 - Tela de listagem de Planos

Nome	Produto	Preferencial	Visível	Padrão	Ordem	Ações
PLANO TESTE 1	Sistema para Clinicas de Odontologia(Exemplo)	✓ Sim	✓ Sim	✓ Sim	0	[Edit] [Delete]
PLANO TESTE 2	Sistema para Clinicas de Odontologia(Exemplo)	✗ Não	✓ Sim	✗ Não	1	[Edit] [Delete]

Fonte: O autor, 2023.

Na Figura 5 está a tela que lista todos os planos, permite trocar a ordenação, filtrar por produto e também nos permite editar ou excluir os mesmos.

Figura 6 - Tela de criação/edição de Planos.

Editar plano

Preferencial: Visível: Padrão:

Nome: PLANO TESTE 1 Produto: Sistema para Clinicas de Odontolog

JSON: [value:0]

HTML:

- DESCRIÇÃO DO PLANO;

SALVAR

Fonte: O autor, 2023.

Essa tela (Figura 6) é a responsável por realizar a criar/editar um plano, contendo alguns alternadores para deixar o plano com algumas características destaques dentre os demais, além disso também é possível personalizar a descrição do plano com o mesmo editor em HTML que é possível observar em Produtos.

Assinaturas: É o módulo responsável por gerenciar todas as assinaturas que ocorrem nos produtos e seus planos (Figura 7).

Figura 7 - Tela de listagem de Inscrições

ID	Usuário	Produto	Plano	Finalizado em	Ações
1	Alisson Teste	Sistema para Clinicos de Odontologia(Exemplo)	PLANO TESTE 1	N/A	

Fonte: O autor, 2023.

A listagem de assinaturas possibilitam ao usuário visualizar e gerenciar as assinaturas que ocorrem aos produtos de seu sistema, é uma tela relativamente simples, pois toda a regra das assinaturas é gerenciada pela API e seu cancelamento e manutenção é totalmente automática.

Usuários: Neste módulo é onde ficam armazenados os usuários do sistema, tanto administradores quanto clientes, é uma tela focada no gerenciamento de usuários, sendo assim possui algumas funcionalidades como bloquear usuário (Figuras 8 e 9).

Figura 8 - Tela de listagem de Usuários

Nome	E-mail	Tipo	Status	Ações
Alisson Teste	alissonalmachado@gmail.com	Membro	Ativo	

Fonte: O autor, 2023.

A Figura 8 representa a listagem de usuários em que podemos pesquisar usuários e realizar algumas ações com essa entidade, como bloquear, editar ou excluir, respectivamente.

Figura 9 - Tela de criação/edição de Usuários

Fonte: O autor

Na Figura 9 pode-se criar/editar usuários e também trocar suas senhas ou editar seus dados pessoais.

VERSÃO CLIENTE:

Dashboard: Este módulo apresenta os produtos para os usuários clientes assinarem ou acessem seus produtos já assinados, lembrando que todo cliente precisa estar atrelado a um administrador, ou seja, o cadastro é feito a partir de um administrador justamente para atrelar uma entidade a outra (Figuras 10, 11 e 12).

Figura 10 - Tela de listagem de Produtos, visão do cliente

Fonte: O autor, 2023.

Esta tela apresentada na figura 10 possui a listagem dos produtos, onde o cliente consegue realizar as ações de assinar planos, acessar o sistema, enviar sugestões ou cancelar o produto assinado.

Figura 11 - Tela de Planos, visão do cliente

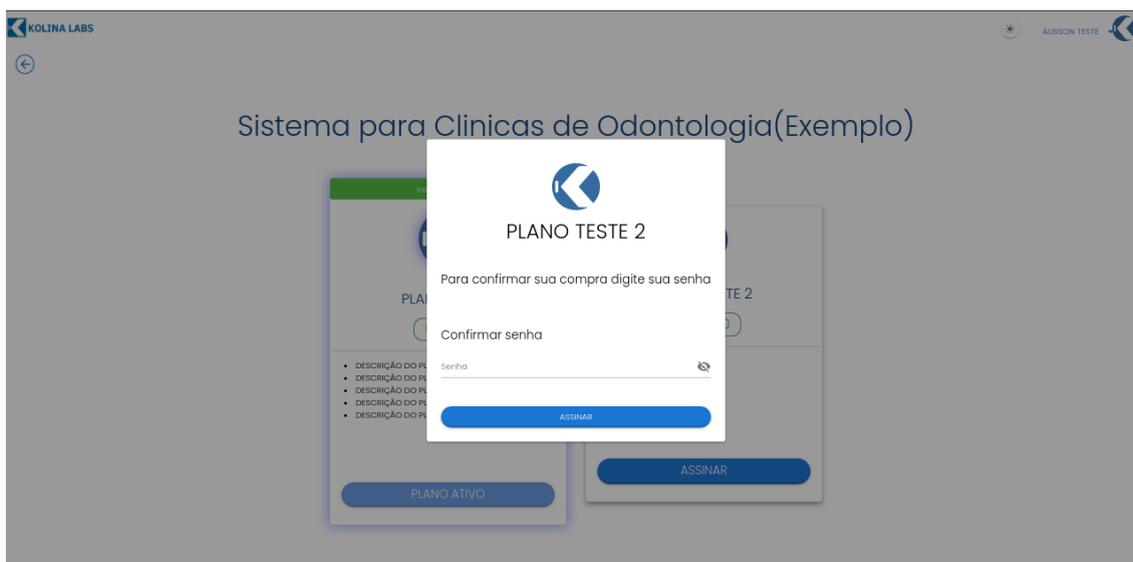
Sistema para Clinicas de Odontologia(Exemplo)



Fonte: O autor, 2023.

Na Figura 11 são mostrados os planos de um produto, e é onde o usuário irá escolher qual plano vai assinar desse determinado produto.

Figura 12 - Tela de Planos, modal de assinar plano, visão do cliente



Fonte: O autor, 2023.

Tela final do processo de assinatura (Figura 12), onde após o usuário preencher seus dados de pagamento, deverá preencher sua senha para terminar de assinar o produto, após isso, na listagem, já será liberado para o usuário acessar o SaaS que foi assinado (produto).

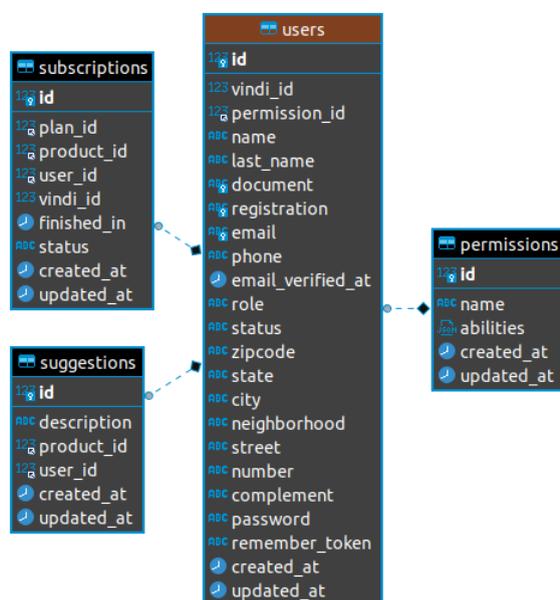
Essas são algumas das principais telas que o Account Center possui, ainda há mais algumas telas que pode-se mencionar, como: permissões, *dashboard* do administrador, edição de perfil do cliente, sugestões, edição de assinatura e minhas assinaturas (versão do cliente).

5. Resultados

Com base nos mapeamentos e nas escolhas de estrutura e arquitetura para o desenvolvimento chega-se em um *roadmap* (roteiro) que foi dividido em *sprints* que podem ser acessadas no Trello(TRELLO, 2023), onde se tem a separação das partes que no final se juntariam e formariam o todo, sendo elas:

Banco de Dados: Criação e estruturação do BD para suportar as funcionalidades do Account Center, nessa etapa, iniciou-se a criação e formato do BD feito em MySQL usando as *migrations* do Laravel para auxiliar no desenvolvimento, nas Figuras 13 e 14 estão representados os diagramas que exemplificam como esse BD funciona:

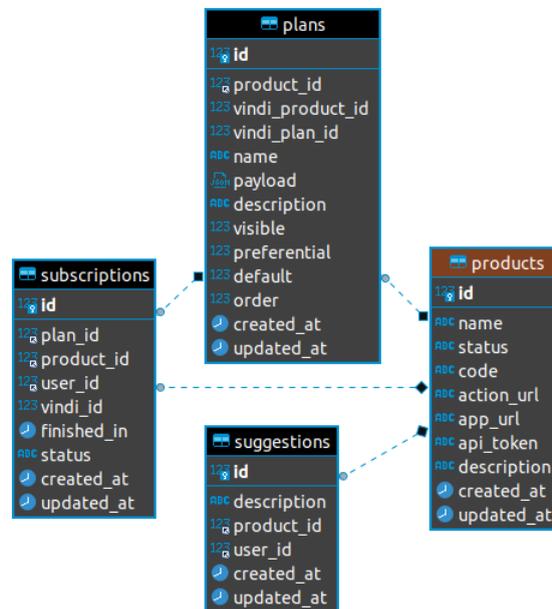
Figura 13 - Fluxo do usuário



Fonte: O autor, 2023.

A Figura 13 se trata de um diagrama em que é apresentada as tabelas e seus respectivos relacionamentos, podemos perceber que a tabela *users* é a entidade principal onde as demais giram em torno dela.

Figura 14 - Fluxo do produto



Fonte: O autor, 2023.

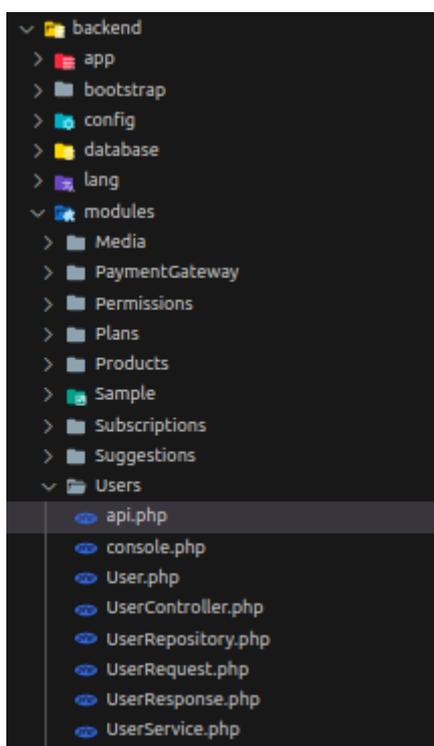
Assim como o diagrama de fluxo de usuário (Figura 13), o diagrama do produto (Figura 14) é também uma das entidades principais e tem várias outras entidades dependentes dele, como inscrições e planos.

API: Após a criação do banco de dados, foi dado início na estrutura da API, sendo ela escrita de forma modular, ou seja, cada entidade tem seu módulo separado contendo a seguinte estrutura de código:

- **api:** onde é registrado as rotas para serem consumidas pela SPA (*frontend*);
- **model:** responsável pela leitura e escrita de dados, e também de suas validações.
- **controller:** responsável por receber todas as requisições do usuário.
- **service:** responsável por manter e gerenciar os métodos e funções que serão chamadas pela controller para serem consumidas pelo usuário.

Com base nessa estrutura, conseguimos separar cada função em seu devido lugar, para melhor qualidade de código e facilidade de manutenção, na Figura 15 o resultado da estruturação:

Figura 15 - Organização de pastas



Fonte: O autor, 2023.

Na Figura 15 é possível observar o resultado da modularização das entidades, mostrando os arquivos que fazem parte da estrutura.

Figura 16 - Arquivo exemplo de rotas: *api.php*.

```
1 <?php You, há 16 meses • Second commit
2
3 use Illuminate\Support\Facades\Route;
4 use Media\MediaController;
5
6 Route::group([
7     'middleware' => ['auth:sanctum', 'user_checker']
8 ], function () {
9     Route::apiResource('media', MediaController::class);
10 });
11
```

Fonte: O autor, 2023.

Na Figura 16 apresenta um exemplo de um arquivo *api.php* onde são mapeadas as rotas, neste caso utilizando o *apiResource* que disponibiliza todos os tipos de requisições (*GET, POST, PATCH, PUT, DELETE*).

Figura 17 - Arquivo exemplo de models: Media.php.

```
1 <?php
2
3 namespace Media;
4
5 use Products\Product;| Kolina, há 16 meses • upload logo prod
6 use Users\User;
7 use Illuminate\Database\Eloquent\Model;
8 class Media extends Model
9 {
10 |
11 |     const MEDIA_TYPE_USER_PROFILE = 'user_profile';
12 |     const MEDIA_TYPE_PRODUCT_LOGO = 'product_logo';
13 |     const SUBJECT_TYPES_MAPPING = [
14 |         self::MEDIA_TYPE_USER_PROFILE => User::class,
15 |         self::MEDIA_TYPE_PRODUCT_LOGO => Product::class
16 |     ];
17 |     const REPLACEABLE_TYPES = [
18 |         self::MEDIA_TYPE_USER_PROFILE,
19 |         self::MEDIA_TYPE_PRODUCT_LOGO
20 |     ];
21 |
22 |     protected $table = 'media';
23 |
24 |     /** @var string[] */
25 |     protected $fillable = [
26 |         'filename',
27 |         'description',
28 |         'path',
29 |         'media_type',
30 |         'mime_type',
31 |         'extension',
32 |         'subject_id',
33 |         'subject_type',
34 |     ];
35 |
36 |     public function subject()
37 |     {
38 |         return $this->morphTo();
39 |     }
40 | }
41
```

Fonte: O autor, 2023

Na Figura 17 tem-se um arquivo exemplo de uma model de uma entidade, neste caso a entidade medias, que é responsável por manter as imagens que são salvas no sistema.

Figura 18 - Arquivo exemplo de uma controller: *MediaController.php*

```
1 <?php
2
3 namespace Media;
4
5 use App\Http\Controllers\Controller;
6 class MediaController extends Controller
7 {
8     use MediaResponse;
9     private MediaService $mediaService;
10
11     public function __construct(MediaService $mediaService)
12     {
13         $this->mediaService = $mediaService;
14     }
15
16     public function index(MediaRequest $request)
17     {
18         $result = $this->mediaService->index($request->validated());
19         return $this->response($result['response'], $result['status']);
20     }
21
22     public function store(MediaRequest $request)
23     {
24         $result = $this->mediaService->store($request->validated());
25         return $this->response($result['response'], $result['status']);
26     }
27
28     public function show($mediaId)
29     {
30         $result = $this->mediaService->show($mediaId);
31         return $this->response($result['response'], $result['status']);
32     }
33
34     public function update(MediaRequest $request, $mediaId)
35     {
36         $result = $this->mediaService->update($mediaId, $request->validated());
37         return $this->response($result['response'], $result['status']);
38     }
39
40     public function destroy($mediaId)
41     {
42         $result = $this->mediaService->destroy($mediaId);
43         return $this->response($result['response'], $result['status']);
44     }
45 }
```

Fonte: O autor, 2023.

Neste arquivo (Figura 18) é possível observar que é mantido todas as chamadas para os métodos que vão executar certas ações na entidade medias, um exemplo é o store, que chama o *service* que possui uma função que realiza o salvamento de uma media na base de dados.

SPA: Nesta etapa foi construído a parte visual do sistema, utilizando uma estrutura de páginas componentizadas para melhor visualização de código e manutenção, também por conta de que a componentização possibilita a reutilização do mesmo código em situações diferentes, o que auxilia muito na hora do desenvolvimento em geral.

Essa parte do desenvolvimento foi a mais rápida por conta também da estruturação do projeto, pois, por se tratar de uma SPA, seu foco é justamente na rápida entrega e na baixa complexidade, pois as SPAs trabalham com apenas uma única

página. Ao invés de recarregar toda a página ou redirecionar o usuário para uma página nova, apenas o conteúdo principal é atualizado de forma assíncrona, mantendo toda a estrutura da página estática.

Priorizando também otimizar a performance da aplicação, reduzindo o conteúdo a ser carregado, tendo como foco na experiência do usuário, que lida com uma interface mais rápida.

6. Considerações finais

Após todos os detalhes, conclui-se que a criação do *Account Center* adotou metodologias e infra estruturas ágeis de modo geral, para priorizar a rápida evolução e desempenho para que o produto final fosse o mais completo possível, atendendo seu objetivo e ainda sim, conseguindo transmitir uma boa experiência para os usuários, mesmo esse não sendo o foco nessa etapa inicial do projeto.

Em relação a seu desenvolvimento, a nível de código, percebemos que já de início o projeto foi executado de maneira pensada prevendo crescimentos e criação de novas funcionalidades por adotar metodologias de desenvolvimento que já esperam um crescimento contínuo e duradouro.

Como produto, o *Account Center* conseguiu criar uma base sólida para futuras versões e melhorias, por ter uma ideia bastante abrangente, possibilitando evoluir em muitos aspectos, um deles seria a expansão das metodologias de assinaturas, possibilitando um controle mais detalhado em relação a tempo e valores, assim como a criação de dashboards com relatórios e gráficos informativos e também uma série funcionalidades que agregam no sistema como plataforma, como possibilidade de integração com outros gateways de pagamento, chat de suporte, notificações via pusher, opções de escolha de tema e personalização de visuais, etc...

Por fim, vale ressaltar que a construção de um *software* desse nível abriu um leque muito grande de novas possibilidades e ideias para futuros projetos pessoais, assim como também mais materiais para possíveis projetos nessa linha de pensamento.

Referências

ACCESSTAGE. Disponível em

<<https://blog.accesstage.com.br/por-que-adotar-a-gestao-centralizada-de-processos-fina-ncieiros>> Acessado em 28 de nov. de 2022.

ASAAS, Disponível em <<https://blog.asaas.com/afinal-o-que-e-gestao-de-assinaturas/>> Acessado em 30 de out de 2023.

ATLASSIAN, Disponível em

<<https://www.atlassian.com/br/agile/kanban/kanban-vs-scrum>> Acessado em 30 de out de 2023.

B2B STACK. Disponível em <<https://blog.b2bstack.com.br/saas/>> Acesso em: 23 de ago. de 2023.

BENLIAN, Alexander, HESS, Thomas, Opportunities and risks of software-as-a-service: Findings from a survey of IT executives, Institute for Information Systems and New Media, University of Munich, Ludwigstr. 28, 80539 Munich, Germany, 4 ago 2011.

DEVLARAVEL, Disponível em

<<https://www.skoob.com.br/livro/pdf/desenvolvendo-com-laravel/livro:675679/edicao:677777>> Acessado em 30 de out de 2023.

GETASTRA, Disponível em

<<https://www.getastra.com/blog/cms/saas-security-guide/>> Acessado em 30 de out de 2023.

IBMSAAS, Disponível em

<<https://www.ibm.com/blog/top-5-advantages-of-software-as-a-service/>> Acessado em 30 de out de 2023.

INTERGATE. Disponível em

<<https://www.intergate.net.br/blog/conheca-3-metodos-de-autenticacao-de-apis/>> Acessado em 23 de ago. de 2023.

JAYATHILAKA, Hiranya, et al, Data Security for Software as a Service, Mehdi Khosrow-Pour, DBA Contemporary Research in Information Science and Technology, Book Series, 2016.

KANBAN, Disponível em <<https://asana.com/pt/resources/what-is-kanban>>

KITCHENHAM, Barbara; CHARTERS, Stuart. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

LARAVEL, Disponível em <<https://laravel.com/docs/10.x>> Acessado em 30 de out de 2023.

LOUKISA, Euripidis, JANSSENB, Marijn, MINTCHEV, Ianislav Determinants of software-as-a-service benefits and impact on firm performance, University of the Aegean, Karlovassi 83200, Greece, 29 dec 2018.

MULTIEDRO. Disponível em

<<https://blog.multiedro.com.br/a-importancia-da-gestao-da-informacao-no-ambiente-corporativo/>> Acessado em 23 de ago. de 2023.

MYSQL, Disponível em <<https://www.mysql.com/>> Acessado em 30 de out de 2023.

PETERSEN Kai, FELDT Robert, MUTJADA Shahid, MATTSON Michael, Systematic Mapping Studies in Software Engineering, School of Engineering, Blekinge Institute of Technology, Box 520, 2008.

QUASAR, Disponível em <<https://quasar.dev/introduction-to-quasar>> Acessado em 30 de out de 2023.

RADIX, Disponível em

<<https://radixweb.com/blog/benefits-of-saas>> Acessado em 30 de out de 2023.

SALESFORCE. Disponível em <<https://www.salesforce.com/br/saas/>> Acessado em 23 de ago. de 2023.

SCRUM, Disponível em <<https://asana.com/pt/resources/what-is-scrum>>

SUPERLOGICA. Disponível em

<https://blog.superlogica.com/assinaturas/saas-o-que-e-e-quais-suas-vantagens/?gclid=Cj0KCQjw3JanBhCPARIsAJpXTx6Y6bOQXFLuYrVTO-5E2OFVvByILGenXwHrZeQQMRWgSLk3ETV8Pr8aAtt5EALw_wcB> Acesso em: 10 de ago. de 2021.

TRELLO, Disponível em

<<https://trello.com/invite/b/p6mEzjmj9/ATTI4efbe1be55eb79cb986b6be5d83aa82b7A5DA485/accountcenter-sprints.>>

TSAI WeiTek, BAI XiaoYing, HUANG Yu1, Software-as-a-service(SaaS): perspectives and challenges, Department of Computer Science and Technology, Tsinghua National Lab of Information Science and Technology, Tsinghua University, Beijing 100084, China, 11 mar 2014.

VUEJS, Disponível em <<https://vuejs.org/guide/introduction.html#what-is-vue>> Acessado em 30 de out de 2023.

VUEJS3COOKBOOK, Disponível em

<<https://github.com/PacktPublishing/Vue.js-3.0-Cookbook>> Acessado em 30 de out de 2023.

WALTER, Sebastian, KUDEL, Thomas, MICHAEL Karl, The Impact of Software-as-a-Service on Software Ecosystems, University of Mannheim, Walldorf, Germany, 2013.