

Daniela Filipa Pessoa Cortesão

KamaeLei Cloud — Software as a Service

Relatório de Dissertação/Estágio
Mestrado em Engenharia Informática
orientada por Nuno Lourenço (DEI) e Ricardo Teixeira (KAMAE)
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Julho de 2018



UNIVERSIDADE DE COIMBRA



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Daniela Filipa Pessoa Cortesão

KamaeLei Cloud — Software as a Service

Relatório de Dissertação/Estágio
Mestrado em Engenharia Informática
orientada por Nuno Lourenço (DEI) e Ricardo Teixeira (KAMAE)
e apresentada ao Departamento de Engenharia Informática
da Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Júri Arguente: Joel Perdiz Arrais
Júri Vogal: Pedro Furtado

Julho de 2018

Resumo

No ramo da advocacia, uma gestão rigorosa e focada é fundamental, pois são inúmeras as tarefas a realizar, e há uma grande quantidade de informações a reter, tornando-se cada vez mais difícil manter toda a informação em papel.

A KAMAE tem como principal objetivo, facilitar o dia-a-dia dos profissionais deste ramo, oferecendo uma vasta gama de produtos de software que visam ajudar em toda a gestão dentro de escritórios de advogados. Com vista a alcançar a internacionalização ao nível mundial, esta empresa encontra-se a desenvolver um serviço cloud, o S4L (Solutions For Lawyers), que atualmente se encontra já em fase beta, estando a ser utilizada em ambientes reais. No entanto, há ainda um longo caminho a percorrer, havendo a necessidade de implementar componentes vitais ao sucesso da solução oferecida.

Foi com o objetivo de completar o S4L que surgiu o estágio apresentado neste documento. Sendo que, foram atribuídas à estagiária, as tarefas de planejar e implementar o backoffice de cliente e ainda completar o backoffice KAMAE, este último já numa fase avançada de implementação, passando também pelo levantamento dos requisitos em falta.

O presente documento descreve e demonstra o trabalho realizado, ao longo do estágio, descrevendo todo o planeamento, realizado numa primeira fase do estágio, e ainda demonstrando e justificando a fase de implementação e todas as decisões que foram tomadas durante esta fase. Este estágio decorreu num ambiente empresarial, e com recurso a uma metodologia de gestão ágil, o SCRUM.

É espectável, que após leitura deste documento, o leitor entenda todo o trabalho que foi realizado pela estagiária e quais as suas motivações, assim como todo o percurso percorrido e desafios encontrados até ao término deste estágio.

Palavras-Chave

“Advocacia”, “KAMAE”, “gestão”, “serviço cloud”, “S4L (Solutions For Lawyers)”, “backoffice de cliente”, “backoffice KAMAE”, “SCRUM”

Abstract

In the law business, tight and focused management is fundamental, because there are many tasks to perform and a lot of information to retain, making it difficult to keep all the data on paper.

KAMAE’s main goal is to facilitate the day-by-day of the professionals of this field, offering a wide range of software solutions that aim to help the management of the law firms to make better decisions.

In order to achieve globalization, this company is developing a cloud service, the S4L (Solutions For Lawyers), which is currently in beta, being used in real environments. However, there is still a long way to go, with the need to implement critical components to the success of the solution.

The internship came to live aligned with the need of finishing the S4L solution. Given that, the trainee was assigned the tasks of planning and implementing the client backoffice and also help with the implementation of the KAMAE backoffice, already in an advanced phase of implementations.

This document describes the work done during the internship, showing all the planning produced in a first stage, and still demonstrating and justifying the implementation and all decisions taken during the second stage. This internship was conducted in a business environment, and using an agile management methodology.

It is also expected that after reading this document, the reader will be able to understand all the work that was done by the trainee and what motivations, as well as the entire journey and challenges encountered until the end of this internship.

Keywords

“law business”, “KAMAE”, “management”, “cloud service”, “S4L (Solutions For Lawyers)”, “client backoffice”, “KAMAE backoffice”, “SCRUM”

Índice

Capítulo 1 Introdução	1
1.1. Enquadramento.....	1
1.2. Objetivos	2
1.3. Trabalho Realizado	3
1.4. Estrutura do documento.....	3
Capítulo 2 Estado da Arte	5
2.1. S4L.....	5
2.2. Backoffice KAMAE	6
2.3. Outras soluções Existentes.....	7
.....	7
2.4. Comparação das soluções.....	8
Capítulo 3 Abordagem.....	11
3.1. Metodologias.....	11
3.2. Tecnologias e Ferramentas	13
3.3. Plano de trabalhos do 2º semestre.....	15
Capítulo 4 Análise de Requisitos	17
4.1. Especificação de Requisitos.....	17
4.2. Requisitos Funcionais.....	17
4.3. Diagramas de Casos de Uso	23
4.4. Requisitos Não Funcionais.....	25
Capítulo 5 Arquitetura.....	27
5.1. Análise da Arquitetura.....	27
Capítulo 6 Implementação	35
6.1. Aspetos de Implementação	35
6.2. Base de Dados	37
6.3. Backoffice Cliente	39
6.4. Backoffice KAMAE	48
6.5. Web API.....	50
6.6. Gateways de Pagamento	51
Capítulo 7 Plano de Riscos.....	55

7.1.	Caracterização do risco	55
7.2.	Levantamento de Riscos	56
7.3.	Matriz de Exposição de Riscos	57
Capítulo 8 Protótipos		59
Capítulo 9 Conclusão e Trabalho Futuro.....		63
9.1.	Reflexão	63
9.2.	Trabalho futuro	64
Referências		65
Apêndice A – Estado da Arte		69
Apêndice B – Plano de trabalhos do 1º semestre		75
Apêndice C – Especificação de requisitos		77
Apêndice D – Arquitetura		99
Apêndice E – Protótipos		101
Apêndice F – Análise aos Gateways de pagamento		107

Lista de Figuras

Figura 2.1- Menu do Backoffice KAMAE	6
Figura 2.2 - Menu Samcart.....	7
Figura 2.3 - Menu Hotmart	8
Figura 3.1 [12] - Ciclo do modelo SCRUM.....	11
Figura 3.2 - Organização de tarefas do Asana	13
Tabela 4.1 - Classificação de requisitos	17
Figura 4.2 - Diagrama de Casos de Uso de downgrade/upgrade de subscrição	24
Figura 4.3 - Diagrama de casos de uso de consultar dados de subscrição.....	24
Figura 4.4 - Diagrama de casos de uso de suspender/retomar subscrição.....	24
Figura 4.5 - Diagrama de casos de uso - Estender funcionalidade da subscrição atual.....	25
Figura 5.1 - Diagrama de arquitetura de alto nível dos componentes do sistema	27
Figura 5.2 - Arquitetura de camadas dos componentes a implementar.....	29
Figura 5.3 - Diagrama de classes do Backoffice de Cliente	30
Figura 5.4 - Diagrama de arquitetura MVC (ASP.Net)	31
Figura 5.5 - Diagrama de comunicação do B.O Cliente com Stripe	32
Figura 5.6 - Modelo ER de tabelas com grande impacto na implementação.....	33
Figura 6.1 - Backoffice de Cliente - Página inicial.....	39
Figura 6.2 - Caixa de diálogo para confirmar cancelamento de subscrição.....	40
Figura 6.3 – BO Cliente: Ciclo de autenticação e validações necessárias	43
Figura 6.4 - BO Cliente: Gráficos disponíveis na página inicial.....	44
Figura 6.5 – BO Cliente: Caixa de diálogo para convidar possíveis afiliados.....	47
Figura 6.6 - BO KAMAE: Dashboard de Pagamentos.....	48
Figura 6.7 - BO KAMAE: Gráfico de pagamentos por meio de pagamento	49
Figura 6.8 - BO KAMAE: Gráfico de próximos pagamentos	49
Figura 7.1 - Matriz de Exposição de riscos.....	57
Figura 8.1 - Página inicial do backoffice	59
Figura 8.2 - Página de consulta dos dados de subscrição.....	60
Figura 8.3 - Formulário de edição dos dados de subscrição.....	60
Figura 8.4 - Página de consulta de conta corrente e faturação	61

Figura 8.5 - Página inicial do backoffice de subscrição suspensa	61
Figura 8.6 - Página de gestão de afiliados	62
Figura A.1 - Hierarquia de Contexto no S4L	69
Figura E.1 - Formulário de dados de faturação	101
Figura E.2 - Formulário de dados de faturação - Sucesso a gravar dados alterados.....	101
Figura E.3 - Formulário de dados de pagamento	102
Figura E.4 - Formulário de dados de pagamento - Sucesso a gravar.....	102
Figura E.5 - Formulário de dados de subscrição - Sucesso a gravar.....	103
Figura E.6 - Formulário de dados de subscrição - Erro ao gravar	103
Figura E.7 - Página de histórico de operações	104
Figura E.8 - Reset dos dados de front-end	104
Figura E.9 - Suspende a subscrição	105
Figura E.10 - Convidar outras pessoas para subscrever o serviço.....	105
Figura F.1 - Upgrade com Stripe	107
Figura F.2 - Downgrade com Stripe	107
Figura F.3 - Suspende com Stripe.....	107
Figura F.4 - Downgrade com Stripe	107
Figura F.5 - Suspende com Stripe.....	107
Figura F.6 - Suspende com Stripe.....	107
Figura F.7 - Cancelar com Stripe.....	107
Figura F.8 - Downgrade com Stripe	107
Figura F.9 - Downgrade com Stripe	107
Figura F.10 - Downgrade com Stripe.....	107

Lista de Tabelas

Tabela 2.1 - Comparação de soluções	9
Tabela 4.1 - Classificação de requisitos	17
Tabela 4.2 - Requisitos Funcionais	20
Tabela 4.3 - UC31 - Backoffice de Cliente: Ver histórico de comissões nos afiliados	21
Tabela 4.4 - UC32 - Backoffice de Cliente: Configurar dados de recebimento de comissões	22
Tabela 4.5 - UC33 - Backoffice de Cliente: Cancelar operação pendente	23
Tabela 6.1 - Tabelas criadas no decorrer do estágio.....	37
Tabela A.1 - Comparação de Soluções existentes.....	73
Tabela C.1 - UC01: Backoffice de Cliente - Gerir Subscrição.....	77
Tabela C.2 - UC02: Backoffice de Cliente - Consultar dados de subscrição	78
Tabela C.3 - UC03: Backoffice de Cliente - Alterar dados de Subscrição	79
Tabela C.4 - UC04: Backoffice de Cliente - Estender funcionalidade da subscrição atual...79	
Tabela C.5 - UC05: Backoffice de Cliente - Fazer upgrade da subscrição	80
Tabela C.6 - UC06: Backoffice de Cliente - Fazer downgrade da subscrição.....	81
Tabela C.7 - UC07: Backoffice de Cliente - Suspende subscrição paga.....	82
Tabela C.8 - UC08: Backoffice de Cliente - Retomar subscrição suspensa	83
Tabela C.9 - UC09: Backoffice de Cliente - Comprar novas licenças.....	83
Tabela C.10 - UC10: Backoffice de Cliente - Alterar dados de pagamento	84
Tabela C.11 - UC11: Backoffice de Cliente - Alterar dados de faturação	85
Tabela C.12 - UC12: Backoffice de Cliente - Consultar espaço ocupado.....	86
Tabela C.13 - UC13: Backoffice de Cliente - Consultar registo de entrada e saída de utilizadores	86
Tabela C.14 - UC14: Backoffice de Cliente - Consultar conta-corrente e faturação.....	87
Tabela C.15 - UC15: Backoffice de Cliente - Descarregar fatura.....	87
Tabela C.16 - UC16: Backoffice de Cliente - Consultar informações de afiliado.....	88
Tabela C.17 - UC17: Backoffice de Cliente - Convidar Afiliados.....	89
Tabela C.18 - UC18: Backoffice de Cliente - Fazer reset dos dados	90
Tabela C.19 - UC19: Backoffice Kamae - Ver dashboard de pagamentos.....	90
Tabela C.20 - UC20: Backoffice Kamae - Ver detalhes no dashboard de pagamentos	91

Tabela C.21 - UC21: Backoffice Kamae - Listar Recorrência de pagamento	92
Tabela C.22 - UC22: Backoffice Kamae - Adicionar recorrência de pagamento	93
Tabela C.23 - UC23: Backoffice Kamae - Remover recorrência de pagamento	93
Tabela C.24 - UC24: Backoffice Kamae - Editar recorrência de pagamento	94
Tabela C.25 - UC25: Backoffice Kamae - Listar tipos de pagamento	95
Tabela C.26 - UC26: Backoffice Kamae - Adicionar tipo de pagamento.....	96
Tabela C.27 – UC27: Backoffice Kamae - Remover tipo de pagamento.....	96
Tabela C.28 - UC28: Backoffice Kamae - Editar tipo de pagamento.....	97
Tabela C.29 - UC29: Backoffice Kamae - Visualizar relatório de subscrições	98
Tabela C.30 - UC30: Backoffice Kamae - Visualizar utilizadores de uma subscrição.....	98

Lista de Acrónimos

SaaS	Software as a Service
REST	Representational State Transfer
HTTPS	Hyper Text Transfer Protocol Secure
HTTP	Hyper Text Transfer Protocol
BO	Backoffice
S4L	Solutions For Lawyers
ASP	Active Server Pages
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
MVC	Modelo Vista Controllo
API	Application Programming Interface
JS	JavaScript
IDE	Integrated Development Environment
UC	Use Case
RF	Requisito Funcional
IIS	Internet Information Services
ORM	Object-relational Mapping
ADO	ActiveX Data Objects
CRUD	Create Read Update Delete
EDM	Entity Data Model
BD	Base de Dados
TFS	Team Foundation Service
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
CDN	Content Delivery Network
SP	Stored Procedure

SDK	Software Development Kit
TLS	Transport Layer Security
PK	Primary Key
FK	Foreign Key
ER	Entity Relationship

Capítulo 1

Introdução

O presente relatório foi elaborado no âmbito da unidade curricular Dissertação/Estágio em Engenharia de Software, com o objetivo de dar a conhecer todo o trabalho desenvolvido durante o estágio curricular, que decorreu ao longo do ano letivo 2017/2018. Este estágio foi dividido em duas fases, sendo que a primeira correspondeu ao planeamento de todo o trabalho a desenvolver, e a segunda fase à implementação da solução proposta.

O projeto foi realizado nas instalações da empresa KAMAE [2], fundada em 1997 e sediada atualmente em Fala, Coimbra, tendo iniciado no dia 11 de Setembro de 2017 e terminado no dia 18 de Junho de 2018.

1.1. Enquadramento

O trabalho de um advogado é muito mais do que simplesmente defender os seus clientes, envolvendo inúmeras tarefas que estes profissionais têm que realizar no seu dia-a-dia laboral. De forma a garantir uma maior eficiência e também controlo do que é feito, é importante para o advogado ter registo de todas as tarefas e de todo o trabalho realizado no exercer da sua função, sendo também fundamental a existência de registos de tempos gastos com cada tarefa.

Para além das tarefas a realizar, nesta profissão existe também uma enorme quantidade de informação a reter, que deve estar disponível e ser de fácil acesso sempre que existe necessidade de consulta, como por exemplo a informação de clientes. O trabalho destes profissionais passa também pela emissão de documentos de faturação, que, apesar de existirem inúmeros softwares de faturação, acaba sempre por ser necessário anexar os registos de faturação aos dados do respetivo cliente.

Manter toda informação anteriormente referida em formato de papel, torna-se um grande desafio, pois mesmo que exista uma organização correta da informação, existindo uma pasta por cliente, a gestão de toda a documentação existente e fundamental para o trabalho destes profissionais acaba por se revelar bastante difícil. Em simultâneo à dificuldade de armazenar e gerir toda a informação em papel, também o controlo do trabalho realizado se torna bastante limitado, pois é difícil manter registo de todas as tarefas realizadas, principalmente quando se trata de uma sociedade de advogados onde existem vários funcionários.

A KAMAE tem como foco principal, o desenvolvimento de software de gestão e apoio à área jurídica, sendo o KamaeLei o seu principal produto. O KamaeLei é um software de gestão, desenvolvido com o objetivo de apoiar, facilitar e rentabilizar o trabalho dos

advogados, ajudando na gestão de várias áreas fundamentais ao seu trabalho, tais como clientes, processos, faturação, controlo de tempos, documentação, prazos, entre outros.

Atualmente, a KAMAE está à procura de alcançar uma internacionalização ao nível mundial, tendo em curso, como parte da estratégia para alcançar esse objetivo, o desenvolvimento de uma solução cloud, o S4L [3] (Solutions For Lawyers), recentemente assim renomeado, à data de início do estágio era ainda o KamaeLei Cloud.

O S4L é constituído por 4 componentes:

1. Um site público, onde os clientes podem obter informações sobre o serviço e subscreve-lo;
2. Uma aplicação Front-end, que será a ferramenta de trabalho utilizada pelo cliente para realizar operações relacionadas com o seu negócio;
3. Um Backoffice de cliente, para que este possa gerir a sua subscrição, que será integrado na aplicação front-end;
4. Um Backoffice de gestão de clientes e produtos, gerido pela equipa de apoio ao cliente e comercial da KAMAE.

Atualmente, o site público e o Front-end encontram-se em fase Beta, estando já disponíveis em 4 línguas, e em 2018 serão disponibilizados para mais 4 línguas. Por outro lado, os Backoffices estão ainda numa fase muito inicial, havendo a necessidade de avançar com o seu desenvolvimento, pois são vitais para o sucesso da solução.

De forma a garantir o sucesso da sua solução, a KAMAE propôs para este estágio o desenvolvimento do BO Cliente, com várias funcionalidades necessárias para o cliente gerir a sua ferramenta, e ainda a conclusão do BO KAMAE, que oferece funcionalidades de gestão da solução pelos elementos da empresa. Ambos os Backoffices são de extrema importância, pois, é fundamental garantir alguma autonomia a ambas partes e fornecer informação constante e atualizada. Por um lado, é importante para o cliente ter a possibilidade de gerir o que deseja ter disponível na sua ferramenta, sem ter que depender da KAMAE. Por outro lado, o facto do cliente ter maior autonomia também é vantajoso para a empresa, pois alivia o número de pedidos de suporte. No que diz respeito ao BO KAMAE, a sua importância é facilmente justificável pela necessidade que a empresa sente em ter sempre informação que permita avaliar o crescimento da sua solução e ainda ser possível, a qualquer momento, alterar os planos/recorrências disponíveis para subscrição ou qualquer outra informação comercial que possa levar ao constante crescimento do negócio.

Esta solução irá ter integração com outros componentes desenvolvidos pela KAMAE, como por exemplo uma aplicação mobile (iOS e Android) mas que estão fora do âmbito deste estágio.

1.2. Objetivos

O principal objetivo do estágio consiste em desenvolver os backoffices de cliente e da KAMAE, ainda em falta, de modo a completar a solução. Contudo, este estágio encontra-se dividido em duas fases, cada uma delas com objetivos distintos.

Na primeira fase, realizada ao longo do primeiro semestre, pretende-se que seja feito o levantamento e especificação dos requisitos em falta dos dois backoffices e também a arquitetura destas duas plataformas e da base de dados.

Na segunda fase do estágio, espera-se que as duas plataformas planeadas na primeira fase sejam implementadas, considerando que o BO da KAMAE tem alguns módulos já implementados. Sendo depois necessário fazer integração com o front-end já desenvolvido. À data final deste estágio, os dois backoffices devem estar já totalmente implementados.

1.3. Trabalho Realizado

No âmbito deste estágio, foi implementado um novo componente da solução S4L, o BO de cliente, que oferece várias funcionalidades de gestão da subscrição a utilizadores da ferramenta, desde que estes tenham permissões para tal. Todas as funcionalidades planeadas para este componente foram totalmente implementadas, no entanto surgiram alguns desafios que levaram ao atraso da implementação.

Após a fase de planeamento, surgiu o pedido de realização de algumas alterações à arquitetura inicialmente proposta, que afetaram substancialmente o desenvolvimento deste componente. Apesar de as alterações não interferirem diretamente com os requisitos planeados, trouxeram a necessidade de comunicação com serviços externos, que não estavam inicialmente planeados, para concluir as operações a realizar no BO cliente. Esta nova necessidade, obrigou à realização de uma análise aos novos serviços externos a integrar, pois era importante entender como estes se comportavam e o que era possível fazer. Estas alterações inevitavelmente resultaram num atraso ao início do desenvolvimento.

Para além disso, ainda no BO cliente, surgiu também a necessidade de implementação de 3 novos requisitos que não estavam previstos. Esses requisitos foram implementados e a sua análise foi acrescentada ao documento no capítulo existente para o efeito.

Por outro lado, no que diz respeito ao BO KAMAE, incluído neste estágio como um componente a terminar, pois à data de início do estágio encontrava-se já num avançado estado de desenvolvimento, foram implementados quase todos os requisitos planeados, ficando a faltam apenas 1 por implementar.

1.4. Estrutura do documento

O presente documento encontra-se dividido em 9 capítulos distintos, cada um com o seu propósito. O capítulo corrente, tem por objetivo enunciar e descrever o problema a resolver e os seus objetivos, fornecendo ao leitor uma ideia inicial do que se pretende com a realização do estágio em causa. No capítulo 2, descreve-se o estado da arte, fazendo uma análise a várias ferramentas concorrentes e identificando as principais diferenças entre cada uma das soluções analisadas e os backoffices S4L. No 3º capítulo é feito o levantamento de aspetos relevantes e com impacto no desenvolvimento do projeto, como metodologias de gestão de projeto, ferramentas utilizadas e o planeamento de cada uma das fases do estágio. Segue-se o capítulo 4, onde é feita a especificação de requisitos de ambos os backoffices,

recorrendo a uma técnica de utilização de casos de uso com a devida descrição e procedimentos. De seguida, é apresentada a arquitetura do sistema, no capítulo 5, onde é possível visualizar diagramas de diferentes níveis arquiteturais. No capítulo 6 é apresentado todo o trabalho realizado ao longo da segunda fase do estágio, descrevendo todos os aspetos relevantes neste projeto e quais as opções tomadas. No capítulo 7, é descrito o levantamento de riscos realizado, identificando os riscos e os respetivos planos de mitigação, de forma a evitar que estes riscos se tornem em problemas reais. Durante a primeira fase deste estágio, foi ainda realizado um protótipo do backoffice cliente, que é descrito no capítulo 8 e também, mais à frente, nos anexos do documento. Por fim, o 9º capítulo corresponde às conclusões retiradas, após conclusão do estágio descrito neste documento, no que diz respeito a todo o trabalho que foi desenvolvido e também ao trabalho que há ainda para fazer.

Existe ainda uma secção de apêndices, onde estão disponíveis várias informações que complementam os diversos capítulos do documento. A informação colocada na secção anteriormente referida, foi considerada pela estagiária como importante para completar a análise ao trabalho realizado durante o estágio.

Capítulo 2

Estado da Arte

Neste capítulo será feita uma análise ao estado da arte, apresentando algumas ferramentas semelhantes e os possíveis problemas que possam aparecer ao nível de gestão do S4L. Esta análise foi realizada com o objetivo de perceber a necessidade de desenvolvimento dos backoffices e entender melhor o contexto da solução, identificando quais as necessidades que possam existir, tanto do lado da KAMAE como do lado do cliente.

2.1. S4L

O S4L é uma solução, em desenvolvimento pela KAMAE, que procura ir de encontro às necessidades dos advogados, mitigando os vários problemas frequentemente enfrentados por estes profissionais. Neste momento, esta solução tem 2 dos seus componentes em fase beta, o site público, que tem o objetivo de fornecer informações e vender a solução, e o front-end, que corresponde à ferramenta de trabalho que se pretende vender no S4L, sendo que este último tem já vários clientes a utiliza-lo para gerir o seu trabalho diário e está disponível em 4 idiomas, português, inglês, espanhol e português do brasil.

O site público [3] é onde os possíveis utilizadores do sistema podem obter informações no que diz respeito ao front-end, como funcionalidades oferecidas e pacotes disponíveis para subscrever (e respetivos preços de subscrição), sendo também neste site que estes podem subscrever o serviço, acedendo a um assistente de subscrição disponível.

A aplicação front-end oferece várias funcionalidades, que em conjunto permitem que advogados individuais ou pequenas/médias sociedades tenham uma gestão adequada de todo o seu negócio. Este produto possibilita aos seus clientes manterem toda a informação do seu trabalho centralizada num único local, de acesso fácil e rápido por todos os utilizadores do sistema, considerando as respetivas permissões de acesso. Este estágio passa pelo desenvolvimento de outros 2 componentes desta solução, não menos importantes do que os já referidos, o BO KAMAE e o BO cliente. O backoffice KAMAE tem por objetivo fornecer aos vários elementos da empresa várias funcionalidades para gerir a ferramenta, o ambiente e também as vendas de forma rápida e centralizada. Do ponto de vista comercial, para além da necessidade de relatórios relacionados com as vendas, é importante a qualquer momento ser possível inserir um novo plano que passe a estar automaticamente disponível na subscrição de novos clientes, ou até um novo módulo, que represente uma nova funcionalidade da ferramenta e que, para além de fazer parte de um plano, possa ser adquirida como extra por clientes que tenham subscrito planos que não incluam essa funcionalidade.

Por outro lado, o BO cliente, tem por objetivo dar alguma autonomia ao cliente, no que diz respeito à gestão da sua ferramenta, garantindo a existência de várias funcionalidades que lhe permitam alterar a sua subscrição, consultar informações de espaço ou de utilização e

consultar a faturação relativa aos seus pagamentos no S4L, num local centralizado, independentemente do meio de pagamento que tenham escolhido no ato da subscrição.

2.2. Backoffice KAMAE

O Backoffice da KAMAE encontra-se já num avançado estado de desenvolvimento, tendo várias funcionalidades já implementadas, apesar de ainda não se encontrar em produção.

As funcionalidades que existem atualmente, permitem já obter várias informações importantes no que diz respeito às subscrições e funcionalidades oferecidas, assim como gerir (adicionar, editar ou remover) todos os aspetos que podem variar nas subscrições e nas ofertas da ferramenta.

É possível gerir planos de subscrição, servidores (pois o front-end S4L pode estar armazenado em vários servidores distintos), módulos (que correspondem a conjuntos de funcionalidades), campanhas (que podem ser por exemplo promoções nas subscrições) e até produtos, pois no futuro a KAMAE pretende evoluir para outras soluções SAAS, que devem ser também geridas neste BO. Contudo, no que diz respeito a pagamentos encontra-se ainda um pouco incompleto, não sendo possível gerir os tipos de pagamento que devem estar disponíveis para a subscrição, nem as recorrências de pagamento. Existindo ainda uma área para gestão de pedidos de suporte, que é mais direcionada à equipa de suporte KAMAE.

No que diz respeito a relatórios, este componente está já bastante completo, tendo uma vasta gama de relatórios que inclui vários gráficos informativos e indicadores. Sendo possível verificar o estado em que se encontra cada servidor (espaço de armazenamento, nº de subscrições, entre outros), assim como obter informações de armazenamento e acessos por subscrição. Existem também um dashboard global, com vários gráficos informativos relativamente às áreas principais (subscrições, pagamentos, pedidos de suporte e alocação de recursos), contudo, a empresa considera útil e fundamental a existência de um dashboard de pagamentos, de forma a realizar uma análise mais profunda no que diz respeito à evolução desta área.

Por fim, através deste BO, é também possível inserir mensagens que devem aparecer no front-end, uma funcionalidade bastante útil por exemplo para informar os utilizadores da realização de operações de manutenção.

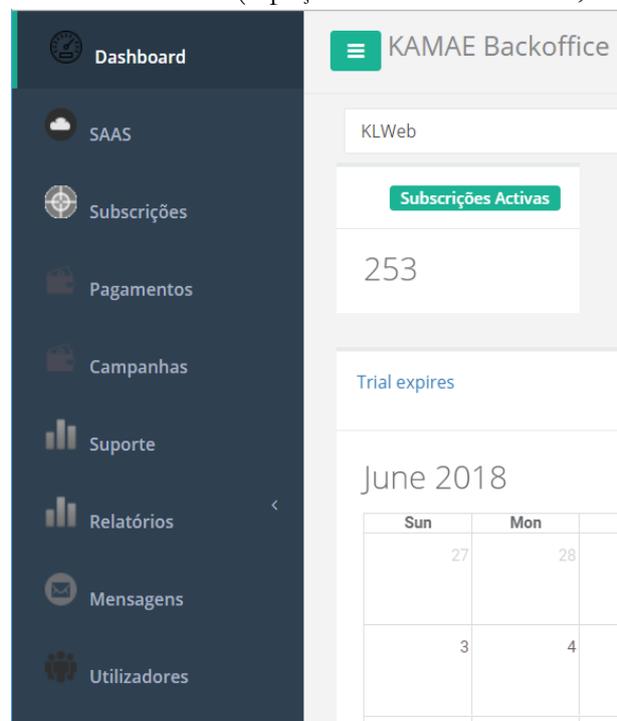


Figura 2.1- Menu do Backoffice KAMAE

2.3. Outras soluções Existentes

Tendo em conta que este estágio incide no desenvolvimento de 2 componentes bastante específicos que complementam o S4L, os backoffices da KAMAE e de cliente, a tarefa de encontrar soluções semelhantes tornou-se um pouco complexa. Esta complexidade justifica-se pelo facto de os componentes a desenvolver não corresponderem a um componente principal, mas sim de um meio para duas partes distintas (a empresa e o cliente) terem autonomia de gerir a atividade na ferramenta que é vendida, o S4L.

Apesar o elevado grau de especificidade destes backoffices, foram encontradas duas ferramentas que apresentam funcionalidades semelhantes ao que é pretendido e que poderiam perfeitamente ser uma alternativa à implementação proposta para este estágio.

De forma a analisar estas duas ferramentas, foi criada uma conta trial em cada uma das soluções, com o objetivo de observar as funcionalidades que cada uma oferece, confrontando as principais diferenças entre as várias soluções.

2.3.1. SamCart

SamCart é uma empresa criada em 2013, com o objetivo de comercializar uma plataforma de checkout para venda de produtos digitais, também esta com o nome SamCart.

Esta ferramenta oferece várias funcionalidades para vendedores incluindo uma vasta gama de relatórios, no entanto, não permite a existência de qualquer tipo de gestão por parte do cliente, ficando este totalmente dependente da equipa de suporte do vendedor, pois não existe uma área de comprador. Apesar da inexistência de área de comprador, o cliente não fica impossibilitado de fazer upgrades/downgrades ou até mesmo de atualizar os seus dados de pagamento, esta tarefa apenas se torna um pouco mais difícil, pois é necessário que o vendedor envie um email, através da sua área no SamCart, para que o cliente consiga realizar a operação desejada.

No que diz respeito ao produto para venda, apesar de ter a possibilidade de criar subscrições, não é possível criar vários planos dentro de um produto. Deste modo, será necessário criar um produto distinto por cada plano pretendido. Sendo que a funcionalidade de upgrade é chamada de upsell e os possíveis upsells devem ser criados pelo vendedor na plataforma.

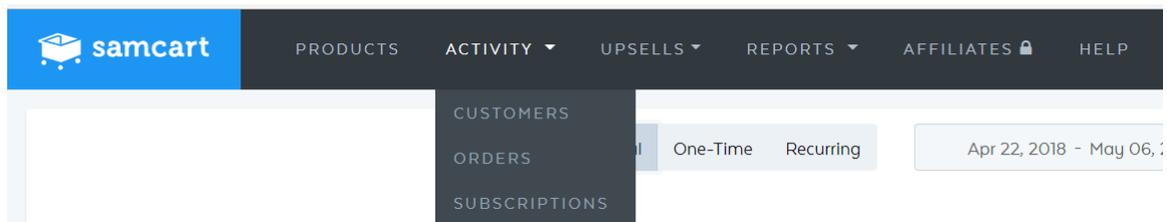


Figura 2.2 - Menu Samcart

2.3.2. Hotmart

O Hotmart é uma plataforma de venda e distribuição direcionada a produtos digitais, desenvolvida por uma empresa, com o mesmo nome, criada em 2011 e sediada no Brasil. Apesar de inicialmente esta empresa se ter focado principalmente no mercado do Brasil, tem apresentado um nível de expansão mundial bastante significativo, estando já presente em diversos países (mais de 170), incluindo vários países europeus.

Esta plataforma oferece diversas funcionalidades que a tornam bastante semelhante ao que se pretende com os backoffices da KAMAE. Pois, se por um lado permite ao comprador gerir todas as suas compras num só local e realizar várias operações sobre estas, por outro lado permite também ao vendedor gerir as suas vendas e obter vários tipos de relatórios e informações úteis à gestão do seu negócio.

O Hotmart trata de todo o processo de venda e faturação ao cliente e ainda da gestão de afiliados, desresponsabilizando assim a KAMAE destas tarefas, que, no entanto, não é o que se pretende. Relativamente a subscrições, que é o pretendido para o caso do S4L, por parte do comprador, é possível fazer upgrade/downgrade (tanto ao nível de pacote como de licenças/funcionalidades), cancelamentos e alteração de dados, sejam dados de pagamento ou de faturação. No entanto, não é possível suspender uma subscrição nem obter qualquer tipo de relatório com ela relacionado. Por outro lado, no que diz respeito a vendas, é possível introduzir os produtos desejados e a qualquer momento alterar os planos, recorrências de pagamento e até oferecer um período trial ou descontos. Relativamente a relatórios é um pouco mais limitado, pois, apesar de oferecer uma vasta gama de relatórios e possibilitar aplicação de filtros, deixa de ser possível consultar informações mais específicas, não tanto relacionadas com a venda em si, mas mais relacionadas com o ambiente e o nível de utilização por parte de cada cliente subscritor.

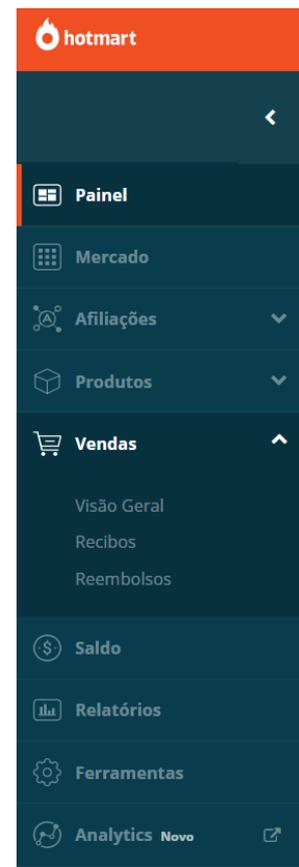


Figura 2.3 - Menu Hotmart

2.4. Comparação das soluções

De forma a reforçar a análise feita às soluções existentes, foram feitos alguns testes às várias plataformas, com o objetivo de identificar as funcionalidades que cada uma oferece.

	Backoffices S4L	SamCart	Hotmart
Integração com o S4L	✓	✗	✗
Atualização de dados do cliente	✓	✓	✓
Cancelamento da subscrição	✓	✓	✓
Upgrade/Downgrade	✓	✓	✓
Faturação	✓	✓	✓
Afiados	✓	✓	✓
Suspensão da Subscrição	✓	✗	✗
Relatórios	✓	✓	✓
Histórico de dados do S4L	✓	✗	✗
Reset dos dados	✓	✗	✗
Planos de pagamento	✓	✗	✓

Tabela 2.1 - Comparação de soluções

Na tabela 2.1 é possível observar quais as principais diferenças, ao nível de funcionalidades, entre as várias ferramentas. É visível que, após implementação dos Backoffices S4L, estes sejam os mais completos, pois sendo projetados e desenvolvidos pela KAMAE devem ir de encontro a tudo o que a empresa considera necessário para uma gestão adequada da sua solução.

No que diz respeito à autenticação e a dados mais específicos desta solução, nem o SamCart, nem o Hotmart podem ser integrados diretamente com o S4L, pois, sendo ferramentas externas não estão preparadas para uma integração a este nível. No entanto, é possível um outro nível de integração, pois qualquer uma das ferramentas tem um sistema de WebHooks que permite notificar o S4L sempre que ocorre algum evento (por exemplo uma nova subscrição um pagamento de uma subscrição existente).

Qualquer uma das ferramentas permite realizar atualização de dados de pagamento e/ou faturação, cancelar uma subscrição ou realizar upgrades/downgrade. Contudo, no caso do SamCart, qualquer uma destas operações passa por uma ação manual de um elemento da KAMAE, pois é necessário ir à página desta ferramenta indicar a operação no cliente pretendido, dado que este não possui uma área para que clientes possam gerir as suas subscrições, tornando o processo um pouco mais demorado.

Existem algumas funcionalidades, consideradas importantes pela KAMAE, que não estão presentes em nenhuma das plataformas testadas, como por exemplo a suspensão e reativação de uma subscrição.

Após esta análise, continua a ser importante a implementação dos backoffices, no entanto é importante referir que, caso se tivesse optado por utilizar uma das ferramentas analisadas, o Hotmart seria a melhor opção, pois permite que o cliente realize as operações desejadas de forma mais autónoma.

Capítulo 3

Abordagem

3.1. Metodologias

Em qualquer projeto de software é de extrema importância realizar controlo do produto adequado e consistente, desde o início até à entrega final do produto. A realização deste controlo é fulcral para o sucesso do produto, tendo por objetivo evitar que se perca o foco do propósito principal e avaliar frequentemente o que é necessário para alcançar o produto desejado, impedindo que haja aumento substancial de custos ou de prazos de entrega.

De forma a facilitar a realização de um controlo adequado, existem várias metodologias para gestão de projetos de software, que permitem o controlo de todo o desenvolvimento do produto. As várias metodologias de controlo dividem-se em dois grupos:

- **Metodologias tradicionais** - pouco ou nada tolerantes a mudanças, que devem ser aplicadas em projetos com requisitos bastante específicos e estáveis, passando sempre por uma longa fase de planeamento antes da implementação;
- **Metodologias ágeis** - mais flexíveis do que as tradicionais, permitindo a adaptação rápida a mudanças, pois o produto é desenvolvido em várias pequenas iterações.

3.1.1. SCRUM [13] – Metodologia ágil

Como exemplo de uma metodologia ágil, temos o SCRUM, que é descrito como um modelo de desenvolvimento de software incremental e iterativo. Este método passa pelo desenvolvimento de uma estratégia flexível de desenvolvimento produtos, onde uma equipa trabalha no sentido de alcançar um objetivo comum. No SCRUM a colaboração de todos os elementos da equipa é imprescindível, pois obriga a que haja comunicação diária entre todos, com a existência de pequenas reuniões diárias em que toda a equipa deve participar.

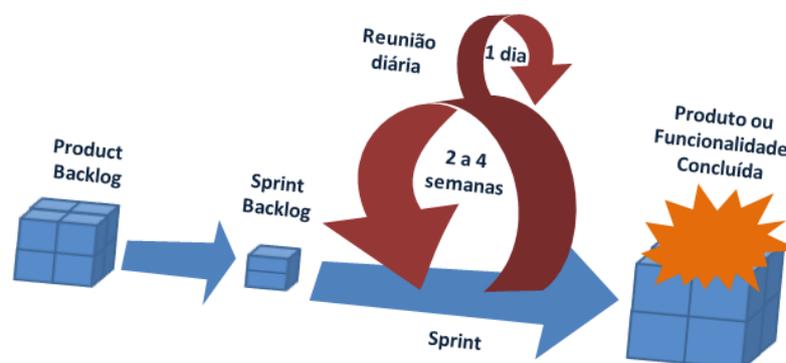


Figura 3.1 [12] - Ciclo do modelo SCRUM

Na figura 3.2 é possível observar o ciclo da metodologia SCRUM, que passa pelo “Product Backlog”, “Sprint Backlog”, sprint e, por fim, funcionalidades ou até o produto totalmente implementados.

- **Product Backlog** – Corresponde a uma lista ordenada de requisitos do produto a implementar. Qualquer elemento da equipa tem acesso a esta lista, podendo editá-la, mas a tarefa de ordenar os itens no backlog é da responsabilidade do gestor do projeto. De forma a ordenar a lista de forma adequada, devem ser feitas estimativas que devem ajudar o gestor a medir o tempo de realização de cada item, pois este fator pode ter influência na ordenação dos itens.
- **Sprint Backlog** – Corresponde a uma lista de objetivos que devem ser realizados pela equipa ao longo de um sprint. Passa pela divisão em tarefas dos itens definidos no Product Backlog e a realização das tarefas deve ser feita de acordo com a prioridade definida.
- **Sprint** – Dentro do sprint temos duas fases bastante importantes, as reuniões diárias e o sprint principal (que o desejável é que tenha uma duração entre 2 a 4 semanas). A reunião diária, chamada de Daily Scrum Meeting, é uma pequena reunião com duração de aproximadamente 15 minutos, em que todos os elementos da equipa de desenvolvimento devem estar presentes, que ocorre diariamente ao longo de todo o sprint. O objetivo desta reunião passa por cada elemento da equipa dar a conhecer a todos os outros elementos, o nível em que está o seu trabalho, respondendo às questões: **“O que fizeste nas últimas 24 horas?”**, **“Plano das próximas 24 horas?”**, **“Desafios encontrados?”**. A realização de reuniões diárias permite que eventuais problemas possam ser detetados, podendo ser discutidos e resolvidos logo após a reunião, reduzindo assim a probabilidade de serem arrastados por um longo período de tempo.
O sprint são as metas de trabalho definidas, sendo iniciados com uma reunião de planeamento, onde são identificadas as tarefas para esse sprint, e terminados com uma reunião de revisão, onde são identificadas quais as tarefas que foram cumpridas e as que ficaram por cumprir, analisando porque ficaram por cumprir. Cabe ao gestor de projetos a tarefa de planear e avaliar cada sprint.

Dado o grande envolvimento da KAMAE com a metodologia SCRUM, esta foi a escolhida no desenvolvimento dos componentes abrangidos pelo estágio, tanto para a primeira fase como para a segunda, com sprints de 30 dias.

- **SCRUM na KAMAE** - Na KAMAE, são realizados Sprints com duração de 30 dias, que consistem nas reuniões Daily Scrum e numa reunião de Sprint Review, que ocorre ao final de cada Sprint. Todos os elementos da equipa de desenvolvimento e de testes, incluindo o gestor de projetos, devem estar presentes tanto nas reuniões diárias de scrum, como no Sprint Review, independentemente dos projetos em que cada um está envolvido.
A reunião de Sprint Review tem uma duração aproximada de 1h30m e é iniciada pelo CEO da empresa, com o papel de Product Owner. Por outro lado, existe um elemento na equipa que tem o papel de Scrum Master, sendo o responsável pelo Sprint Backlog e por conduzir a reunião, passando a informação aos presentes de quais as tarefas e a

respetiva prioridade, registrando também as que foram ou não cumpridas no Sprint terminado e quais as que irão integrar o Sprint a iniciar. Após o Sprint Review, as tarefas que integram o novo Sprint, são afixadas pelo Scrum Master num local da empresa que é facilmente visível por todos os interessados.

3.2. Tecnologias e Ferramentas

À data de início deste estágio, as tecnologias e as ferramentas tinham já sido definidas pela empresa, dado que parte da solução estava já implementada, é necessário manter a coerência, utilizando as mesmas tecnologias e frameworks já utilizadas.

3.2.1. Ferramentas de controlo de Projeto

- **Asana** [10] – Ferramenta online para controlo de projeto, que permite lançar tarefas dentro dos vários projetos em desenvolvimento numa empresa, atribuindo um responsável pela sua realização e um prazo limite de conclusão. Dentro do Asana, cada elemento da equipa é responsável por dar por concluída cada uma das suas tarefas, sempre que termina a sua execução. Para além da organização já referida, esta ferramenta permite também uma divisão em sub-tarefas, de forma a facilitar o controlo do trabalho realizado, evitando assim a existência de tarefas com prazos demasiado longos. É também possível fazer comentários em todo o trabalho lançado, sendo que qualquer elemento da equipa o pode fazer, e verificar o histórico de operações dentro de cada lançamento (por exemplo alteração da data limite). O Asana permite identificar facilmente quais as tarefas em atraso e quais as com prazo de término mais próximo (assinalando as datas com cores distintas, vermelho no caso de atraso e verde no caso de data limite próxima).

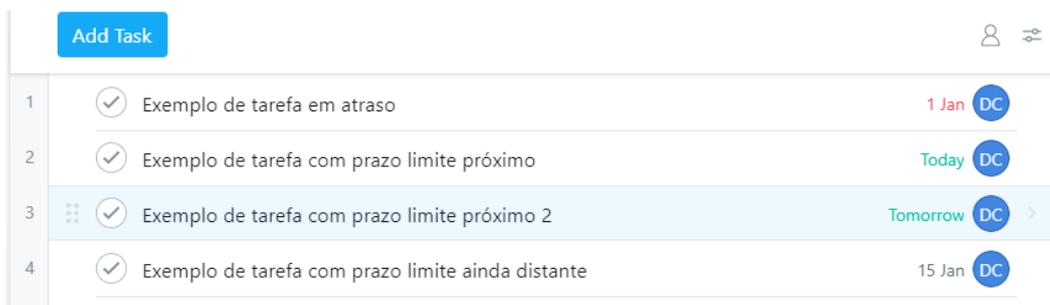


Figura 3.2 - Organização de tarefas do Asana

- **InstaGantt** [11] – Ferramenta online para construção de gráficos Gantt, com ligação direta ao Asana. É possível sincronizar os projetos e tarefas existentes no Asana, com os respetivos prazos, eliminando assim a necessidade de introduzir manualmente todas as tarefas e datas. Esta possibilidade de sincronização, simplifica bastante a realização deste tipo de gráficos, pois passa a ser apenas necessário ajustar as datas de início de cada tarefa, dado que as datas limite ficam imediatamente disponíveis.

- **TFS (Team Foundation Server)** [19] – Plataforma desenvolvida pela Microsoft, de integração com o Visual Studio, que incide numa gestão colaborativa do ciclo de vida de projetos. O TFS permite armazenar e partilhar com os restantes elementos da equipa, todo o código fonte do projeto, garantindo uma correta gestão de versões. Sempre que alguém faz “Check-in”, é criada uma nova versão e todas as anteriores são mantidas, sendo possível recuperar uma versão anterior a qualquer momento. Para além de gestão de código, esta ferramenta oferece ainda outras funcionalidades bastante uteis na gestão de um projeto, como controlo de Work Items (que podem ser Bugs, novos cenários ou tarefas).

3.2.2. Ferramentas de comunicação

- **Slack** [14] – Ferramenta de comunicação corporativa. O slack permite a existência de conversas e partilha de conteúdo dentro de uma equipa, que vai desde conversas mais restritas, entre apenas 2 ou alguns elementos da equipa, até canais de comunicação, onde todos os participantes de determinado canal recebem todo o que é lá partilhado.

3.2.3. Tecnologias e Ferramentas de desenvolvimento

- **ASP.net** [15] - É uma plataforma da Microsoft, para construção de sites e serviços web, baseada na framework .NET, utilizando HTML, JavaScript e CSS. Esta plataforma permite a construção de aplicações com 2 possíveis tipos de arquitetura, MVC ou Web Forms, sendo que a responsabilidade de escolher a arquitetura mais adequada é da equipa envolvida no projeto. Oferece ainda uma estrutura que facilita a construção de serviços HTTP, ideal para construir aplicações RESTful, o ASP.net Web API.
- **Bootstrap** [16] – Framework de front-end, que permite construir aplicações web responsivas, baseando-se em HTML, CSS e JS, de forma rápida e com bastante facilidade.
- **Microsoft SQL Server** [18] – Ferramenta de gestão de bases de dados SQL desenvolvida pela Microsoft, tendo como função principal armazenar e recuperar dados solicitados por outras aplicações.
- **Microsoft Visual Studio 2012** [17] – IDE de desenvolvimento de software, desenvolvido pela Microsoft, dedicado principalmente ao .NET e a linguagens como por exemplo o C, C++ e C#. Permite o desenvolvimento de aplicações web, com utilização da plataforma ASP.net.
- **Ngrok** [20] – O ngrok é uma pequena ferramenta que permite criar túneis seguros para um ambiente local (localhost). Esta ferramenta foi utilizada ao longo do estágio, na fase de implementação de funcionalidades que envolviam ligações a meios de pagamento e recebimento das notificações por eles enviadas. Por exemplo, no caso da realização de um upgrade, é enviado para a API do Stripe um pedido de atualização do plano subscrito. Neste caso, o webhook do Stripe dispara um evento que deve ser recebido na Web API KAMAEL a indicar a operação que foi realizada e os dados relevantes para a sua identificação. Não é possível fazer comunicação direta do webhook para a Web API em localhost, deste modo, o ngrok serve como

intermediário, criando um túnel que permite rececionar o evento, de forma segura, no ambiente localhost.

3.3. Plano de trabalhos do 2º semestre

Nesta subsecção será apresentado o gráfico Gantt correspondente à divisão de tarefas para o 2º semestre do estágio curricular. O número de horas a cumprir neste plano passa por 40 horas semanais, ao longo de 20 semanas, resultando num total de 800 horas.



Figura 3.3 - Plano de trabalho do 2º semestre

Na figura 3.3, pode-se visualizar o gráfico Gantt correspondente ao planeamento do trabalho a realizar no 2º semestre. A ordem das tarefas a realizar, foi definida com ajuda do orientador na empresa, e passou por identificar quais as tarefas mais prioritárias, de forma a ser possível colocar versões em testes desde uma fase mais inicial da implementação.

Tal como aconteceu com os gráficos do planeamento do 1º semestre (anexo B), também neste caso se optou por uma organização em semanas, de forma a facilitar a visualização. No entanto, revelou-se importante colocar as datas junto ao nome de cada tarefa, pois na 2ª fase do estágio o número de horas de trabalho é bastante superior ao da primeira e, deste modo, o plano torna-se mais perceptível.

Capítulo 4

Análise de Requisitos

A análise de requisitos é fundamental para identificar todas as funcionalidades a implementar no sistema e detalhar essas funcionalidades, de forma a identificar possíveis problemas e incoerências na especificação do sistema. Nesta secção serão especificados todos os requisitos a implementar nos dois componentes (Backoffices de cliente e KAMAE) e os casos de uso associados.

Na segunda fase deste estágio, surgiu a necessidade de adicionar 3 novos requisitos. A análise desses novos requisitos foi adicionada a este capítulo, sendo que estão identificados por *. Por outro lado, não foi possível implementar um dos requisitos do backoffice KAMAE, por motivos que serão explicados mais abaixo, esse requisito encontra-se identificado com *.

Na secção de anexos, existe ainda uma área dedicada a uma análise mais detalhada dos requisitos que tinham sido já definidos e planeados na primeira fase do estágio (Anexo C), que foi criada com o objetivo de detalhar os casos de uso necessários e o modo de funcionamento de cada um.

4.1. Especificação de Requisitos

Prioridade dos Requisitos:

De forma a classificar os requisitos, identificando quais os mais importantes e quais os menos importantes, será utilizada a classificação descrita na tabela abaixo (tabela 4.1).

Classificação	Descrição
Must	O requisito é imprescindível, tendo que fazer parte do sistema final.
Should	Se possível, o requisito deve estar contido no sistema.
Could	Requisito desejável, mas não necessário.
Won't	Requisito que não necessita de estar na versão a implementar, mas possivelmente importante numa versão futura.

Tabela 4.1 - Classificação de requisitos

4.2. Requisitos Funcionais

Na tabela abaixo (Tabela 4.2) estão identificados todos os requisitos planeados para o desenvolvimento do sistema que integra este estágio. Assim como uma breve descrição de cada um.

ID	Designação	Prioridade	Casos de Uso	Descrição
RF01	Consultar / Alterar dados de subscrição	Must	UC01, UC02, UC03	O utilizador pode alterar os dados associados à sua subscrição ou simplesmente consultá-los.
RF02	Estender a funcionalidade da subscrição atual	Must	UC01, UC04	O utilizador pode estender a funcionalidade da subscrição atual, adquirindo mais espaço de armazenamento disponível, ou até funcionalidades extra.
RF03	Fazer upgrade da subscrição	Must	UC01, UC05	O utilizador pode fazer upgrade da subscrição, adquirindo um pacote superior ao atualmente subscrito.
RF04	Fazer Downgrade da subscrição	Must	UC01, UC06	O utilizador pode fazer downgrade da subscrição, adquirindo um pacote inferior ao atualmente subscrito.
RF05	Suspender Subscrição Paga	Must	UC01, UC07	O utilizador pode suspender a subscrição temporariamente, deixando de ter acesso às funcionalidades e passando a pagar apenas 1€ por mês, enquanto a subscrição estiver suspensa.
RF06	Retomar subscrição suspensa	Could	UC01, UC08	Após suspender a subscrição, o utilizador pode retomá-la, voltando a pagar o valor subscrito e voltando a ter acesso às funcionalidades.
RF07	Comprar Novas licenças	Must	UC01, UC09	O utilizador pode comprar novas licenças de utilizadores, podendo criar novos utilizadores no front-end.
RF08	Alterar Dados de Pagamento	Should	UC01, UC10	O utilizador pode alterar os dados de pagamento da sua subscrição.
RF09	Alterar dados de Faturação	Should	UC01, UC11	O utilizador pode alterar os dados de faturação da sua subscrição.

RF10	Consultar espaço ocupado	Should	UC01, UC12	O utilizador pode consultar o espaço ocupado com os seus registos e o espaço que tem ainda disponível.
RF11	Registo de entrada e saída de utilizadores	Should	UC01, UC13	O utilizador pode consultar o registo de toda a atividade realizada no sistema, de entradas e saídas de utilizadores associados à sua subscrição.
RF12	Consulta conta-corrente e faturação	Should	UC01, UC14, UC15	O utilizador pode consultar a conta-corrente e faturação da sua subscrição, sendo possível descarregar as suas faturas.
RF13	Consultar Informações de Afiliado	Should	UC16, UC17	O utilizador pode consultar as suas informações de afiliado, no backoffice de cliente, sendo possível partilhar o seu link de afiliado com outros possíveis interessados, que possam vir a ser seus afiliados.
RF14	Fazer Reset a todos os dados já inseridos	Won't	UC18	Caso o utilizador não tenha feito documentos de faturação no sistema, pode fazer reset à sua aplicação, no backoffice de cliente, limpando todos os dados já inseridos, excepto utilizadores.
RF15	Gerir recorrências de pagamento	Must	UC21, UC22, UC23, UC24	O utilizador KAMAE pode adicionar uma nova recorrência de pagamento de subscrições. O utilizador pode também apagar ou editar uma recorrência já existente.
RF16	Gerir tipos de pagamento	Must	UC25, UC26, UC27, UC28	O utilizador KAMAE pode adicionar um novo tipo de pagamento de subscrições. O utilizador pode também apagar ou editar um tipo já existente.
RF17	Dashboard de pagamentos	Should	UC19	O utilizador KAMAE pode visualizar um dashboard de pagamentos, onde terá vários gráficos disponíveis (quantos

				pagamentos serão realizados no próximo mês e no próximo ano, número de subscrições com cada tipo de pagamento, número de clientes com pagamentos em atraso, número de clientes com pagamentos a realizar no mês corrente, número de clientes que pagam em cada dia do mês e do ano).
RF18	Consulta de detalhes no dashboard de pagamentos	Could	UC20	Dentro do dashboard de pagamentos, ao carregar em cima de um gráfico, o utilizador KAMAÉ pode visualizar uma tabela com os detalhes do gráfico clicado.
*RF19	Relatório de subscrições	Should	UC29, UC30	O utilizador KAMAÉ deve poder ter acesso a um relatório global de subscrições.
*RF20	Histórico de comissões	Should	UC31	O utilizador pode consultar o seu histórico de comissões de afiliados, no Backoffice de Cliente.
*RF21	Gerir dados de recebimentos	Should	UC32	O utilizador pode configurar onde deseja receber as suas comissões de afiliados, no Backoffice de Cliente.
*RF22	Cancelar operação pendente	Could	UC33	Existem algumas alterações à subscrição que apenas têm efeito no final do período já pago (operações de downgrade, suspensão ou cancelamento da subscrição). Enquanto essas operações se encontrarem no estado “Pendente”, o utilizador não pode realizar qualquer outra alteração que envolva pagamentos. Para evitar um “bloqueio”, o utilizador pode a qualquer momento cancelar uma operação que esteja pendente.

Tabela 4.2 - Requisitos Funcionais

Nome	*Ver histórico de comissões nos afiliados
ID	UC31
Descrição	O utilizador pretende visualizar o seu histórico de comissões, para verificar o que já recebeu e o que irá ainda receber.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de gestão de Afiliados.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no link “Histórico”, por baixo da tabela de afiliados; 2. A aplicação redireciona o utilizador para a página de histórico de comissões.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza a tabela com o seu histórico, onde pode ver comissões já recebidas e comissões que irá receber com a respetiva data de previsão do recebimento.

Tabela 4.3 - UC31 - Backoffice de Cliente: Ver histórico de comissões nos afiliados

Nome	*Configurar dados de recebimento de comissões
ID	UC32
Descrição	O utilizador pretende configurar os seus dados para receber as comissões de afiliados, podendo escolher transferência bancária ou paypal.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de gestão de afiliados.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no link “Gerir dados de recebimentos”; 2. A aplicação redireciona o utilizador para a página de configuração dos dados de recebimento de comissões; 3. O utilizador escolhe a forma como pretende receber as comissões (transferência bancária ou paypal); 4. A aplicação mostra os dados correspondentes ao meio de recebimento selecionado; 5. O utilizador preenche o formulário com os seus dados bancários ou dados da conta paypal; 6. O utilizador carrega no botão “Confirmar”; 7. A aplicação valida os campos preenchidos;

	<ol style="list-style-type: none"> 8. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 9. O utilizador carrega no botão “Confirmar”; 10. A aplicação valida a palavra-passe introduzida; 11. A aplicação mostra uma mensagem ao utilizador a informar que fez upgrade com sucesso
Fluxo Alternativo	<p>8a: O utilizador não preencheu os campos obrigatórios 8a1: A aplicação assinala a vermelho os campos obrigatórios não preenchidos; 8a2: Continua em 6.</p> <p>11a: A palavra-passe introduzida não está correta 11a1: A aplicação informa o utilizador de que a palavra-passe está incorreta; 11a2: Continua em 8.</p>
Pós-Condições	Os novos dados de recebimento definidos pelo utilizador são guardados na BD e o utilizador é redirecionado para a página de gestão de afiliados.

Tabela 4.4 - UC32 - Backoffice de Cliente: Configurar dados de recebimento de comissões

Nome	*Cancelar operação pendente
ID	UC33
Descrição	O utilizador pretende cancelar uma alteração à sua subscrição que está no estado pendente.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de dados de subscrição; 3. O utilizador tem uma alteração à subscrição no estado pendente.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Cancelar”, que se encontra junto à informação de operação pendente; 2. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 3. O utilizador carrega no botão “Confirmar”; 4. A aplicação valida a palavra-passe introduzida; 5. A aplicação mostra uma mensagem ao utilizador a informar que a operação foi cancelada com sucesso
Fluxo Alternativo	<p>5a: A palavra-passe introduzida não está correta 5a1: A aplicação informa o utilizador de que a palavra-passe está incorreta; 5a2: Continua em 2.</p>

Pós-Condições	O cancelamento é registado na BD e é feita a comunicação com o gateway de pagamento associado à subscrição para realizar o cancelamento.
----------------------	--

Tabela 4.5 - UC33 - Backoffice de Cliente: Cancelar operação pendente

4.3. Diagramas de Casos de Uso

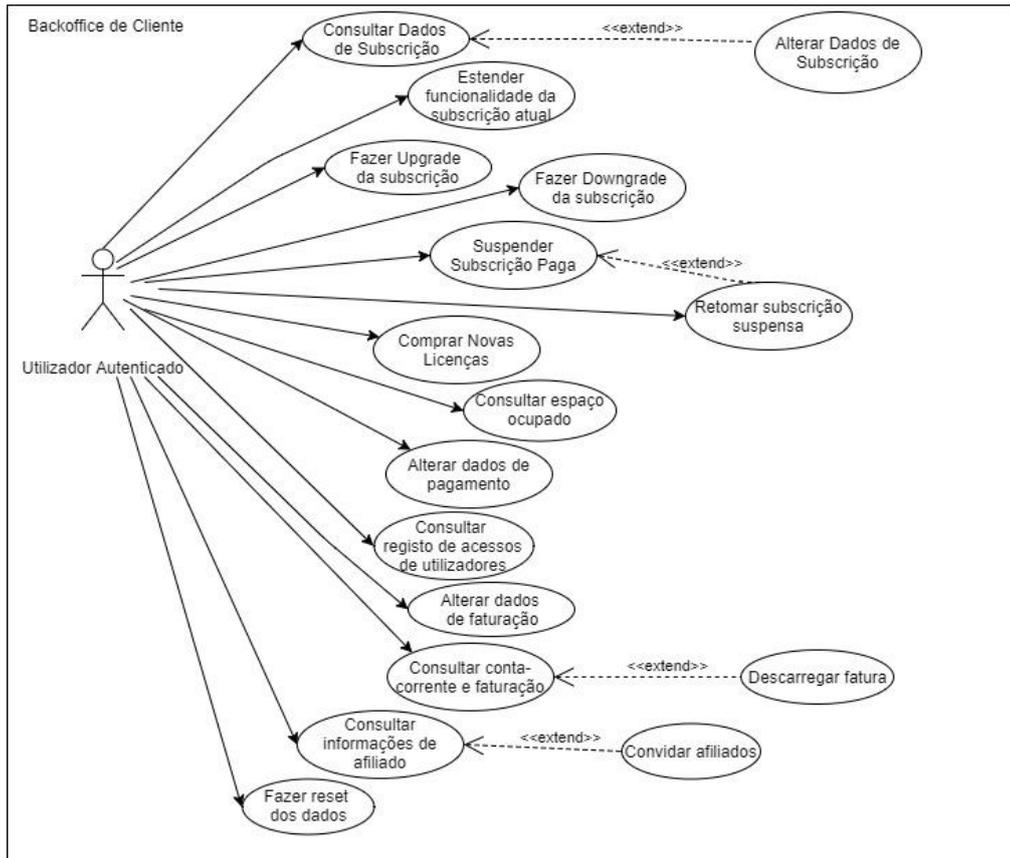


Figura 4.1 - Diagrama de Casos de uso completo do Backoffice de Cliente

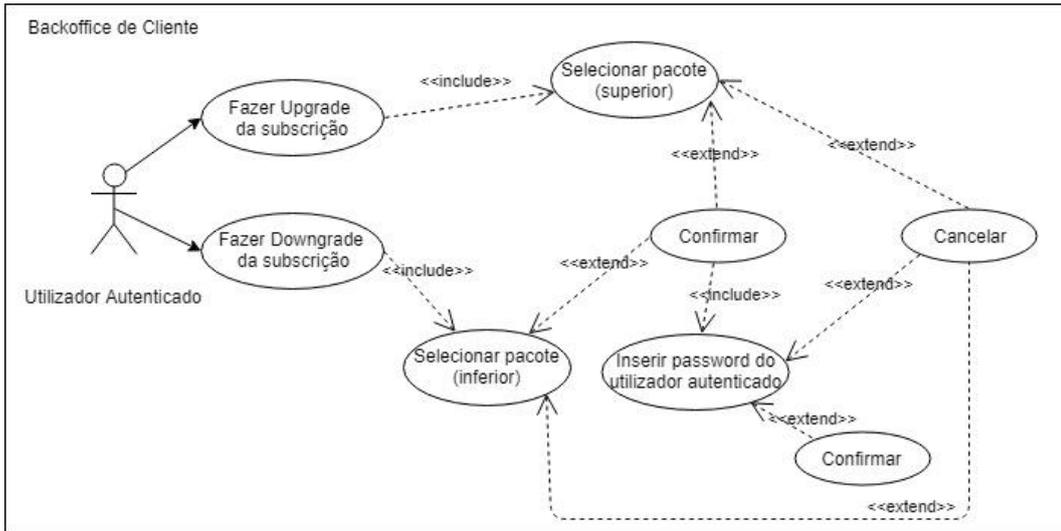


Figura 4.2 - Diagrama de Casos de Uso de downgrade/upgrade de subscrição

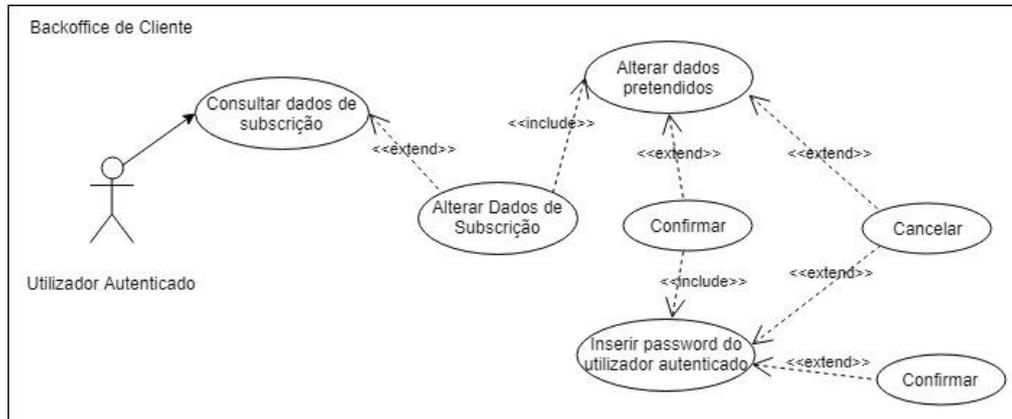


Figura 4.3 - Diagrama de casos de uso de consultar dados de subscrição

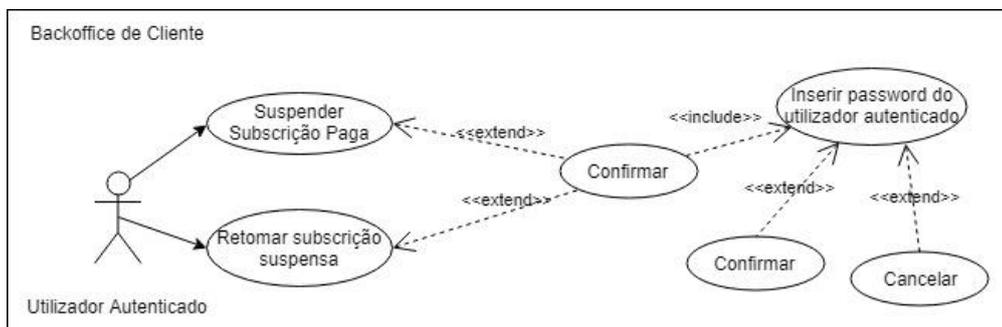


Figura 4.4 - Diagrama de casos de uso de suspender/retomar subscrição

Integridade – A aplicação deverá garantir que cada utilizador tem acesso apenas aos dados relativos à sua subscrição.

Interoperabilidade – O sistema tem que estar preparado para permitir facilmente a integração de novos componentes.

Capítulo 5

Arquitetura

A definição da arquitetura é uma fase imprescindível no planeamento de qualquer projeto, de forma a compreender, a analisar e demonstrar o funcionamento de toda a solução.

5.1. Análise da Arquitetura

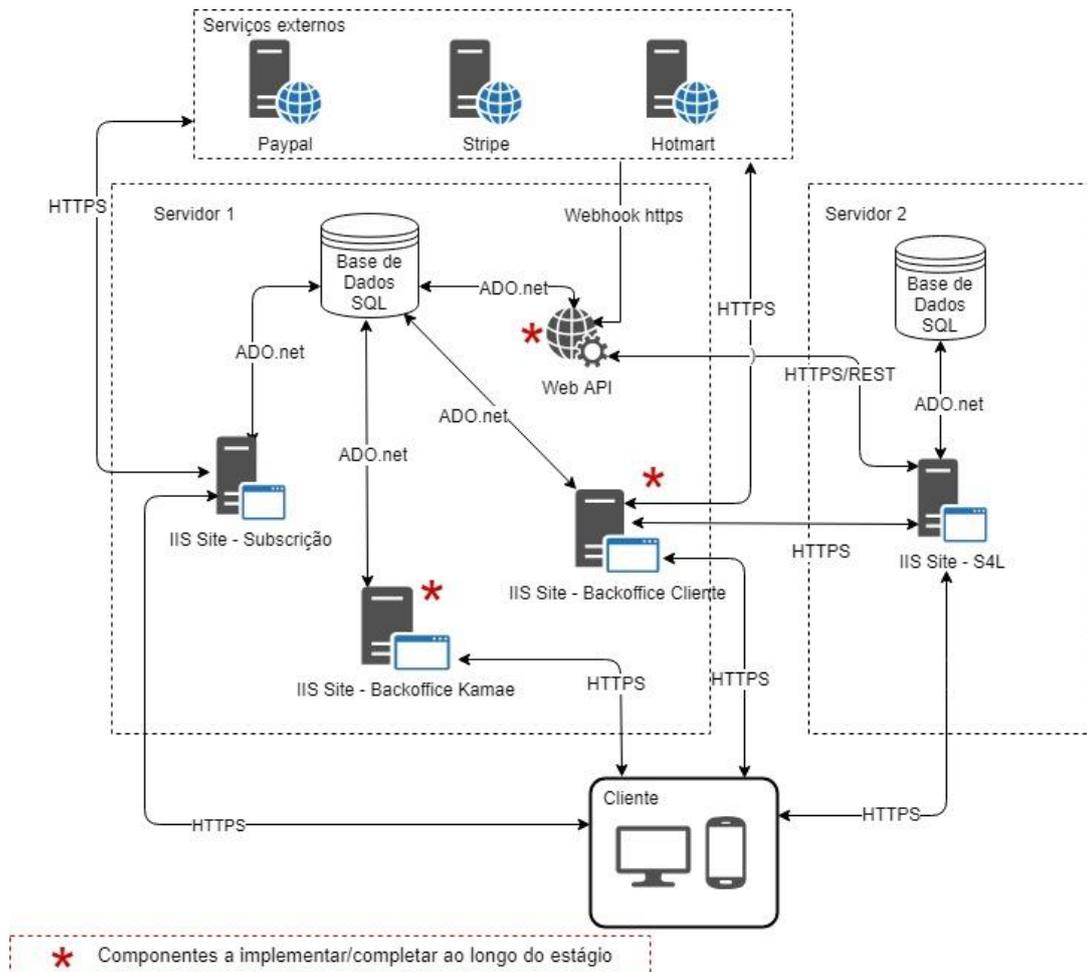


Figura 5.1 - Diagrama de arquitetura de alto nível dos componentes do sistema

Como é possível verificar no diagrama de arquitetura representado na figura 5.1, a solução S4L é composta por vários componentes. Contudo, existem 4 que são considerados principais, pois serão acedidos diretamente por utilizadores finais: o site público, que integra a subscrição; o BO da KAMAE; o BO de cliente; e, por fim, o site de frontend S4L. Para além dos componentes já referidos, todos os outros (2 bases de dados, uma Web API e os serviços externos), são também vitais ao bom funcionamento do sistema.

A existência de duas bases de dados distintas é justificada pela necessidade de manter toda a informação de subscrição e de backoffices separada da informação correspondente aos sites dos clientes S4L (registos inseridos pelos utilizadores no frontend). Estas bases de dados podem até estar instaladas em servidores distintos, pois não existe qualquer relação ou comunicação entre elas. No entanto, existem alguns registos que devem existir em ambas as bases de dados, cabendo à web API, a partilha destes registos, devendo estar alojada no mesmo servidor que a base de dados de backoffice, de forma a gerir corretamente a que subscrição pertencem os pedidos recebidos e quais as informações que devem ser partilhadas.

Tal como acontece com a web API, também os backoffices devem estar alojados no servidor na base de dados de BO, sendo assim possível aceder a todos os registos relacionados com os detalhes de subscrições e clientes da solução. Por outro lado, o site de frontend deve estar alojado no mesmo servidor onde se encontra a base de dados correspondente às informações inseridas neste site, de forma a que seja feita comunicação direta entre este site e a base de dados, utilizando entity framework.

O backoffice de cliente, apesar de também estar alojado no servidor de backoffice, apenas pode ser acedido através do site de frontend, necessitando de autenticação realizada neste último.

Por fim, aparece a necessidade de ligação a serviços externos, o strip, o hotmart e o paypal, todos eles com o objectivo de concretizar os pagamentos por parte dos clientes. Existem dois componentes que se ligam a estes serviços, o site de subscrição, que necessita de concretizar a subscrição, realizando o 1º pagamento e definindo as recorrências no provider escolhido pelo utilizador, e o BO de cliente, que por disponibilizar funcionalidades de upgrade/downgrade ou aquisição de extras, necessita de atualizar os pagamentos consoante as operações de cada cliente subscritor. Por outro lado, existem ainda uma ligação destes serviços externos à web API, que corresponde a notificações enviadas via WebHook sempre que ocorre um evento do lado de algum dos gateways de pagamento.

Todos os acessos do cliente aos sites, serão feitos utilizando o protocolo HTTPS, garantindo assim a segurança das ligações e dos dados transmitidos.

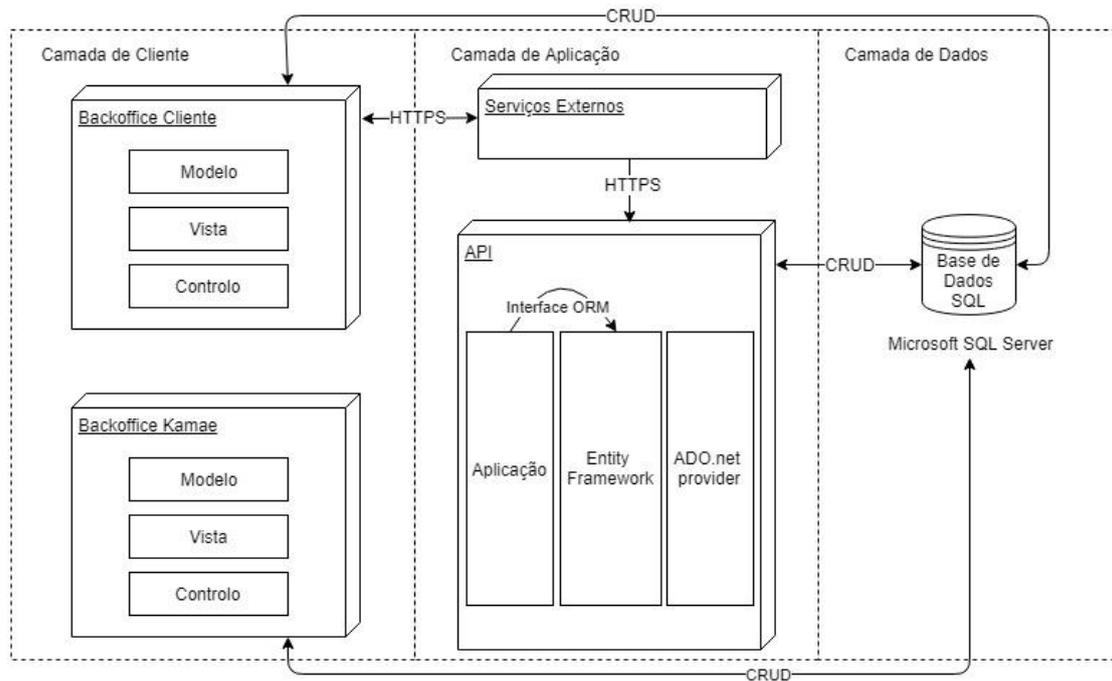


Figura 5.2 - Arquitetura de camadas dos componentes a implementar

Na figura 5.2, é possível visualizar um diagrama arquitetural que representa as 3 camadas do sistema, camada de cliente, camada de aplicação e camada de dados.

É possível verificar que ambos os backoffices, que se encontram na camada de cliente, respeitam o padrão de arquitetura MVC e ambos se ligam diretamente à base de dados para realizar operações CRUD. Para além desta ligação à BD, o BO Cliente liga-se ainda a componentes externos, que estão presentes na camada de aplicação. Sendo que esses componentes externos correspondem a gateways de pagamento (Stripe, no caso da implementação feita durante este estágio). Por outro lado, estes serviços externos, ligam-se também à web API KAMAE, com o objetivo de comunicar qualquer evento que ocorra (por exemplo um pagamento), recorrendo a um webhook. Todas as comunicações já referidas são realizadas em HTTPS, para garantir a segurança dos dados transmitidos.

Como é possível verificar no diagrama, o mapeamento dos dados provenientes da base de dados na web API, é feito utilizando entity framework. Entity framework é um ORM da Microsoft que permite transformar os dados da base de dados em objetos, facilitando a tarefa de acesso a dados, pois evita a necessidade de preocupação com tabelas e colunas. A transferência de dados para a aplicação é feita com base no EDM (modelo de dados da entidade), qua passa pela realização de um modelo conceitual dos dados na aplicação destino (neste caso, nos backoffices), descrevendo as entidades e respetivas relações da base de dados.

Por fim, é utilizada uma base de dados em Microsoft SQL Server, presente na camada de dados do diagrama, com o objetivo de armazenar todos os dados relacionados com as subscrições realizadas no S4L, e imprescindíveis para a gestão e controlo da solução e dos seus utilizadores.

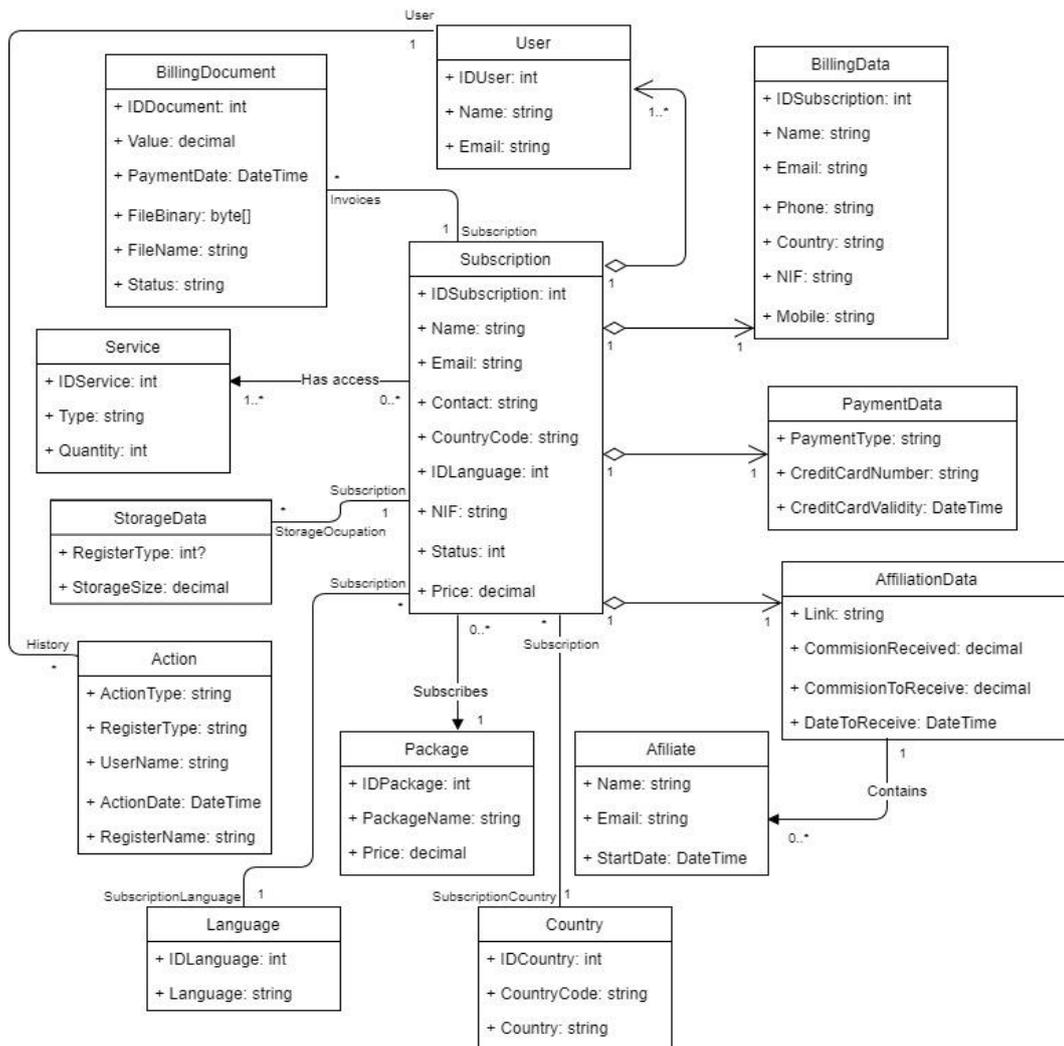


Figura 5.3 - Diagrama de classes do Backoffice de Cliente

Na figura 5.3 está representado um diagrama de classes correspondente ao Backoffice de cliente. Não se justificou a realização de um diagrama de classes de dados do Backoffice da KAMAE, pois, estando parte deste componente já implementado, o número de classes a implementar será bastante inferior ao necessário para o Backoffice de cliente, e possivelmente irão existir relacionamentos com classes já implementadas.

Neste diagrama é possível visualizar que a subscrição acaba por ser o centro de tudo, todos os relacionamentos passam de alguma forma por esta classe. Sendo este um sistema cloud, a subscrição é o que permite identificar todos os dados relativos a um determinado cliente, justificando assim o facto de tudo funcionar em torno desta classe.

A classe utilizador, é necessária e imprescindível para identificar o utilizador autenticado, mas, por questões de segurança, nunca será armazenada a palavra-passe do utilizador na aplicação, sendo necessário validar o utilizador sempre na base de dados, através da WebAPI.

As classes `BillingData` serve para armazenar os dados de faturação da subscrição, que podem ser editados pelo utilizador neste backoffice. Por outro lado, a classe `PaymentData` armazena os dados de pagamento, também estes editáveis pelo utilizador.

Relativamente à classe `BillingDocument`, esta é necessária para armazenar dados de conta corrente e respectivos documentos de faturação. A visualização do histórico da subscrição ou de espaço de armazenamento disponível, só é possível com a existência das classes `Action` e `StorageData`, respetivamente, pois estas permitem armazenar todas as informações de histórico e de armazenamento recebidas da webAPI na aplicação. A gestão de afiliados e de dados de afiliação é feita com recurso às classes `Affiliate` e `AffiliationData`.

Por fim, as classes `Package` e `Service`, permitem gerir o pacote associado à subscrição e os serviços a que os utilizadores podem aceder, respetivamente.

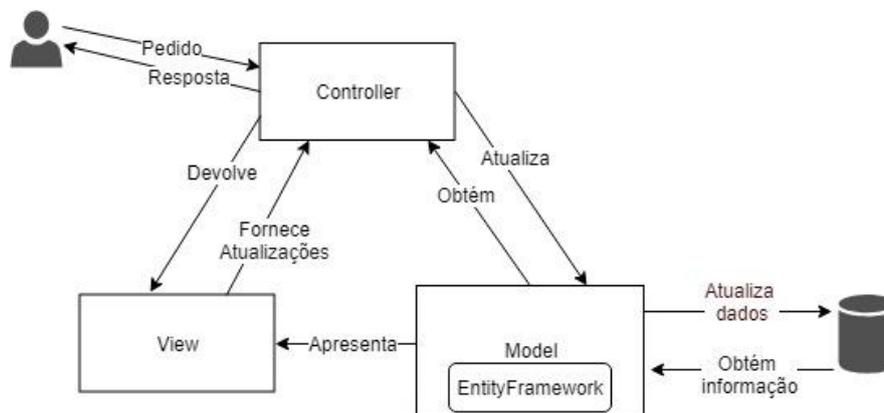


Figura 5.4 - Diagrama de arquitetura MVC (ASP.Net)

A figura 5.4 tem por objetivo apresentar a arquitetura MVC do ASP.Net, utilizado na implementação dos componentes que compõe o estágio apresentado no presente documento.

Este tipo de arquitetura envolve 3 componentes principais, o modelo, a vista e o controlo, com características e tarefas distintas. Sempre que um utilizador realiza um pedido numa aplicação ASP.Net, o controlador é responsável por obter o modelo que se adequa ao pedido do cliente e retornar a vista adequada com o respetivo modelo. Por outro lado, o trabalho do modelo dentro deste processo, é obter os dados necessários da base de dados.

Por fim, a tarefa da vista passa por apresentar o modelo que lhe foi fornecido pelo controlador, numa interface adequada ao pedido do utilizador. Este componente é também responsável por enviar para o controlador os dados necessários para que este atualize o modelo, que por sua vez deve atualizar a informação armazenada na BD.

Deste modo, o mais importante a reter deste diagrama é que, o modelo gere as informações da base de dados, a vista é responsável por apresentar a interface com os dados provenientes do modelo, e o controlador interliga estes 2 componentes, decidindo qual é o modelo e a vista que devem ser apresentados ao utilizador.

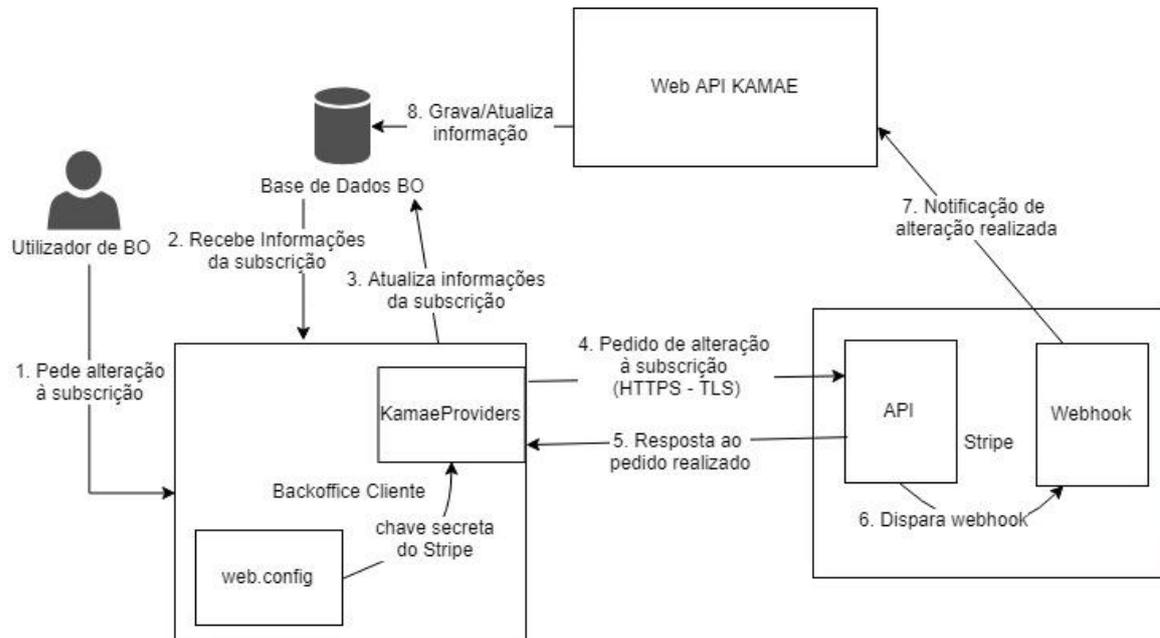


Figura 5.5 - Diagrama de comunicação do B.O Cliente com Stripe

Na figura 5.5 é possível visualizar um diagrama que contempla uma visão geral sobre como funciona a comunicação com o Stripe. Este diagrama tem por objetivo demonstrar quais os passos necessários para realizar a integração com este gateway e quais as tecnologias e componentes envolvidos em todo o processo.

O procedimento é iniciado com um pedido para realizar uma alteração à subscrição (upgrade, downgrade, cancelamento, suspensão, reativação ou aquisição de extras) por parte do utilizador. Após confirmação do pedido, cabe ao BO ir à BD recolher todas as informações que forem necessárias à atualização em causa, e também atualizar já o que for necessário, sendo que todas as alterações realizadas ficam pendentes até confirmação do meio de pagamento associado à subscrição. De seguida, utilizando a referência ao projeto KamaeProviders (projeto responsável por realizar todas as chamadas às API de qualquer meio de pagamento que seja utilizado), é feito um pedido à API Stripe, com o objetivo de realizar a operação desejada, que responde imediatamente com indicação do estado do pedido realizado. Esta chamada é feita por HTTPS, utilizando TLS, de modo a garantir que as informações transmitidas estão em segura, pois tratam-se de dados sensíveis que, se apanhados por terceiros, podem causar prejuízos graves ao utilizador envolvido na operação.

Após a solicitação à API para que seja realizada a alteração, a ferramenta webhook disponibilizada pelo Stripe dispara uma notificação, que é enviada para a Web API da KAMAE. Assim que recebe a notificação, a Web API realiza todas as operações necessárias para concluir o procedimento, confirmando as alterações, em caso de sucesso, ou, caso tenha ocorrido algum problema, revertendo as alterações realizadas.

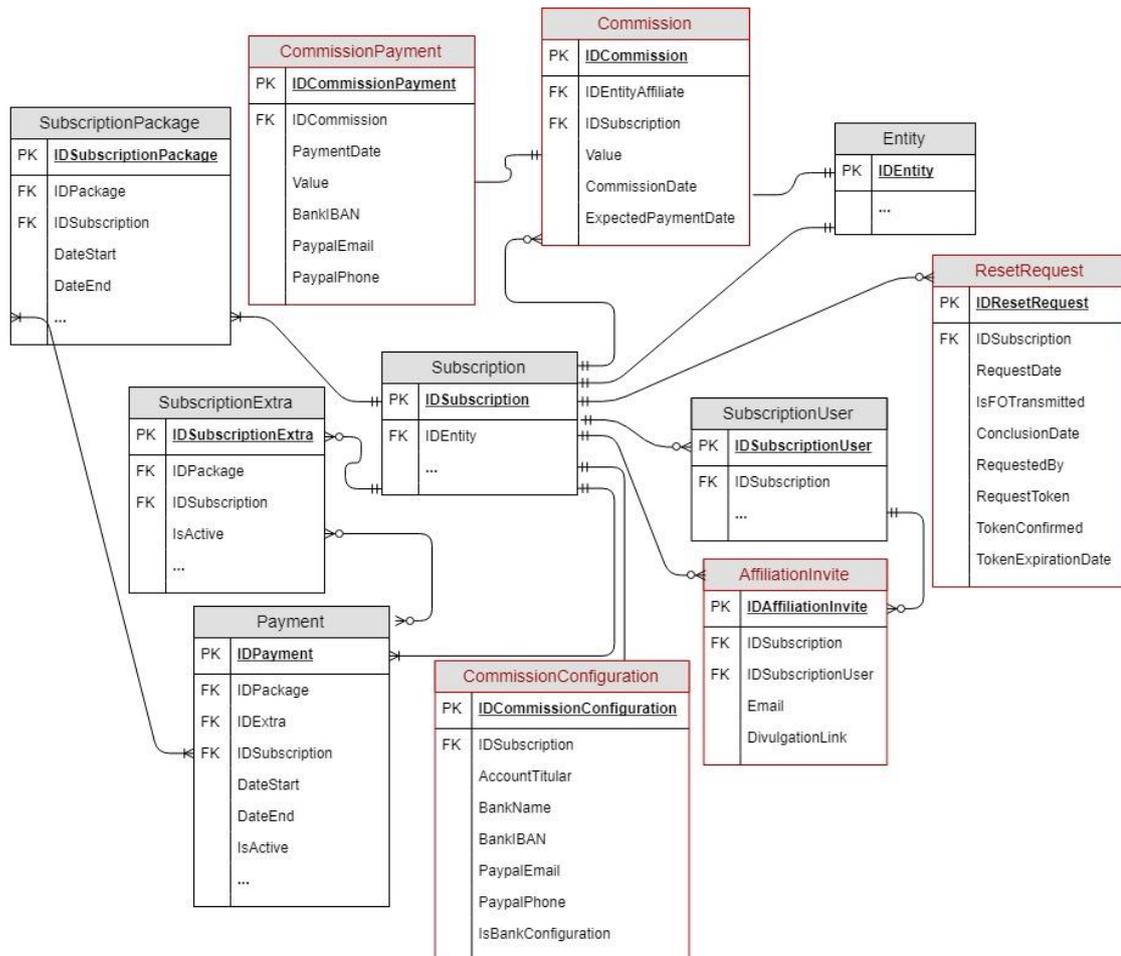


Figura 5.6 - Modelo ER de tabelas com grande impacto na implementação

O diagrama da figura 5.6, pretende demonstrar algumas das tabelas que foram utilizadas e foram imprescindíveis à implementação das funcionalidades a desenvolver ao longo deste estágio.

Algumas das tabelas ainda não existiam e houve a necessidade de criá-las durante a implementação dos componentes, sendo que se encontram destacadas a vermelho. Contudo as restantes tabelas visíveis no modelo acima apresentado, apesar de já existirem, tiveram um papel fundamental ao longo de toda a implementação e, por esse motivo, não podiam ficar de fora do presente documento. Por não se considerar necessário, no que diz respeito a estas últimas, não são apresentados todos os atributos que as contituem, no entanto, os atributos considerados necessários para que o leitor entenda o trabalho realizado, estão representados no diagrama.

As tabelas “Commission”, “CommissionPayment”, “CommissionConfiguration” e “AffiliationInvite” foram criadas com o objetivo de completar a área de gestão de afiliados e todas as funcionalidade necessárias nesta área. Sendo que, têm por objetivo contemplar as funcionalidades que dizem respeito às comissões recebidas pela subscrição e ao envio de convites para contactos que possam vir a ser afiliados. Mais à frente, no capítulo 6, que diz respeito à implementação, será dada uma breve explicação à necessidade de cada uma das tabelas.

Por outro lado, a tabela “ResetRequest” foi criada para permitir que os utilizadores de backoffice façam reset a todos os dados introduzidos na ferramenta de trabalho.

No que diz respeito às tabelas que já existiam, foram fundamentais para gerir ciclos de pagamento e pacotes/extras associados a cada subscrição, sendo que “Payment”, “SubscriptionPackage” e “SubscriptionExtra”, foram as que tiveram mais impacto na fase de implementação do BO cliente.

Capítulo 6

Implementação

Neste capítulo, será apresentado e demonstrado o produto implementado ao longo da segunda fase, tal como todas as decisões tomadas que se revelaram importantes no decorrer deste estágio.

6.1. Aspetos de Implementação

No que diz respeito à implementação, existem alguns aspetos que tiveram que ser considerados e respeitados, pois são fundamentais para o bom funcionamento das aplicações.

A implementação dos componentes planeados para este estágio, foi realizado com recurso à plataforma da Microsoft ASP.Net. Esta plataforma oferece dois tipos de arquitetura distintos, web forms e MVC, e permite escrever código em várias linguagens, sendo as principais C# e Visual Basic. Neste estágio foi utilizada a arquitetura MVC com recurso à linguagem C#.

Para entender melhor o que foi implementado e o processo de desenvolvimento, é necessário primeiro perceber como funciona o MVC e a estrutura de um projeto ASP.Net, que já foi também referida e no capítulo 5 (com recurso a um diagrama).

6.2.1. Estrutura do Projeto

- **Web.config** - O ficheiro Web.config é fundamental em qualquer projeto ASP.NET, trata-se de um ficheiro de configuração, com uma estrutura XML. É aqui que são colocadas todas as configurações necessárias ao bom funcionamento do sistema, ou até parâmetros que possam ser fundamentais para a execução de algumas funcionalidades, mas que não tenham que ter valores fixos, podendo ser alterados a qualquer momento com o sistema em produção. Estes parâmetros devem ser colocados dentro do elemento `<appSettings>`. É também neste ficheiro que devem ser declaradas todas as connection strings, dentro do elemento `<connectionStrings>`, necessárias para realizar ligações às bases de dados.

O exemplo abaixo, mostra a declaração de um parâmetro necessário à aplicação, neste caso trata-se da chave privada necessária para fazer chamadas à API do Stripe.

```
<appSettings>
  <add key="StripeKey" value="[Chave API Stripe]" />
</appSettings>
```

O acesso a este parâmetro, sempre que necessário, é feito da seguinte forma:

```
ConfigurationManager.AppSettings["StripeKey"]
```

Por outro lado, o exemplo abaixo mostra a declaração de uma connection string, que deve ser feita na área própria para o efeito.

```
<connectionStrings>
  <add name="DBConnection" connectionString="Data
Source=[DataSource];Integrated Security=false; User=[user];
password=[password]; DataBase=[BDNome]" providerName="System.Data.SqlClient" />
</connectionStrings>
```

- **Model** – Camada Modelo da aplicação. Responsável pela manipulação de dados, sendo que também as classes do Entity Framework podem ser consideradas modelos.
- **View** – Camada Vista da aplicação. Representa toda a interface da aplicação, renderizando os dados provenientes do modelo de forma a mostrar a informação corretamente ao utilizador. Em ASP.NET corresponde a ficheiros construídos principalmente com HTML.
- **Controller** – Camada Controlo da aplicação. Responsável por controlar a interação do utilizador com a aplicação. Corresponde a uma classe simples (C# no caso deste estágio), que deve retornar as vistas que o utilizador deve visualizar em cada momento.

Exemplo de declaração de uma classe Controller:

```
public class HomeController : Controller {...}
```

- **Ficheiros javascript (.js)** – Ficheiro para criação de código javascript. Apesar de ser possível colocar o código javascript dentro das próprias vistas, é bastante útil mantê-lo separado. Esta separação é vantajosa pois passa a ser possível utilizar o mesmo ficheiro em várias vistas, podendo reutilizar os métodos em vários locais distintos, sem ter a necessidade de os criar em cada local necessário, assim basta declarar o ficheiro js em todas as vistas em que ele for necessário.
- **Ficheiros css (.css)** – Ficheiro para criação de estilos (css). Com ficheiros deste tipo acontece exatamente a mesma coisa que com os ficheiros js, em que a separação é benéfica por tornar possível a reutilização.

Os tipos de ficheiros anteriormente referidos (js e css), sempre que possível (i.e, sempre que se tratam de ficheiros externos), estão a ser carregados via CDN, ao invés de colocar os ficheiros dentro do projeto, pois é aconselhável a utilização deste modo de carregamento para conteúdos estáticos. A utilização de uma CDN, reduz o número de pedidos realizados ao servidor onde está armazenado o sistema. Por outro lado, optou-se pela utilização de ficheiros minificados, pois são mais leves, reduzindo o peso de carregamento da página.

6.2.2. Plugins relevantes

Ao longo da implementação, revelou-se necessária a utilização de alguns plugins JQuery, de forma a facilitar a criação de alguns componentes necessários. Dos plugins adicionados, é importante destacar o DataTables e o GoogleCharts, pois foram bastante úteis para criação de tabelas e gráficos, respetivamente.

- **DataTables** – Plugin JQuery para criação de tabelas. Bastante útil, pois, para além de criar as tabelas, oferece várias funcionalidades como filtrar, ordenar, exportar.

- **GoogleCharts** – Plugin JQuery, desenvolvido pela Google, para criação de gráficos. Revelou-se bastante útil, pois possui uma grande diversidade de gráficos, de implementação bastante fácil.

6.2. Base de Dados

Com o objetivo de implementar os dois componentes propostos para este estágio, BO KAMAE e BO Cliente, foi necessário interagir com a base de dados já existente no servidor de backoffice. Apesar desta base de dados estar já bastante completa, houve a necessidade de criar algumas tabelas que estavam ainda em falta para contemplar todas as funcionalidades planeadas. Contudo, algumas tabelas já existentes revelaram-se fundamentais para o sucesso da implementação. No capítulo 5, é possível visualizar um modelo ER com algumas das tabelas que fazem parte da base de dados.

6.2.1. Novas Tabelas

De forma a contemplar todos os requisitos planeados no decorrer deste estágio, foi necessário criar algumas tabelas na Base de Dados de Backoffice, que não estavam ainda previstas, mas que se revelaram fundamentais para o correto funcionamento do projeto.

Nome	Descrição
AffiliationInvite	Esta tabela serve para armazenar todos os convites que o utilizador de BO envia para os seus contactos.
Commission	Tabela necessária para registo das comissões que uma subscrição tem direito. Necessária no BO Cliente apenas para leitura de informação. Cada comissão tem que estar sempre associada a uma entidade, que corresponde à entidade do afiliado, e à subscrição que é beneficiada pela comissão em causa.
CommissionPayment	Tabela necessária para registo dos pagamentos de comissões que são feitos, por esse motivo, têm inevitavelmente que ter sempre uma comissão associada. Necessária no BO Cliente apenas para leitura de informação.
CommissionConfiguration	Tabela para armazenar as configurações para recebimento de comissões no BO de Cliente.
ResetRequest	Tabela para registo de todos os pedidos de reset de dados, que são feitos no BO Cliente.

Tabela 6.1 - Tabelas criadas no decorrer do estágio

A tabela 6.1 apresenta todas as tabelas que foram criadas no decorrer do estágio, indicando o nome e uma pequena descrição de cada uma. Contudo, é possível visualizar estas tabelas, com mais detalhe, no modelo ER apresentado no capítulo 5.

6.2.2. Outras tabelas

Para além das tabelas que foi necessário criar, algumas das tabelas que já existiam na BD, merecem alguma atenção neste documento. Pois foram fundamentais para a implementação de algumas funcionalidades e a ausência de referência a elas, pode levar a que o leitor não compreenda na totalidade a informação que se pretende transmitir.

- **Payment** – Tabela onde são inseridas as informações de cada pagamento relativas a uma subscrição. Os registos aqui inseridos não se tratam de cada pagamento que é realizado (essa informação é armazenada na tabela PaymentRegister), mas sim a configuração do pagamento, isto é, o meio de pagamento utilizado, a recorrência de pagamento, o valor a pagar, o que está a ser pago (idsubscriptionPackage ou IDSubscriptionExtra, nunca são considerados os 2 em simultâneo), entre outros. Sendo que, este tipo de registo tem sempre uma data de início (data em que foi ou será realizado o primeiro pagamento associado), uma data de próximo pagamento e uma data de fim, esta última pode ou não estar preenchida. Esta tabela é de máxima importância, tendo em conta que é a principal responsável pelo controlo de ciclos de pagamento no lado da KAMAE.
- **SubscriptionPackage** – Quase tão importante como a Payment, a SubscriptionPackage é a tabela onde são armazenadas as informações do pacote subscrito. Tem um estado (ativo ou inativo) e também uma data de início e uma data de fim, pois um registo SubscriptionPackage nunca deve ser eliminado nem alterado (à exceção da data de fim e do estado). Sempre que o pacote da subscrição muda, é necessário desativar o atual e inserir um novo registo.
- **SubscriptionStatus** – Tabela onde é armazenado o estado da subscrição (que pode ser ativo, suspenso ou cancelado).

6.2.3. Stored Procedures

Por outro lado, apesar da utilização de EntityFramework, revelou-se necessária a utilização de stored procedures. Uma Stored Procedure corresponde a um conjunto de comandos SQL, bastante semelhante a uma função, que pode receber parâmetros de entrada e retornar resultados das operações realizadas. As SP's são armazenadas na base de dados e podem ser executadas através de sistemas externos.

Para algumas operações, a utilização de SP's, torna-se mais eficiente, evitando a recolha de dados desnecessários da BD e até eliminando a necessidade de realização de vários pedidos consecutivos.

Um exemplo de utilização de SP's no sistema a implementar é, o caso de um upgrade, em que existem várias tabelas envolvidas, que necessitam de atualização de registos já existentes e/ou adição de novos registos (como a tabela onde fica registada a informação de pagamentos e a tabela onde é armazenada a informação relativa ao pacote subscrito). Nesta situação, a utilização exclusiva de EntityFramework, iria tornar o processo mais demorado, pois, por cada atualização ou inserção, seria necessária uma nova chamada à BD. Recorrendo a SP's, é feita apenas uma chamada e todas as operações necessárias são realizadas nesse único acesso à base de dados, tornando a operação mais eficiente.

A criação de uma Stored Procedure é feita da seguinte forma:

```
CREATE PROCEDURE [dbo].[spNome]
  (@Parameter1 int = null)
AS
BEGIN
  [Comandos SQL]
END
GO
```

Após a sua criação, a SP pode ser executada do seguinte modo:

```
EXEC [dbo].[spNome] @Parameter1 = [VALOR]
```

6.3. Backoffice Cliente

O Backoffice de cliente foi implementado em ASP.NET, utilizando um template responsivo, o Inspinia, que é construído com bootstrap, HTML5 e CSS. A escolha do template a utilizar e das tecnologias referidas foi uma decisão da empresa.

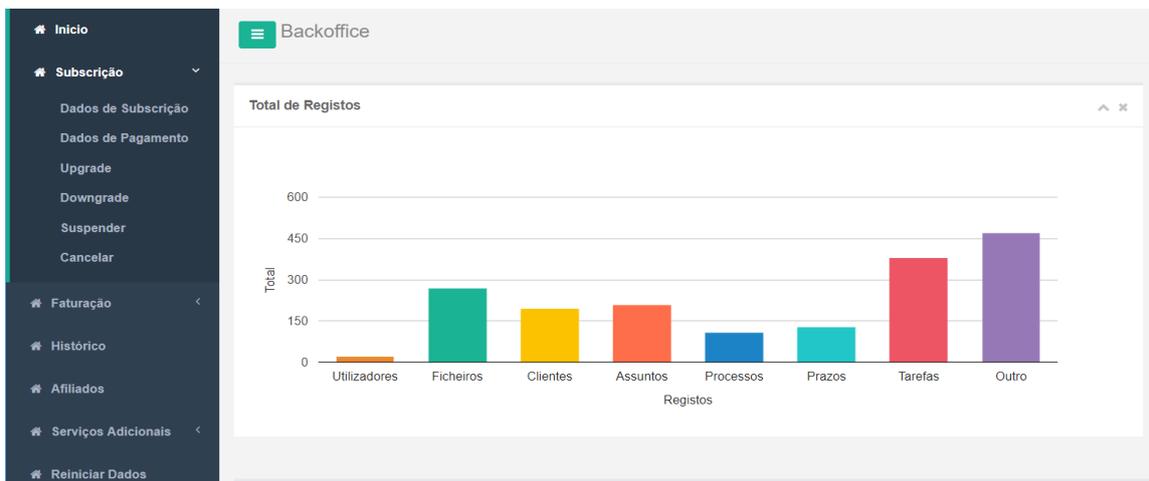


Figura 6.1 - Backoffice de Cliente - Página inicial

Todos os requisitos planeados para este BO, foram totalmente implementados. Contudo houve necessidade de fazer algumas reformulações substanciais e acrescentar 3 novos requisitos (já referidos no capítulo 4). As alterações necessárias deveram-se, principalmente, a uma reformulação significativa ao nível arquitectural, que passou pela eliminação de um componente inicialmente previsto (o proxy), sendo que esta alteração surgiu com a deslocação deste BO para outro servidor, e pela necessidade de comunicação com componentes externos (os gateways de pagamento).

A necessidade de integrar este componente com os gateways de pagamento, surgiu com o facto de várias das funcionalidades planeadas passarem pela realização de alterações à subscrição e aos serviços subscritos. Sempre que um utilizador de Backoffice faz uma alteração que afete a sua subscrição, é necessário informar o meio de pagamento inicialmente escolhido por este, na fase da subscrição, de forma a atualizar os valores a pagar ou os dados de pagamentos.

6.3.1. Envio de Emails de Confirmação

Algumas das funcionalidades implementadas, tratam-se de alterações à subscrição que, assim que tenham efeito tornam-se totalmente irreversíveis (cancelamento e reset dos dados), podendo resultar na perda total dos dados inseridos na ferramenta da subscrição em causa. De forma a aumentar a segurança na realização destas operações, optou-se por acrescentar mais uma validação necessária por parte do utilizador.

Esta validação de segurança acrescentada, passa pelo envio de um e-mail, com um token de confirmação, que deve ser introduzido pelo utilizador na mesma caixa de diálogo em que valida a operação com a sua palavra-passe.

A decisão de utilizar um token ao invés de enviar simplesmente um link de confirmação, prendeu-se com o facto de, neste caso, o utilizador ser “obrigado” a manter-se na página até confirmar a operação, não podendo ignorar o e-mail que lhe foi enviado. O envio de um link poderia induzir o utilizador em erro, pois, após confirmar a operação com a sua password, o utilizador poderia simplesmente ignorar o e-mail que lhe foi enviado e assumir que a alteração por si realizada tinha sido bem-sucedida e iria ter efeito em breve. O token é gerado por uma SP já existente na BD para criação de outros tokens é armazenado na tabela correspondente à operação em causa, sendo que o utilizador tem um prazo de 10 minutos para o utilizar. Na figura 6.2, é possível observar a caixa de diálogo que corresponde à confirmação de um cancelamento da subscrição.

Figura 6.2 - Caixa de diálogo para confirmar cancelamento de subscrição

Por outro lado, nas restantes operações que impliquem alterações à utilização da ferramenta, apesar de não ser necessária uma validação por e-mail, é enviado um e-mail a todos os utilizadores, após confirmação da alteração, para que todos tenham conhecimento de que poderão deixar de ter acesso a algumas funcionalidades ou passar a ter acesso a outras.

De forma a evitar a replicação de código sempre que é necessário enviar um e-mail, a estagiária optou por criar um método de envio de e-mail mais genérico, que possa ser chamado sempre que necessário com a passagem dos parâmetros que podem variar (os destinatários, o assunto e o corpo da mensagem):

```
public static void SendEmail(List<string> emails, string subject, string body){
    MailMessage message = new MailMessage();
    message.Subject = subject;
    message.From = new
System.Net.Mail.MailAddress(ConfigurationManager.AppSettings["SmptUser"]);
    foreach (string email in emails)
        message.To.Add(email);
}
```

```

message.Body = body;
message.IsBodyHtml = true;
System.Net.Mail.SmtpClient smtp = new
System.Net.Mail.SmtpClient(ConfigurationManager.AppSettings["SmtpServer"]);
smtp.Credentials = new
System.Net.NetworkCredential(ConfigurationManager.AppSettings["SmtpUser"], Confi
gurationManager.AppSettings["SmtpPass"]);
smtp.EnableSsl =
Convert.ToBoolean(ConfigurationManager.AppSettings["SmtpSSL"]);
smtp.Port = Convert.ToInt32(ConfigurationManager.AppSettings["SmtpPort"]);
smtp.Send(message);
}

```

Tal como é possível observar acima, os parâmetros necessários para envio de email (servidor SMTP, utilizador SMTP, SSL e porta SMTP), foram declarados no web.config, de forma a facilitar a sua alteração sempre a empresa o desejar.

6.3.3. Integração com Meios de Pagamento

Todas as funcionalidades com necessidade de ligação aos meios de pagamento, para que sejam concluídas com sucesso, foram totalmente integradas com Stripe, utilizando um SDK oficial, o Stripe.Net. Sendo este um SDK oficial, o Stripe garante que todas as conexões aos seus servidores são feitas através de TLS, garantindo assim que todos os pedidos são HTTPS, e os certificados são verificados em cada conexão realizada. Estas características são bastante importantes, pois as informações transmitidas são altamente sensíveis e é fundamental garantir a segurança de todos os dados dos clientes KAMAE.

A decisão de iniciar a implementação com a integração do Stripe, deveu-se ao facto de um outro componente da solução (a subscrição) estar também já totalmente integrado com este meio de pagamento. Assim, a falta de integração com os restantes meios de pagamento, não representa um atraso à entrada deste componente em produção, pois o importante é garantir consistência entre o que é fornecido na subscrição e o que é possível fazer no BO.

- **Stripe**

Para fazer a integração com este meio de pagamento, foi necessário instalar a biblioteca Stripe.Net, utilizando o Nuget, que é uma ferramenta de gestão de pacotes .NET, integrada no Visual Studio. Esta integração encontra-se detalhada num diagrama, que é possível encontrar no capítulo 5.

A integração é feita de duas formas distintas, por um lado, o Backoffice de Cliente comunica com este meio de pagamento através de uma API por eles fornecida, por outro, é feita a comunicação de todas as operações realizadas, para a Web API Kamae, utilizando um webhook.

Qualquer chamada à API do Stripe, envolve a utilização de uma chave secreta, fornecida por esta ferramenta, na área de programador, existente na conta de utilizador Stripe. Sendo que, todas estas chamadas, foram definidas num projeto à parte (KamaeProviders), devidamente referenciado no projeto do BO, e foram implementadas de forma genérica, sem qualquer ligação à base de dados, recebendo apenas os

parâmetros fundamentais à realização das chamadas e retornando os dados necessários para realização de operações posteriores. Esta separação foi feita para que estes métodos possam ser reutilizados em aplicações semelhantes que possam vir a ser desenvolvidas pela KAMAE.

Exemplo de uma chamada à API Stripe, para realização de um pagamento:

```
public static bool CreatePayment(string customerID){
    var payment = new Stripe.StripeCharge();
    var option = new Stripe.StripeChargeCreateOptions();
    option.CustomerId = customerID; option.Amount = 2.99;
    var service = new
Stripe.StripeChargeService(ConfigurationManager.AppSettings["StripeKey"]);
    StripeCharge result = service.Create(option);
}
```

Neste exemplo, o CustomerId é fundamental, pois, como em qualquer pedido realizado à API, é necessário identificar o cliente que está a realizar o pagamento, de forma a cobrar o valor ao cliente correto. Este CustomerId é obtido após a criação do cliente, seja através da API ou diretamente na página Web do Stripe, e deve ser sempre armazenado, por exemplo na Base de Dados, para que possa ser utilizado para qualquer operação que envolva o cliente em questão.

No caso do S4L, este campo já existia armazenado na BD, pois a primeira interação com o Stripe é feita na fase da subscrição, que já se encontra implementada e não faz parte do planeamento deste estágio. Sendo apenas necessário obter o CustomerId que pertence à subscrição do utilizador de BO a realizar a operação.

Foi necessário definir métodos para fazer os seguintes pedidos:

- **Verificar se o plano existe** – Isto é necessário para verificar se o plano já foi criado no gateway de pagamento, pois caso não exista, é necessário criar. Este pedido é necessário por causa dos extras, pois, para o Stripe, estes são identificados como planos distintos, que são associados ao pacote. Sempre que um utilizador adiciona um novo extra à sua subscrição, é feita esta verificação antes de qualquer outra operação. Para verificar se um plano existe, basta indicar qual o nome do plano pretendido.
- **Criar o plano** – Em sequência do pedido anteriormente referido, caso um plano não exista, é necessário pedir a sua criação, passando os dados necessários à criação do plano (como o valor, recorrência de pagamento, nome do plano, entre outros).
- **Obter informação do cliente** – Para qualquer alteração à subscrição de um cliente, é necessário identificar o cliente em causa. Considerando que o id correspondente ao cliente é sempre armazenado na BD, passando este parâmetro, é possível obter todas as informações necessárias a uma correta identificação do cliente e da sua subscrição dentro do Stripe.
- **Atualizar informações de cartão de crédito** – Este pedido é necessário pois existe um requisito com este mesmo objetivo. Para atualizar as informações de cartão de crédito, é necessário primeiro identificar o cliente, utilizando a chamada anteriormente referida, e de seguida atualizar o cliente com os novos dados.
- **Realizar um pagamento** – Em algumas alterações à subscrição, será necessário realizar pagamentos isolados para que essas alterações possam ter efeito (mais à frente

será abordada a justificação para estes pagamentos isolados). Para realização deste pedido à API, é necessário indicar qual o cliente (CustomerId) e qual o valor a cobrar no pagamento que vai ser realizado.

- **Cancelar uma subscrição** – Pedido necessário para cancelar uma subscrição ativa, sendo que é possível indicar que o cancelamento é imediato ou que deve ser apenas no final do período pago. Para indicar ao Stripe que uma subscrição deve ser cancelada, é necessário obter o id da subscrição a cancelar, que é retornado no pedido de informação do cliente.
- **Atualizar o plano de subscrição** – A necessidade de realização deste pedido passa pela execução de operações de upgrade/downgrade, nesta situação o que acontece é que o plano principal da subscrição irá ser substituído por outro, superior ou inferior. O pedido realizado não é nada mais do que uma simples atualização à subscrição, tendo recolhido anteriormente o id correspondente.
- **Adicionar/remover planos da subscrição** – Um pouco semelhante ao pedido anterior, é também uma atualização à subscrição. Contudo, neste caso em vez de simplesmente atualizar o plano principal, passa por adicionar novos planos ou remover planos “secundários” à subscrição existente (necessário para adicionar/remover extras).

6.3.3. Funcionalidades Implementadas

- **Autenticação**

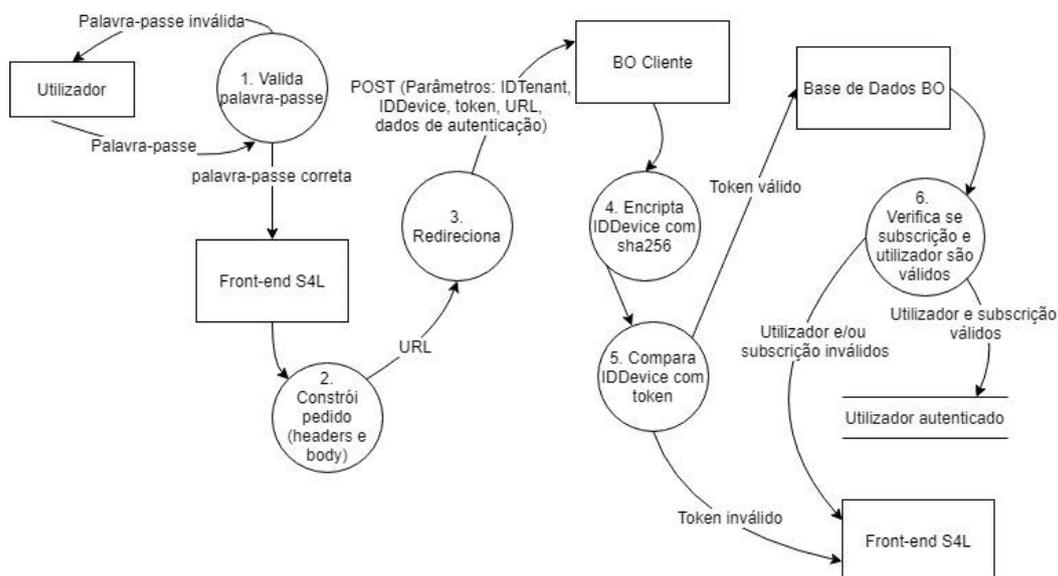


Figura 6.3 – BO Cliente: Ciclo de autenticação e validações necessárias

O momento de autenticação é bastante importante, existindo algumas validações necessárias para verificar que o utilizador tem acesso ao BO e a qual subscrição pertence. Tendo em conta que não existe uma página de autenticação, a primeira validação é feita do lado do front-end, pois é o único local que dá acesso a este componente. Esta primeira validação não foi da responsabilidade da estagiária, no entanto, todas as outras, realizadas já dentro do BO, tiveram que ser implementadas no âmbito deste estágio.

Quando o utilizador entra na página de BO, devem trazer alguns parâmetros fundamentais no body e no header do pedido, que foram inseridos no lado do front-end. Os parâmetros que devem fazer parte do pedido são: o IDTenant (que é o id responsável pela identificação da subscrição em todo o sistema), o IDDevice, um token (que corresponde ao IDDevice encriptado com o algoritmo sha256, utilizando uma chave de segurança), o URL da ferramenta front-end e os dados de autenticação do utilizador (email e palavra-passe, esta última encriptada também com o algoritmo sha256).

A primeira validação a realizar, é verificar se o token e o IDDevice correspondem. Caso esta validação seja bem-sucedida, todos os dados (à exceção do token) são enviados para a SP de autenticação BO, que verifica se existe uma subscrição com o IDTenant e o URL transmitidos, e se o utilizador existe, pertence à subscrição em causa e se tem permissões de BO. Caso todas estas condições se confirmem, o utilizador é autenticado com sucesso, podendo utilizar o BO, caso contrário, é redirecionado para o URL de onde veio.

- **Página Inicial**

A página inicial do BO Cliente, apresenta alguns gráficos e indicadores para que o utilizador possa visualizar facilmente informações relativas à sua ferramenta de trabalho S4L e ao espaço de armazenamento. Todos estes gráficos foram criados utilizando o plugin GoogleCharts, já referido neste documento.

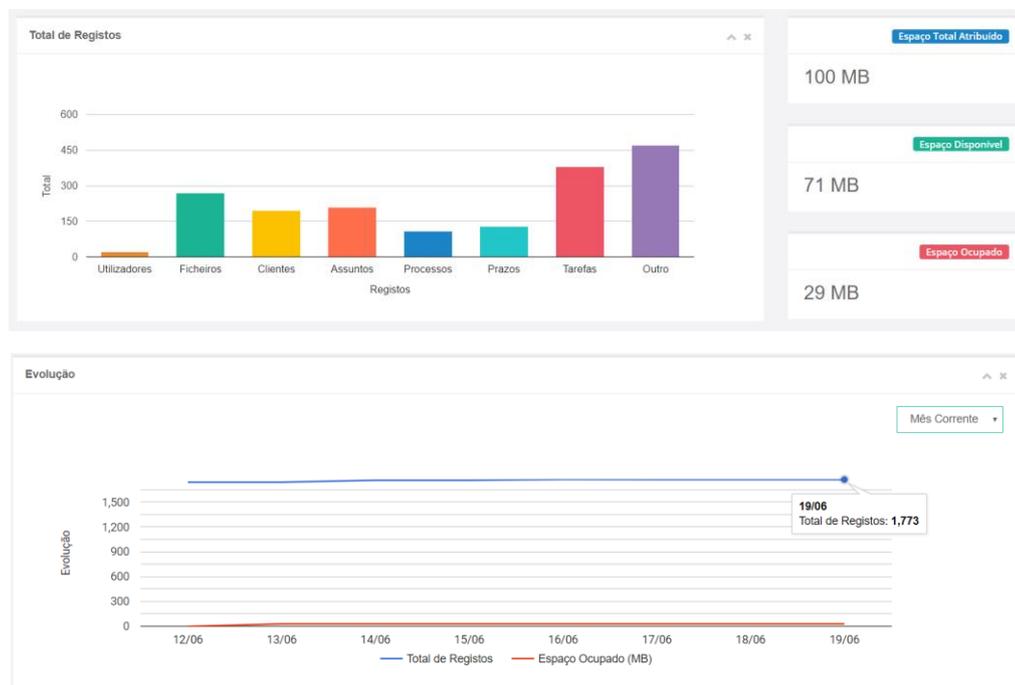


Figura 6.4 - BO Cliente: Gráficos disponíveis na página inicial

Como é demonstrado na imagem acima, os gráficos existentes são “Total de Registos” e “Evolução”. Já os indicadores são “Espaço Total Atribuído”, “Espaço Disponível” e “Espaço Ocupado”.

O gráfico “Total de Registos” mostra ao utilizador a quantidade de cada um dos tipos de registos que podem ser inseridos no front-end S4L. Por outro lado, o gráfico “Evolução”, permite ao utilizador observar como tem evoluído o trabalho ao longo de 4 possíveis períodos de tempo (mês corrente, mês anterior, ano corrente, ano anterior ou todos), no que diz respeito ao número de registos e ao total de espaço ocupado.

No que diz respeito aos indicadores disponíveis, todos eles estão relacionados com o espaço de armazenamento, sendo que a informação mais importante a reter pelo utilizador é o espaço ainda disponível.

Todas estas informações, provém da BD de FO e são inseridas na base de dados de BO através de um serviço que não se integra no estágio aqui apresentado.

- **Subscrição e serviços adicionais**

Dentro dos menus de Subscrição e de Serviços Adicionais, encontram-se várias funcionalidades que resultam em alterações aos serviços/funcionalidades que a subscrição contempla, e também ao preço final a pagar. Todas estas alterações, implicam comunicação com o meio de pagamento que é utilizado na subscrição afetada, de modo a que todas as alterações sejam transmitidas, evitando que sejam cobrados valores errados ao cliente.

As funcionalidades que passaram por esta integração foram as seguintes:

- **Alteração de Dados de Pagamento** – A necessidade de ligação aos meios de pagamento apareceu pois esta funcionalidade passa pela alteração aos dados de cartão de crédito. Considerando que os ciclos de pagamento e os débitos são geridos por meios externos, não há neessidade de armazenar este tipo de informações, que são extremamente sensíveis, na base de dados da KAMAE, pois já são armazenadas no gateway de pagamento associado à subscrição. Assim sendo, quando os dados são alterados devem imediatamente ser atualizados no meio de pagamento utilizado na subscrição em causa, para que os próximos pagamentos possam ser debitados do local correto.
- **Cancelamento** – Sempre que um utilizador de backoffice manifesta o desejo de cancelar a sua subscrição, o meio de pagamento deve ser imediatamente informado, para que não ocorra mais nenhum débito. Contudo, esta operação não tem efeito imediato, dado que o cancelamento apenas é realizado na data em que termina o período já pago, evitando assim a necessidade de reembolso.

No que diz respeito às atualizações que esta operação implica na BD:

- Todos os registos Payment que se encontrarem ativos no momento e sem data de fim definida, passam a ter a data de fim igual à data do próximo pagamento;
- O estado da subscrição é atualizado para cancelado;
- O registo de pacote da subscrição ativo passa a ter a data de fim igual à data do próximo pagamento.
- **Dowgrade** – Um downgrade, para além de implicar uma redução de funcionalidades, implica também redução no valor a pagar. Deste modo, é necessário indicar ao meio de pagamento, o valor que deve ser cobrado em débitos futuros, atualizando o plano existente dentro do gateway de pagamento. Tal como

acontece com o cancelamento, um downgrade apenas terá efeito no final do período já pago, evitando reembolsos.

No que diz respeito às atualizações que esta operação implica na BD, as informações de ciclo de pagamento e de pacote associado à subscrição devem ser atualizadas, de modo a que os registos atuais apenas fiquem ativos até à próxima data de pagamento e que sejam inseridos novos registos para contemplar o novo plano.

- **Upgrade** – A realização de upgrades, também necessita de comunicação com o meio de pagamento utilizado, pois implica alterações ao preço do serviço, que devem ser comunicadas, de forma a garantir que os débitos futuros terão o valor correto. Por outro lado, ao contrário do que acontece com as funcionalidades anteriormente referidas, neste caso a alteração tem efeito imediato. Para isso, é necessário realizar um pagamento imediato, que tem por objetivo cobrar a diferença de valor correspondente aos dias que faltam para terminar o período já pago, sendo que esta operação implica também um pedido à API do gateway.

Nesta situação, as alterações a realizar na BD são as mesmas que no caso do downgrade, com a diferença de que os registos anteriores ficam logo inativos e os novos registos têm início na data atual.

- **Subscrição/reativação de subscrição** – No Stripe, não existe a possibilidade de suspender subscrições ativas. Deste modo, foi necessário procurar uma alternativa, de forma a conseguir implementar esta funcionalidade. A alternativa encontrada foi, criar um plano para o estado suspenso de uma subscrição com o valor de 1€, e ativar este plano quando o utilizador manifestar o desejo de suspender a sua subscrição, alterando o plano ativo no Stripe pelo plano de suspensão. No caso contrário, e que o utilizador pretende reativar a subscrição, o plano deve ser novamente alterado para o anterior e o valor a pagar por cada período deve ser o valor que era pago antes da suspensão. Sempre que um utilizador manifesta o desejo de suspender a subscrição, a suspensão apenas terá efeito no final do período já pago. Contudo, o contrário não se verifica, sendo que a reativação é imediata.

Esta operação implica a atualização do estado da subscrição e também do registo de ciclo de pagamento, pois estando suspensa, a subscrição passa a ter um custo de 1€, sendo reativada volta ao preço anterior.

- **Aquisição de extras** – Esta funcionalidade pode implicar vários tipos de alterações distintas, dependendo do tipo de extra que tenha sido adquirido. No caso de uma aquisição de licenças, será necessário indicar ao meio de pagamento que a quantidade da subscrição alterou, sendo que esta alteração automaticamente leva a um aumento de um valor a pagar. Por outro lado, na aquisição de qualquer outro tipo de extras, será necessário adicionar um novo produto à subscrição, aumentando assim o valor final a pagar. Qualquer uma destas alterações tem efeito imediato, o que leva à realização de um pagamento para cobrar a diferença entre o valor já pago sem extras e o valor que deveria ser pago com os extras, para os dias que ainda faltam até à próxima data de pagamento.

A implementação de gestão de extras, revelou-se um verdadeiro desafio, pois envolve verificações sempre que são realizadas operações de upgrade, dado que é necessário remover todos os extras que o utilizador tenha adquirido anteriormente e que façam parte do novo plano.

- **Cancelamento de operação pendente** – Sempre que um utilizador cancela uma operação pendente, é necessário fazer um pedido à API do gateway de pagamento, de forma a cancelar a alteração que foi anteriormente feita. Por exemplo, no caso do cancelamento da subscrição, quando o utilizador indica que deseja cancelar esta operação, é enviado um pedido para reverter o cancelamento que tinha sido feito anteriormente, isto apenas é possível porque a alteração ainda não teve efeito, pois caso a subscrição já tivesse sido efetivamente cancelada, não seria possível reverter.
- **Faturação**

No menu de faturação, o utilizador tem acesso ao formulário de edição dos dados de faturação e também à tabela de pagamentos. Estas funcionalidades apenas envolvem comunicação com a BD, para recolha e atualização de dados (esta última apenas no caso da edição de dados de faturação).

No caso da tabela de pagamentos, onde o utilizador pode visualizar os pagamentos que já realizou e respetivas faturas, foi planeado que o utilizador deve ter a possibilidade de descarregar a fatura relativa a cada pagamento realizado. Apesar de esta funcionalidade ter sido implementada, para que seja possível descarregar o documento, o binário deve estar inserido na BD, na tabela que já existe para o efeito.

A seguinte instrução retorna o ficheiro que será descarregado:

```
Return File(fileBytes, System.Net.Mime.MediaTypeNames.Application.Octet, fileName);
```

- **Histórico**

O histórico é onde o utilizador pode ver a utilização da ferramenta front-end, podendo visualizar entradas e saídas da aplicação ou criação dos vários tipos de registos possíveis (ficheiros, tarefas, despesas, prazos, entidades, assuntos e processos). A informação é mostrada numa tabela, construída com recurso ao plugin datatables, já referido anteriormente, e é possível definir 2 tipos de vistas distintas, uma vista temporal e outra por utilizador. Sempre que alguma das vistas é alterada, a informação mostrada na tabela é atualizada.

- **Afiliados**

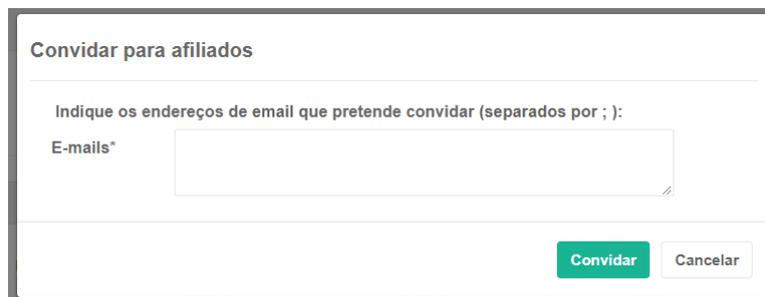


Figura 6.5 – BO Cliente: Caixa de diálogo para convidar possíveis afiliados

A área de gestão de afiliados foi alvo de adição de novos requisitos, pois existiam algumas funcionalidades que não foram planeadas, mas que se revelaram importantes. Contudo, esta é uma área bastante simples, que apenas envolve acessos à base de dados, para recolha ou atualização de dados, e envio de e-mails, para enviar o link de afiliados aos contactos que o utilizador desejar.

O novo requisito de configurar dados para recebimentos, prendeu-se com o facto de a KAMAE sentir a necessidade de saber para onde deve realizar o pagamento de comissões que estão associadas à subscrição.

- **Reiniciar Dados**

Esta funcionalidade é considerada de alto risco, dado que representa uma eliminação total dos dados inseridos na ferramenta de trabalho S4L. De forma a garantir que o utilizador percebe o risco que representa, mas que deseja realmente reiniciar os seus dados, foi adicionada uma validação extra, o envio de um token por e-mail, já anteriormente referido. O token é armazenado na tabela ResetRequest, assim como a respetiva data de expiração.

Do lado do BO cliente, apenas são registados pedidos para fazer reset aos dados, pois não existe acesso à BD onde estão armazenados os registos a apagar. O resto do processo, será realizado por um serviço que não está integrado no planeamento deste estágio, contudo, foi desenvolvido por outro programador da empresa.

Caso a subscrição que vai ser alvo de reset seja portuguesa, é necessário verificar se existem documentos de faturação, pois, caso existam, não é possível reiniciar os dados, por questões legais. A informação referente à existência de documentos de faturação, é recebida a cada login feito no BO Cliente, proveniente do front-end. Por outro lado, também não será possível registar pedidos de reset, caso exista algum pedido pendente, pois, tal como já foi referido, esta operação não é imediata e pode demorar algumas horas até que esteja totalmente concluída.

6.4. Backoffice KAMAE

Tal como o Backoffice de cliente, também este foi implementado em ASP.net e utilizando o Inspinia, pois o desenvolvimento deste componente tinha já sido iniciado com recurso a este template.

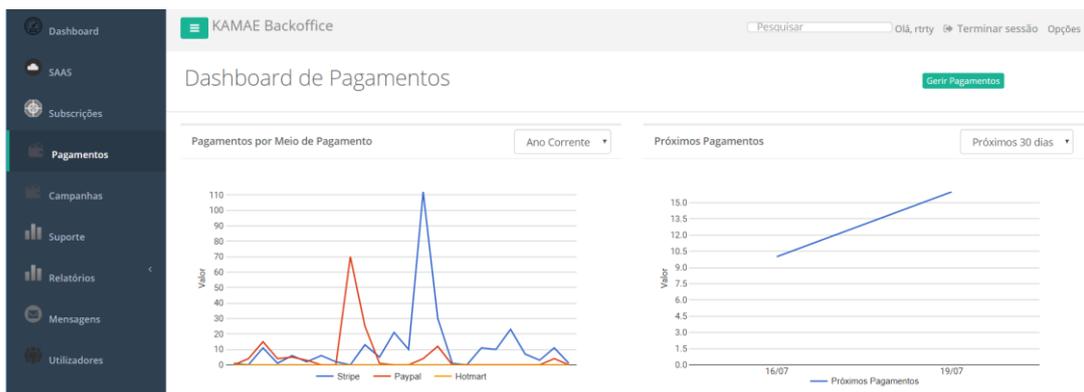


Figura 6.6 - BO KAMAE: Dashboard de Pagamentos

Ao contrário do que aconteceu com o BO Cliente, neste caso nem todos os requisitos planeados foram implementados, ficando a faltar implementar 1 requisito. As alterações que foram feitas e a necessidade de análise aos gateways pagamento, levou a um atraso no início

da implementação deste BO, o que resultou na não implementação do requisito RF19 (Relatório de subscrições).

As funcionalidades implementadas neste BO, incidiram exclusivamente na área de pagamentos, tendo sido desenvolvido um dashboard e uma área para gestão dos tipos de pagamento e das recorrências de pagamento.

- **Dashboard de Pagamentos**

Apesar de já existirem gráficos implementados com recurso a outros plugins, a estagiária optou pela utilização do GoogleCharts, tal como no BO cliente. Esta decisão não foi muito difícil de tomar, pois ao visualizar o código dos gráficos já desenvolvidos, verificou-se que eram utilizados vários plugins distintos, consoante o tipo de gráfico. Deste modo, o carregamento da página torna-se mais lento, pois há mais conteúdo para carregar. Assim, utilizando sempre o mesmo plugin, para além de beneficiar a performance, também é mais simples de entender para outros programadores que possam pegar no projeto futuramente.

Os gráficos disponíveis são:



Figura 6.7 - BO KAMAE: Gráfico de pagamentos por meio de pagamento

Pagamentos por meio de pagamento - demonstra os valores que foram pagos através de cada um dos meios de pagamento disponíveis, durante o período selecionado pelo utilizador. Os vários períodos que o utilizador pode selecionar são “Mês corrente”, “Mês Anterior”, “Ano Corrente” e “Ano Anterior”. É possível visualizar este gráfico na figura 6.7.

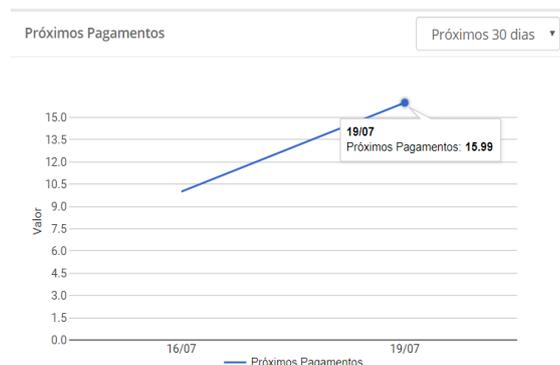


Figura 6.8 - BO KAMAE: Gráfico de próximos pagamentos

Próximos Pagamentos – Demonstra os valores dos próximos pagamentos que irão ser realizados. Sendo que o tempo de visualização também é escolhido pelo utilizador (“Próximos 30 dias”, “Próximos 60 dias” ou “Próximos 90”). Este gráfico pode ser visualizado na figura 6.8.

Carregando em cima de qualquer um destes gráficos, será aberta uma caixa de diálogo com uma tabela para mostrar a informação mais detalhada.

Para além dos gráficos já referidos, neste dashboard foi também implementada uma tabela que mostra informações de pagamentos (data do último pagamento, data do próximo pagamento, valor, entre outras informações relevantes), por subscrição existente.

- **Gestão de Tipos de Pagamento e gestão de recorrências de pagamento**

Estas funcionalidades são bastante semelhantes, estando disponíveis na mesma página, apenas divididas pela existência de separadores. Qualquer uma destas funcionalidades implica apenas mostrar a tabela com os respetivos dados e criar, remover ou editar já existentes, fazendo as respetivas alterações na BD. As tabelas foram criadas com o plugin datatables, que já estava a ser utilizado neste BO, no entanto, foi apenas necessário alterar o carregamento deste plugin para CDN, pois estava a ser feito diretamente apartir de uma pasta do projeto.

6.5. Web API

Foi necessário criar uma Web API, com o objetivo de estar constantemente à espera de receber notificações provindas dos webhooks, que são disponibilizados pelos vários gateways de pagamento para transmitir informações sobre o estado dos pagamentos/subscrições.

De forma a realizar testes essenciais à implementação, foi necessário instalar uma ferramenta, o ngrok, que permite criar túneis seguros em ambientes localhost. Esta necessidade deveu-se ao facto de não ser possível utilizar endereços localhost nos webhooks, e surgiu com o objetivo de verificar quais as informações recebidas em cada tipo de evento, de forma a identificar a subscrição a que cada operação corresponde e qual a alteração ou pagamento que foi realizado.

- **Ngrok** - O ngrok trata-se de uma aplicação consola, que permite a criação de túneis seguros em ambientes localhost. Foi utilizado, no decorrer da implementação, para criar um túnel para a Web API localhost, de forma a ser possível receber notificações provindas dos webhooks dos meios de pagamento.

Para isso, foi utilizado o seguinte comando no ngrok:

```
ngrok http -header-host=localhost [porta]
```

```

ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Account             Daniela Cortesao (Plan: Free)
Version             2.2.8
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://ad092b8f.ngrok.io -> localhost:28966
                    https://ad092b8f.ngrok.io -> localhost:28966

Connections
  ttl    opn    rt1    rt5    p50    p90
   1     0     0.01  0.00  20.61  20.61

HTTP Requests
-----
POST /api/Payment/StripePaymentNotifications 200 OK

```

Figura 6.9 - Ngrok (aplicação consola)

Webhook details	
URL	https://ad092b8f.ngrok.io/api/Payment/StripePaymentNotifications
Mode	TEST

Figura 6.10 - Detalhes do webhook do Stripe

Após executar o comando referido, todos os eventos recebidos são visíveis na consola, assim como o seu estado (bem-sucedido ou com ocorrência de algum problema), tal como é visível na Figura 6.9. Os endereços disponíveis logo à frente de “Forwarding”, representam as possíveis ligações à Web API em localhost (http e https), devendo pelo menos um deles ser indicado na página de configuração do Webhook (figura 6.10).

6.6. Gateways de Pagamento

Apesar de apenas ter sido feita a integração do BO de cliente com o meio de pagamento Stripe, foi necessário realizar uma análise aos restantes (hotmart e paypal), de forma a identificar quais as características e quais as limitações de cada um. Esta análise encontra-se descrita e demonstrada, com recurso a diagramas, no anexo F.

No que diz respeito ao Stripe e o Paypal, foi possível verificar que ambos são bastante flexíveis, permitindo realizar todas as operações desejadas, com recurso a API’s por eles fornecidas.

No entanto, no caso do Hotmart, que, por se tratar de uma plataforma de vendas, como foi já descrito no presente documento, sendo até identificado como uma ferramenta alternativa aos Backoffices a implementar, verificou-se que a integração não seria assim tão simples, pelo menos no período em que este estágio decorreu. Apesar do Hotmart ter em desenvolvimento uma API, o Hotconnect, que, tal como acontece com os restantes gateways, irá permitir alguma flexibilidade ao nível da implementação das funcionalidades desejadas, atualmente ainda não se encontra em ambiente de produção. Esta característica, dificulta a integração com esta plataforma neste momento, pois qualquer operação a realizar (upgrade,

downgrade, cancelamento da subscrição, suspensão) teria que passar sempre por uma ação do utilizador na sua conta Hotmart, retirando autonomia ao BO. Prevê-se que no futuro, com a entrada da API Hotconnect em produção, esta dificuldade será ultrapassada.

Independentemente das limitações de cada um dos gateways de pagamento, todos eles disponibilizam Webhooks, que permitem disparar eventos com informações relevantes relacionadas com as operações realizadas. Sendo que, os Webhooks são uma ferramenta muito útil em qualquer sistema que envolva pagamentos ou gestão de vendas, que é o caso de ambos os Backoffices abrangidos por este estágio. Para receber as notificações provindas dos Webhooks, foi criada uma Web API, que é responsável por realizar as operações necessárias dependendo do tipo de evento que é recebido.

A comunicação com qualquer uma das APIs é feita recorrendo a objetos JSON, assim como as notificações provindas de qualquer um dos Webhooks. Para trabalhar com este tipo de objetos, foi necessário instalar o pacote Newtonsoft, que é uma poderosa framework de JSON para .NET [21], já utilizada em outros componentes do sistema.

O exemplo abaixo demonstra a declaração do método para receber notificações do webhook do Stripe:

```
[AcceptVerbs("GET", "POST")]
[HttpPost]
public string StripePaymentNotifications(HttpRequestMessage request) {...}
```

6.6.1.Stripe

Dos vários eventos que não disparados pelo Webhook do Stripe, houveram alguns que se mostraram mais relevantes para uma correta implementação das funcionalidades planeadas.

- ***invoice.payment_succeeded*** – Notificação responsável por informar a ocorrência de um pagamento bem-sucedido. Neste caso, a Web API faz um pedido à BD para inserir um registo do pagamento realizado.
- ***customer.subscription.updated*** – Notificação responsável por informar a ocorrência de uma atualização a uma subscrição existente.
- ***invoice.payment_failed*** – Notificação responsável por informar que um pagamento falhou. Sempre que é recebida uma notificação deste tipo, é acrescentado um registo de falha de pagamento na BD, associado à subscrição a que corresponde o pagamento falhado.
- ***customer.subscription.deleted*** – Notificação responsável por informar que houve um cancelamento de uma subscrição existente. Foi utilizada apenas para esse efeito, permitindo uma atualização quase instantânea das informações relativas à subscrição na BD. Após recebimento desta notificação, a subscrição envolvida na operação, passa a ficar no estado “Cancelado”, embora este estado esteja temporariamente marcado como pendente, pelas razões já explicadas anteriormente.

Em todas as notificações acima referidas, existem alguns parâmetros que estão sempre presentes nos JSON recebido do webhook, e que são fundamentais para a correta identificação da subscrição a que o evento se refere.

6.6.2. Paypal

A integração com a API do Paypal, passa pela utilização de duas chaves distintas, o id de cliente e uma chave secreta e é feita do seguinte modo:

```
var accessToken = new  
OAuthTokenCredential(ClientID,ClientSecret).GetAccessToken();  
APIContext context = new APIContext(accessToken);
```

Apesar de não ter sido feita a integração com o paypal, foi possível identificar já alguns eventos disparados pelo webhook que permitem identificar as operações a realizar no BO.

- **PAYMENT.SALE.COMPLETED** – Notificação de um pagamento bem-sucedido.
- **PAYMENT.SALE.DENIED** – Notificação de um pagamento falhado.
- **BILLING.SUBSCRIPTION.DELETED** – Notificação de uma subscrição cancelada.

Na secção de anexos, será possível observar a análise feita a este meio de pagamento.

6.6.3. Hotmart

Numa análise ao Hotmart, foi possível concluir que, atualmente, uma implementação completa com este meio de pagamento não é possível, pois a API por eles fornecida não permite realizar as operações pretendidas sem necessidade de interação do utilizador com a página Hotmart. No entanto, existe uma nova API em desenvolvimento, que eventualmente, no futuro, poderá vir a possibilitar esta integração.

Apesar de terem sido feitos alguns testes com a ferramenta de webhooks disponibilizada, a informação obtida não foi muito conclusiva, sendo necessário realizar testes mais específicos quando realmente for possível fazer integração com este meio de pagamento.

Capítulo 7

Plano de Riscos

Um risco é um acontecimento ou fator indesejado e não planejado que pode levar ao insucesso do produto, impedindo que sejam cumpridos todos os objetivos definidos no início do planejamento do projeto em causa. É importante identificar desde cedo os riscos, de forma a encontrar estratégias de mitigação para que, mais tarde, estes não se tornem em problemas reais.

Deste modo, nesta secção será feita uma apresentação e análise dos possíveis riscos que podem ter influência no desenvolvimento do produto. Será ainda definido um plano necessário para mitigar cada um dos riscos identificados.

7.1. Caracterização do risco

De forma a classificar os riscos identificados, será atribuído a cada um dos riscos uma classificação para cada um dos atributos: Impacto, probabilidade e time frame. Esta classificação permitirá identificar quais os riscos mais prioritários, para que se possa agir mais rapidamente no sentido de os mitigar.

Impacto – De modo a classificar o impacto que cada risco poderá ter, caso se tornasse num problema real, será utilizada a seguinte escala:

- **Catastrófico** – Um risco com impacto de nível catastrófico terá como consequência a falha do projeto, pois não será possível alcançar os critérios de sucesso definidos;
- **Crítico** – Um risco com impacto de nível crítico, terá grande influência no sucesso do projeto, é possível conseguir alcançar os critérios de sucesso, mas o esforço/custo será bastante superior ao expectável;
- **Marginal** – Um risco com impacto de nível marginal poderá influenciar os critérios de sucesso, mas não de forma relevante, não merecendo grande atenção.

Probabilidade – De modo a classificar a probabilidade de um risco acontecer, será utilizada a seguinte escala:

- **Alta** – Risco com probabilidade de acontecer superior a 70%;
- **Média** – Risco com probabilidade de acontecer entre 40% e 70%;
- **Baixa** – Risco com probabilidade de acontecer inferior a 40%.

Time Frame – De modo a classificar o tempo em que um fator pode ser considerado um risco, será utilizada a seguinte escala:

- **Longo** – Alguns meses (mais do que 3)
- **Médio** – Entre 1 a 3 meses
- **Curto** – Menos de 1 mês (algumas semanas)

7.2. Levantamento de Riscos

R01 - Especificação inadequada de requisitos.

Descrição – Uma especificação inadequada de requisitos pode levar a uma interpretação incorreta do que é necessário implementar, ou até a falta de requisitos importantes ao bom funcionamento do produto, podendo vir a resultar num atraso no desenvolvimento do projeto.

Plano de Mitigação – Todos os requisitos especificados devem passar por uma revisão por parte da empresa, incluindo os possíveis futuros pedidos de alteração/adição de requisitos.

Caracterização – Impacto: Crítico; Probabilidade: Média; Time Frame: Médio.

R02 - Pedido de novos requisitos por parte da empresa.

Descrição – Ao longo do desenvolvimento do projeto, pode haver pedido de novos requisitos por parte da empresa. Apesar do pedido de novo requisitos não ser algo muito preocupante, pois, tendo em conta que está a ser utilizada uma metodologia ágil, estas alterações são perfeitamente aceitáveis, pode levar a atraso no desenvolvimento.

Plano de Mitigação – Ao longo das reuniões diárias de sprint, a estagiária deve passar informação clara em relação aos requisitos em que está a trabalhar e o que conseguiu realizar ao longo dia, de forma a que seja mais fácil para a empresa perceber em que estado está o projeto e avaliar se é aceitável ou não adicionar novos requisitos.

Caracterização – Impacto: Crítico; Probabilidade: Média; Time Frame: Longo.

R03 - Fraca documentação em relação ao que está implementado no backoffice da KAMAE.

Descrição – Não existe qualquer documentação relativamente ao código já implementado no backoffice da KAMAE, nem à estrutura do projeto. Existem apenas alguns Work Items, com descrições de requisitos que foram implementados neste componente, que apesar de puderem vir a ser úteis no entendimento das funcionalidades já implementadas, nada ajudam na compreensão do código já produzido. A inexistência de documentação relativamente a aspetos de codificação, pode levar a uma dificuldade acrescida de integração com este componente por parte da estagiária, podendo vir a causar atrasos no desenvolvimento do produto.

Plano de Mitigação – Sempre que existirem dúvidas relativamente à estrutura do código ou a determinadas funcionalidades é necessário pedir imediatamente informações aos elementos da equipa que estiveram anteriormente envolvidos no componente em questão.

Caracterização – Impacto: Marginal; Probabilidade: Alta; Time Frame: Longo.

R04 – Estimativas erradas no planeamento do 2º semestre.

Descrição – A falta de experiência por parte da estagiária na realização de estimativas, poderá resultar na realização de estimativas bastante irreais relativamente a uma ou várias tarefas, podendo levar ao atraso na realização das tarefas em causa.

Plano de Mitigação – A realização de estimativas deve ser feita pela estagiária, com apoio da restante equipa de desenvolvimento, no início de cada sprint, com base no backlog previamente definido. O Gráfico Gantt deve estar sempre atualizado, de forma a tornar claro o estado do trabalho desenvolvido.

Caracterização – Impacto: Catastrófico; Probabilidade: Média; Time Frame: Médio.

R05 – Pouco conhecimento da framework Bootstrap

Descrição – A estagiária possui um nível de conhecimento reduzido de Bootstrap, pois utilizou pouco esta framework, o que poderá levar a que seja necessário um esforço acrescido para aquisição de mais conhecimento.

Plano de Mitigação – Realização de uma pequena prova de conceitos, com objetivo de adquirir algum conhecimento relativamente à framework bootstrap.

Caracterização – Impacto: Marginal; Probabilidade: Média; Time Frame: Curto.

7.3. Matriz de Exposição de Riscos

Após realização do levantamento de riscos, e classificação de cada um dos atributos anteriormente referidos para cada risco (impacto, probabilidade e time frame), será agora apresentada a matriz de exposição de riscos (figura 7.1). Esta matriz permite-nos ter uma visão geral da prioridade dos riscos, tendo em conta a classificação dos atributos.

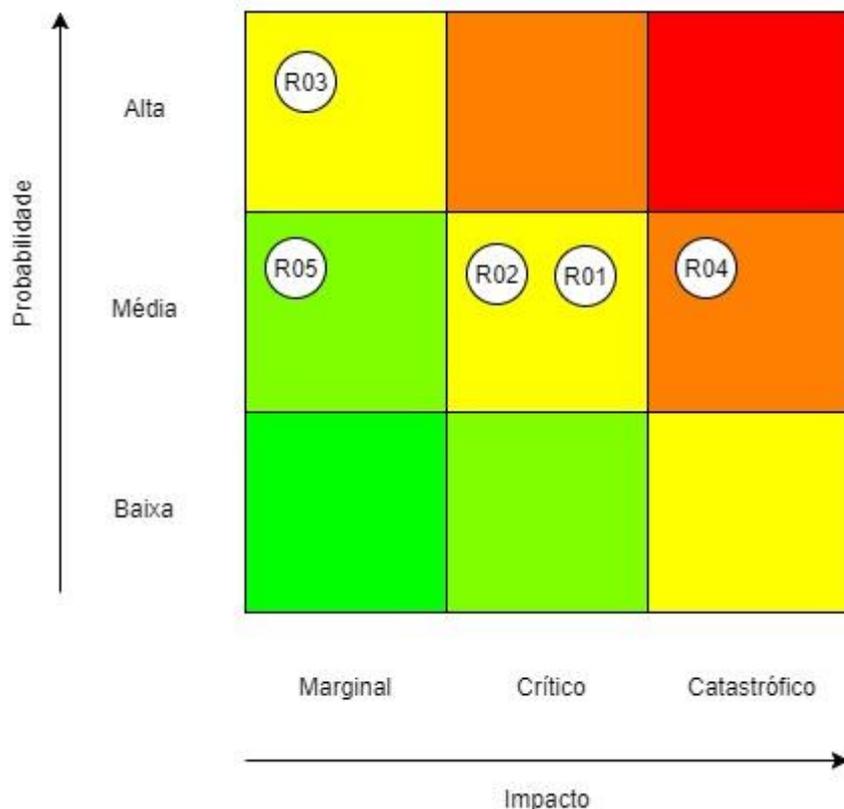


Figura 7.1 - Matriz de Exposição de riscos

Capítulo 8

Protótipos

No decorrer da primeira fase do estágio, foi pedido à estagiária que realizasse um protótipo do backoffice de cliente, de forma a analisar e avaliar se a interface é adequada. De forma a demonstrar o trabalho realizado, nesta secção serão apresentados alguns dos ecrãs produzidos na realização do protótipo. Os restantes ecrãs (secundários ou com interface semelhante aos aqui apresentados) estarão disponíveis para consulta, na secção de Apêndice E, nos anexos do presente documento.

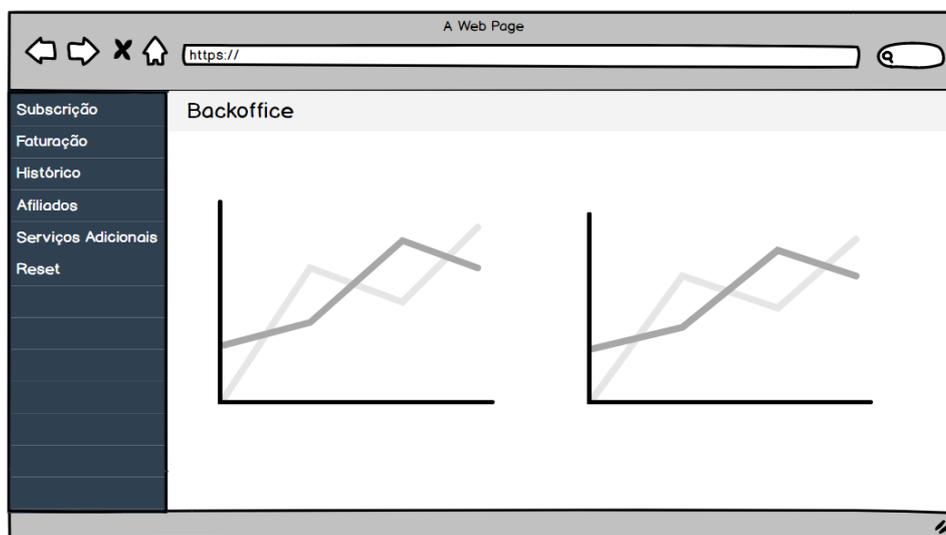


Figura 8.1 - Página inicial do backoffice

A figura 8.1, representa a página inicial do backoffice, visualizada por qualquer utilizador com uma subscrição ativa que aceda a este componente do sistema. Mais do que simplesmente uma página inicial, é aqui que o utilizador pode ver os gráficos relativos ao armazenamento ocupado, por todos os dados inseridos no seu site, e o espaço que tem ainda disponível.

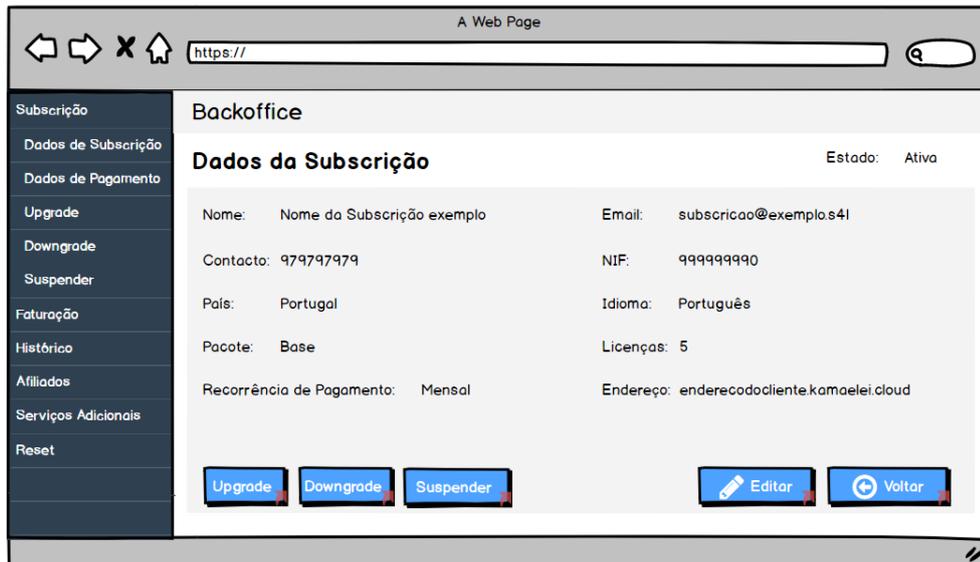


Figura 8.2 - Página de consulta dos dados de subscrição

A figura 8.2 representa a página de consulta dos dados da subscrição, acessível através do sub-menu “Dados de Subscrição”, existente dentro da opção “Subscrição” do menu principal.

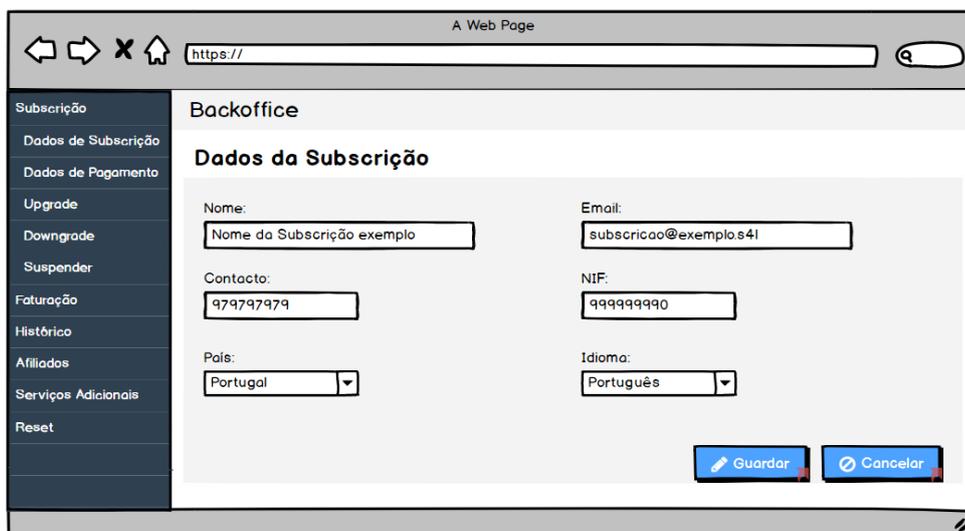
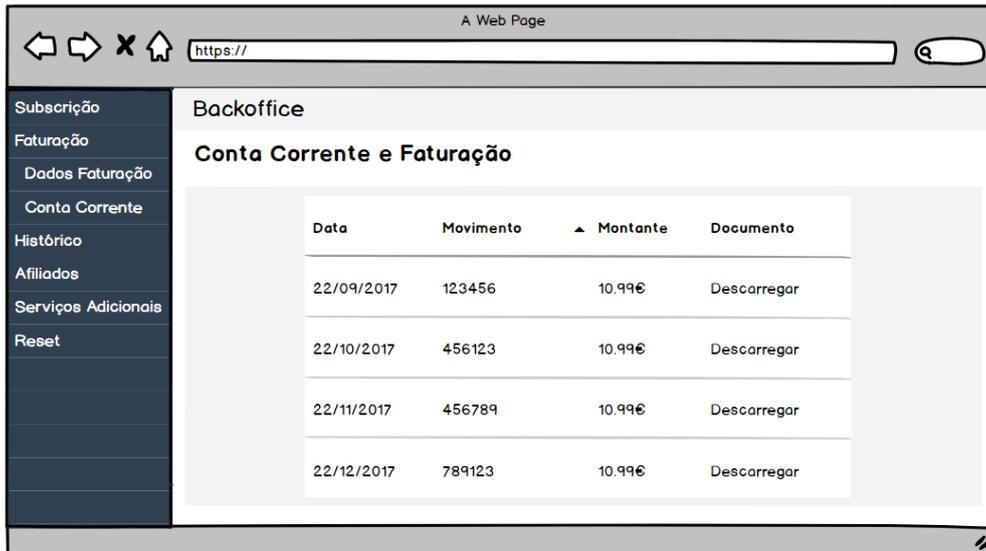


Figura 8.3 - Formulário de edição dos dados de subscrição

A figura 8.3, representa o formulário de edição dos dados de subscrição, onde o utilizador pode alterar as informações que desejar relativamente à sua subscrição. Nos anexos será possível encontrar figuras semelhantes a esta que demonstram os 2 possíveis comportamentos da aplicação quando se carrega no botão “Guardar” (sucesso ou dados inválidos).



Data	Movimento	Montante	Documento
22/09/2017	123456	10,99€	Descarregar
22/10/2017	456123	10,99€	Descarregar
22/11/2017	456789	10,99€	Descarregar
22/12/2017	789123	10,99€	Descarregar

Figura 8.4 - Página de consulta de conta corrente e faturação

A figura 8.4, representa a página de consulta de conta corrente e faturação, sendo também nesta página que o utilizador pode descarregar as suas faturas, através do botão “descarregar”, presente na tabela.

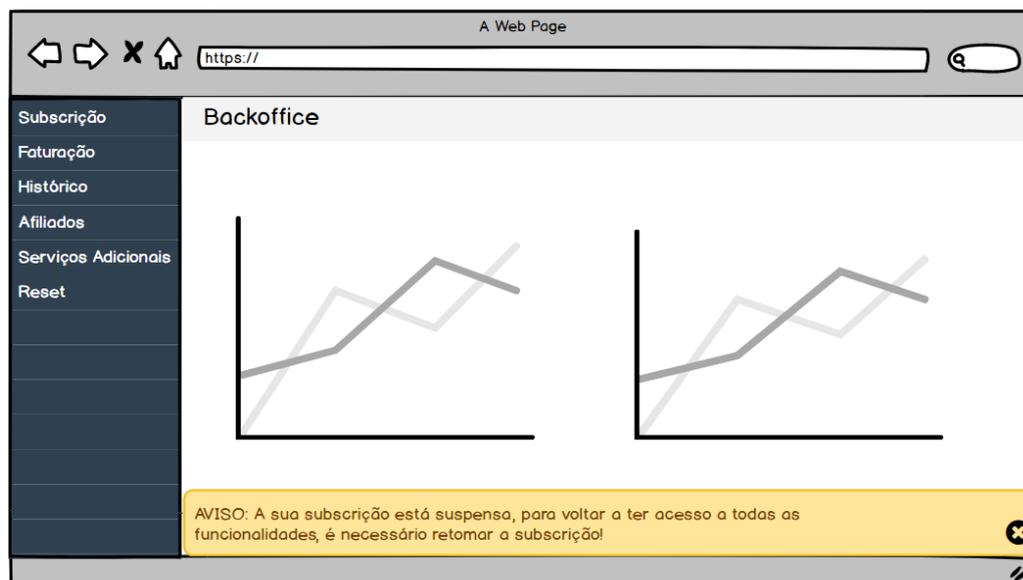


Figura 8.5 - Página inicial do backoffice de subscrição suspensa

A figura 8.5 representa a página inicial do backoffice quando a subscrição do utilizador se encontra suspensa. Tal como é possível visualizar, o número de funcionalidades disponíveis é reduzido e o utilizador terá que reativar a sua subscrição para voltar a ter acesso à totalidade das funcionalidades.

Capítulo 9

Conclusão e Trabalho Futuro

9.1. Reflexão

Este projeto, que se dividiu em duas fases, teve como principal objetivo o desenvolvimento e implementação de dois componentes considerados vitais ao correto funcionamento da solução S4L da KAMAE, o backoffice de cliente e o backoffice KAMAE.

O objetivo da primeira fase do estágio, passou pelo planeamento dos componentes a implementar, identificando e detalhando todos os requisitos a desenvolver, assim como a arquitetura que compõe o sistema. Contudo, no início da 2ª fase, que correspondeu à fase de implementação dos componentes propostos, o planeamento já feito sofreu algumas alterações, não tanto ao nível de requisitos, mas mais ao nível arquitetural, que obrigaram a um esforço extra para planear alguns aspetos que não tinham sido ainda considerados.

Considera-se que as alterações que o projeto sofreu no início da 2ª fase do estágio, acabaram por se revelar vantajosas, pois apesar do atraso causado no projeto, resultaram na aquisição de novos conhecimentos, dado que foi necessário trabalhar com ferramentas que inicialmente não estavam previstas. Um exemplo da aprendizagem adquirida com estas alterações é que, a estagiária desconhecia totalmente a existência do método webhook, que, com a integração dos meios de pagamento, se revelou imprescindível. Deste modo, foi necessário entender o que é o webhook e o seu modo de funcionamento.

Tal como já foi possível verificar, pois já foi referida a não implementação de um dos requisitos planeados, nem tudo correu como esperado. As várias fases de testes que foram planeadas e consequentes correções de bugs não foram realizadas. A falta de disponibilidade por parte da pessoa responsável pelos testes dentro da empresa, impediu a realização destas fases. Por outro lado, a necessidade de correção de bugs, também poderia vir a resultar num atraso maior da implementação por parte da estagiária.

Apesar dos desafios encontrados, considera-se que o estágio correu bem e que o resultado final vai de encontro às expectativas da empresa que recebeu a estagiária para realizar este trabalho.

Para além de conhecimentos ao nível técnico, este estágio permitiu também a aquisição de nível de gestão de projeto, especialmente no que diz respeito à metodologia de gestão SCRUM, pois foi a metodologia utilizada ao longo de todo o estágio. O ambiente empresarial é sempre um pouco diferente do ambiente académico e este contato com uma empresa é bastante vantajoso para a aquisição de conhecimentos, sobretudo no que diz respeito ao modo como o trabalho é realizado e vai evoluindo dentro de uma empresa.

9.2. Trabalho futuro

Há ainda algum trabalho a desenvolver, considerando todos os meios de pagamento e cenários que a KAMAÉ pretende abranger nos componentes desenvolvidos. Contudo, o trabalho realizado ao longo deste estágio, após realização dos testes necessários e imprescindíveis, já abrange o que a empresa pretende para uma primeira fase de entrada em produção, pelo menos o BO cliente.

No que diz respeito ao BO cliente, no decorrer do estágio, a empresa manifestou o desejo de ter mais algumas informações disponíveis na área de gestão dos dados de subscrição. O objetivo será colocar separadores, de modo a agrupar os vários tipos de informação. Sendo que, algo que a empresa deseja ter nesta área, é uma tabela com todos os utilizadores que existem associados à subscrição.

Por outro lado, no que diz respeito ao BO KAMAÉ, para além das novas funcionalidades que possam ser ainda necessárias implementar, foi possível concluir que existem algumas correções a realizar que podem resultar em melhorias consideráveis à interação com este componente.

O S4L é um projeto de grandes dimensões, que estará sempre em constante evolução. Por este motivo, os componentes implementados neste estágio, estarão sempre sujeitos a novas necessidades de implementação e alterações, de modo a cobrir todas as necessidades da empresa e dos seus clientes.

Referências

- [1] “Clio,” [Online]. Available: <https://www.clio.com>. [Acedido em 15 10 2017].
- [2] “Kamae,” [Online]. Available: <http://www.kamae.pt/>. [Acedido em 15 10 2017].
- [3] “Solutions4Lawyers,” [Online]. Available: <https://www.solutions4lawyers.com>. [Acedido em 17 10 2017].
- [4] “App4Legal,” [Online]. Available: <https://app4legal.com/en>. [Acedido em 21 10 2017].
- [5] “Infosysta,” [Online]. Available: <http://www.infosysta.com>. [Acedido em 21 10 2017].
- [6] “Appfolio,” [Online]. Available: <https://www.appfolioinc.com>. [Acedido em 21 10 2017].
- [7] “MyCase,” [Online]. Available: <https://www.mycase.com/>. [Acedido em 21 10 2017].
- [8] “Best Law Practice Management Software,” [Online]. Available: <https://www.capterra.com/law-practice-management-software/>. [Acedido em 22 10 2017].
- [9] “Forense - Sistema de Gestão de Escritórios para Advogados,” [Online]. Available: <http://www.forense.pt/site/>. [Acedido em 20 10 2017].
- [10] “Asana,” [Online]. Available: <https://app.asana.com>. [Acedido em 20 10 2017].
- [11] “Instagantt,” [Online]. Available: <https://instagantt.com>. [Acedido em 21 10 2017].
- [12] “Scrum-process1.png,” [Online]. Available: <https://i2.wp.com/www.mindmaster.com.br/wp-content/uploads/2014/06/Scrum-Process1.png?resize=630%2C315>. [Acedido em 20 11 2017].
- [13] “What is Scrum?,” [Online]. Available: <https://www.scrum.org/>. [Acedido em 20 11 2017].
- [14] “Where work happens | Slack,” [Online]. Available: <https://slack.com/>. [Acedido em 18 09 2017].
- [15] “ASP.NET Overview | Microsoft Docs,” [Online]. Available: <https://docs.microsoft.com/pt-pt/aspnet/overview>. [Acedido em 19 09 2017].
- [16] “Bootstrap . The most popular HTML, CSS and JS library in the world.,” [Online]. Available: <https://getbootstrap.com/>. [Acedido em 04 10 2017].
- [17] “Visual Studio IDE, Editor de Código e Mobile Center,” [Online]. Available: <https://www.visualstudio.com/>. [Acedido em 18 09 2017].
- [18] “SQL Server 2016 | Microsoft,” [Online]. Available: <https://www.microsoft.com>. [Acedido em 27 11 2017].

- [19] “Microsoft previous versions of technical documentation | Microsoft Docs,” [Online]. Available: [https://msdn.microsoft.com/en-us/library/ms242904\(VS.80\).aspx](https://msdn.microsoft.com/en-us/library/ms242904(VS.80).aspx). [Acedido em 15 04 2018].
- [20] “ngrok - secure introspectable tunnels to localhost,” [Online]. Available: <https://ngrok.com/product>. [Acedido em 25 03 2018].
- [21] “Json.NET - Newtonsoft,” [Online]. Available: <https://www.newtonsoft.com/json>. [Acedido em 25 03 2018].

Apêndices

Apêndice A – Estado da Arte

Considerando que na 2ª fase do estágio foi necessário realizar uma reformulação ao estado da arte escrito ao longo do primeiro semestre, este apêndice vem como um complemento ao 2º capítulo do presente documento. Apesar da necessidade de reformulação, considerou-se que a análise anteriormente feita era importante para a compreensão da solução onde este estágio se integra, por esse motivo, é aqui apresentada uma análise mais aprofundada ao S4L e a algumas soluções semelhantes.

1. S4L

Existem várias funcionalidades no S4L que merecem destaque, sendo elas lançamento de tarefas, prazos e despesas, faturação certificada, com vários tipos de documentos disponíveis, contabilização de tempos, através de cronómetros disponíveis no site e que no final são lançados como tarefas. Todos os tipos de lançamentos que podem ser feitos ficam obrigatoriamente associados a um contexto, que consiste numa hierarquia de cliente, assunto e processo. O cliente corresponde à entidade (coletiva ou singular) que recorre aos serviços do advogado/sociedade, um cliente pode estar associado a vários assuntos, mas este apenas pode ter um cliente principal. O termo “assunto” surge como substituição à pasta física, frequentemente utilizada por advogados para armazenar informações relativas ao cliente e processos desse cliente. Por este motivo, para poder fazer lançamentos (por exemplo associar tarefas) associados a um determinado cliente, tem sempre que existir pelo menos um assunto associado ao cliente em questão, dado que qualquer lançamento tem que ficar associado a um assunto e, conseqüentemente, a um cliente. Por fim, o assunto pode estar associado a vários processos, contudo um processo apenas pode ter um assunto. O processo pode ser judicial ou extrajudicial, mas não é um registo obrigatório. É possível identificar partes contrárias, tanto ao nível de assunto, como de processo, adicionando todas as entidades necessárias como contactos.

Uma outra funcionalidade oferecida neste sistema, e que é considerada fundamental, é a possibilidade de extrair relatórios de vários tipos, sendo até possível os utilizadores personalizarem os seus próprios relatórios ou relatórios de equipa e guardarem esta personalização (desde que tenham permissões para o fazer), de forma a obterem a informação que considerem mais relevante sempre que necessário.

Sendo que os advogados trabalham mediante prazos estabelecidos, ao longo de toda a utilização do sistema, estarão sempre visíveis alertas com a informação sobre os prazos do utilizador autenticado.

Figura A.1 - Hierarquia de Contexto no S4L.

2. Outras Soluções existentes

Existem inúmeras soluções de gestão de trabalho para advogados [8], contudo, de forma a analisar outras soluções existentes, foram selecionadas 4 soluções que se mostraram mais relevantes, dado que são semelhantes ao que se pretende com o S4L e, no caso do sistema Forense, por se tratar de um concorrente direto. Após selecionar os 4 produtos a analisar, foi criada uma conta trial em cada uma das soluções, para observar as funcionalidades que são oferecidas e as principais diferenças entre as várias plataformas.

- **Clio**

A Clio [1] é uma empresa criada em 2008 por dois amigos, Jack Newton e Rian Gauvreau, com o objetivo de desenvolver e comercializar uma solução com o mesmo nome. É uma solução SaaS, com necessidade de subscrição, sendo possível subscrever um plano anual ou um plano mensal. Pelo que foi possível apurar, está disponível apenas em inglês.

As funcionalidades que esta plataforma oferece, são bastante semelhantes às já referidas no S4L, contudo, o modo de funcionamento é um pouco distinto. Dispõe de funcionalidades para registo e controlo de tempos, registo de trabalho, faturação e permite ainda gerar vários tipos de relatórios.

O contexto do sistema Clio incide sobre registos do tipo contacto (semelhante aos clientes no S4L) e registos do tipo assunto, não existe o tipo de registo processo, dado que todo o trabalho incide sobre o assunto. No entanto, existe alguns tipos de registos que não necessitam de estar associados a um contexto, por exemplo as tarefas. O facto de ser possível lançar alguns tipos de registos sem ligação ao contexto, poderá ser uma desvantagem, pois poderá causar dificuldades na faturação do trabalho realizado ao cliente correspondente.

Funcionalidades como introdução de intervenientes da parte contrária, criação de honorários ou criação de avenças não foram encontradas nesta plataforma. Ao nível da faturação, não foi encontrada uma grande diversidade de documentos de faturação, apenas sendo possível criar uma fatura simples com os dados de pagamento.

Foi possível constatar que, à exceção do calendário, que não está disponível em todo o site, não existe em qualquer local informação à cerca dos prazos do utilizador. Dado que os advogados trabalham muito com base em prazos, seria importante que de alguma forma fosse possível visualiza-los em qualquer página da aplicação.

O lançamento de tempos através de cronómetro, como funcionalidade imprescindível que é para a área jurídica, está também presente nesta plataforma.

- **App4Legal**

A App4Legal [4] é uma solução criada pela empresa Infosysta [5]. A Infosysta é uma empresa de consultadoria e desenvolvimento de software, responsável pelo desenvolvimento de várias soluções de gestão de trabalho.

Esta solução é SaaS, oferecendo a possibilidade de subscrever um plano mensal, mas está também disponível como não SaaS, sendo neste caso instalada num servidor à responsabilidade do cliente. Está disponível em 4 idiomas (inglês, árabe, espanhol e francês) e oferece funcionalidades para registo e controlo de tempos, registo de trabalho e faturação.

A hierarquia de contexto, tal como acontece com o sistema Clio, incide sobre contactos e assuntos, contudo, é possível criar processos litigiosos, mas estes não estão relacionados com a hierarquia referida.

A criação de relatórios funciona com base em configurações efetuadas no momento de gerar o relatório, não sendo possível guardar a configuração. O modo de funcionamento desta funcionalidade talvez não seja o mais adequado, pois os utilizadores são obrigados a realizar uma nova configuração sempre que necessitam de gerar um relatório, acabando por perder mais tempo. Por outro lado, o tipo de relatórios que é possível gerar é bastante limitado e um pouco confuso.

Nesta plataforma, foi possível verificar que existe um sistema de notificações, bastante útil para alertar os utilizadores dos prazos com data de término mais próxima.

- **MyCase**

MyCase [6] é o nome de um sistema de gestão de práticas legais, criado pela empresa AppFolio [7]. Esta empresa foi fundada em 2006 por Klaus Schauer e Jon Walker, com a missão de ajudar pequenas e médias empresas a crescer e competir no mercado, fornecendo o software e serviços de gestão.

A solução MyCase está disponível apenas no idioma inglês e, tal como todas as anteriormente referidas, é SaaS, oferecendo um plano de subscrição mensal. Oferece funcionalidades para registo e controlo de tempos, registo de trabalho, faturação e gerar vários tipos de relatórios que possam interessar ao cliente. Esta ferramenta dá também a possibilidade de criar utilizadores do tipo cliente, associando estes utilizadores a contactos existente, para que estes possam assim consultar dados relativos aos seus processos, com as devidas limitações.

Em relação ao contexto, na aplicação MyCase não existem registos de assuntos, existindo apenas o processo que tem que estar associado a um contacto (semelhante ao cliente do S4L). Foi possível contactar que todos os tipos de registos (tarefas, despesas, entre outros) têm que estar associados a um contexto, para que, mais tarde, todo o trabalho possa ser facilmente faturado.

A funcionalidade de relatórios é bastante completa, tendo disponíveis vários tipos, no entanto, tal como acontece com a App4Legal, não é possível gravar relatórios personalizados (por exemplo relatório mensal de tarefas), forçando o utilizador a estar sujeito ao que a plataforma oferece, configurando apenas as datas desejadas.

Esta solução oferece uma funcionalidade que merece destaque, pois pode revelar-se bastante útil no ramo da advocacia, os workflows (Figura 2.6). Esta funcionalidade é descrita como uma ferramenta para poupar tempo e aumentar a eficiência diária do

utilizador, permitindo criar modelos com grupos de tarefas e/ou eventos e aplicar facilmente estes modelos a processos específicos, passando todos os registos do modelo a ficar associados ao processo.

- **Forense**

Forense [9] é uma ferramenta de gestão na área jurídica, desenvolvida pela empresa Gravidade Zero. Este produto assemelha-se ao S4L, pois também existe uma aplicação desktop, o Forense, que está agora a evoluir para uma plataforma web, o Forense Web. Contudo, apesar desta plataforma se ter mostrado bastante boa ao nível de performance, está ainda numa versão um pouco instável, pois em apenas alguns testes foram detetados vários problemas funcionais. Esta disponível nos idiomas português, brasileiro, francês, espanhol e inglês.

A hierarquia de contexto presente neste sistema, incide sobre registos do tipo cliente e do tipo processo, existindo também a possibilidade de os clientes terem utilizadores associados, para poderem aceder à plataforma, e identificação das partes contrárias dos processos.

Apesar de ser possível configurar vários tipos de relatórios, não é possível guardar configurações, para nova consulta posteriormente, obrigando o utilizador a fazer uma nova configuração sempre que pretende consultar um relatório.

Não foi possível encontrar registos do tipo honorários, nem funcionalidades de faturação, à exceção de criação de avenças, talvez por ser uma ferramenta ainda bastante recente.

- **Comparação das soluções existentes**

De forma a reforçar a análise feita às soluções existentes, foram feitos alguns testes às várias plataformas front-end, de forma a identificar as funcionalidades que cada uma oferece.

	S4L	Clio	App4Legal	MyCase	Forense
SaaS	✓	✓	✓	✓	✓
Clientes	✓	✗	✗	✗	✓
Assuntos	✓	✓	✓	✗	✗
Processos	✓	✗	✓	✓	✓
Contactos	✓	✓	✓	✓	✓
Partes contrárias	✓	✗	✓	✗	✓
Faturação	✓	✓	✓	✓	✗
Workflows	✗	✗	✗	✓	✗
Cronómetros	✓	✓	✓	✓	✓
Clientes/contactos podem ter acesso	✓	✗	✗	✓	✓
Relatórios personalizáveis	✓	✓	✗	✓	✓
Lançamento de Tarefas e Despesas	✓	✓	✓	✓	✓
Avenças	✓	✗	✗	✗	✓

Honorários	✓	✗	✗	✗	✗
Calendário	✓	✓	✓	✓	✓
Sistema de notificações	✓	✗	✓	✗	✗

Tabela A.1 - Comparação de Soluções existentes

Na tabela A.1 é possível observar quais as principais diferenças, ao nível de funcionalidades, entre as várias soluções. É visível que o front-end do S4L foi criado com o objetivo de ser uma plataforma o mais completa possível, para responder a todas as necessidades dos seus utilizadores, pois oferece várias funcionalidades que não foram encontradas em nenhuma das restantes ferramentas testadas.

A falta de funcionalidades de criação/gestão de clientes, nas plataformas testadas, não é relevante, pois ao contrário do que acontece no S4L, este tipo de entidade é substituída pela entidade de contactos.

Em todas as aplicações testadas existe uma hierarquia de contexto, no entanto difere entre todas. A plataforma cujo contexto é mais semelhante com o da S4L, é a App4Legal, pois também é possível criar contactos (no S4L são clientes), assuntos e processos, assumindo a mesma ordem. Apesar de todas as aplicações trabalharem com base no contexto, em algumas das soluções concorrentes testadas, é possível lançar alguns tipos de registos sem qualquer ligação a um contexto (por exemplo tarefas) ao contrário do que acontece no S4L.

É importante referir que, ao nível de usabilidade, a plataforma App4Legal, mostrou-se um pouco inferior às restantes soluções, pois por vezes houve dificuldade em perceber como executar algumas funcionalidades, por exemplo adicionar um contacto.

Apêndice B – Plano de trabalhos do 1º semestre

Nesta subsecção será apresentado um gráfico Gantt que representa o plano de trabalhos para o 1º semestre. Será ainda possível visualizar um segundo gráfico Gantt, que irá demonstrar o desvio que ocorreu na realização de cada tarefa em relação ao plano inicial. Ambos os gráficos foram produzidos com o apoio da ferramenta online Instagantt.

De forma a facilitar a visualização da distribuição de tarefas, os gráficos foram produzidos utilizando uma organização em semanas, sendo que, cada dia representado, corresponde ao primeiro dia da respetiva semana.

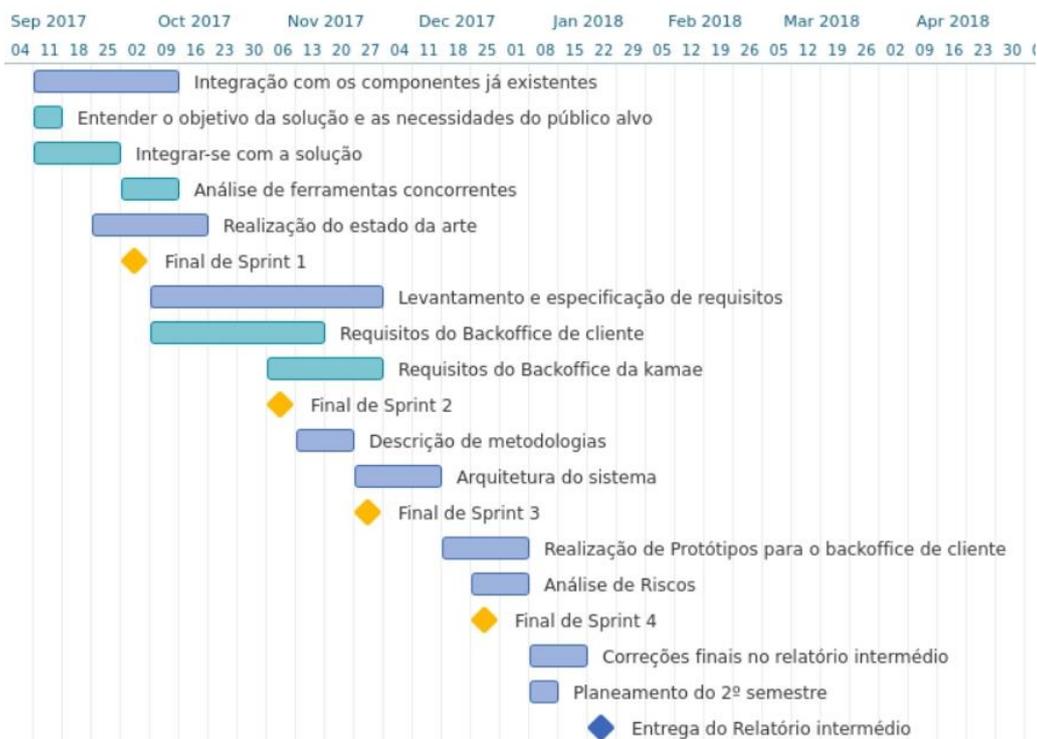


Figura B.1 - Plano Inicial do 1º semestre

Na figura B.1 é possível visualizar o plano inicial da primeira fase do estágio, correspondente ao primeiro semestre. Nesta fase, a distribuição de tarefas passou por organizar o trabalho ao longo de 20 semanas em 16 horas semanais, correspondendo a 2 dias de trabalho por semana (8 horas por dia) e resultando num total de 320 horas.

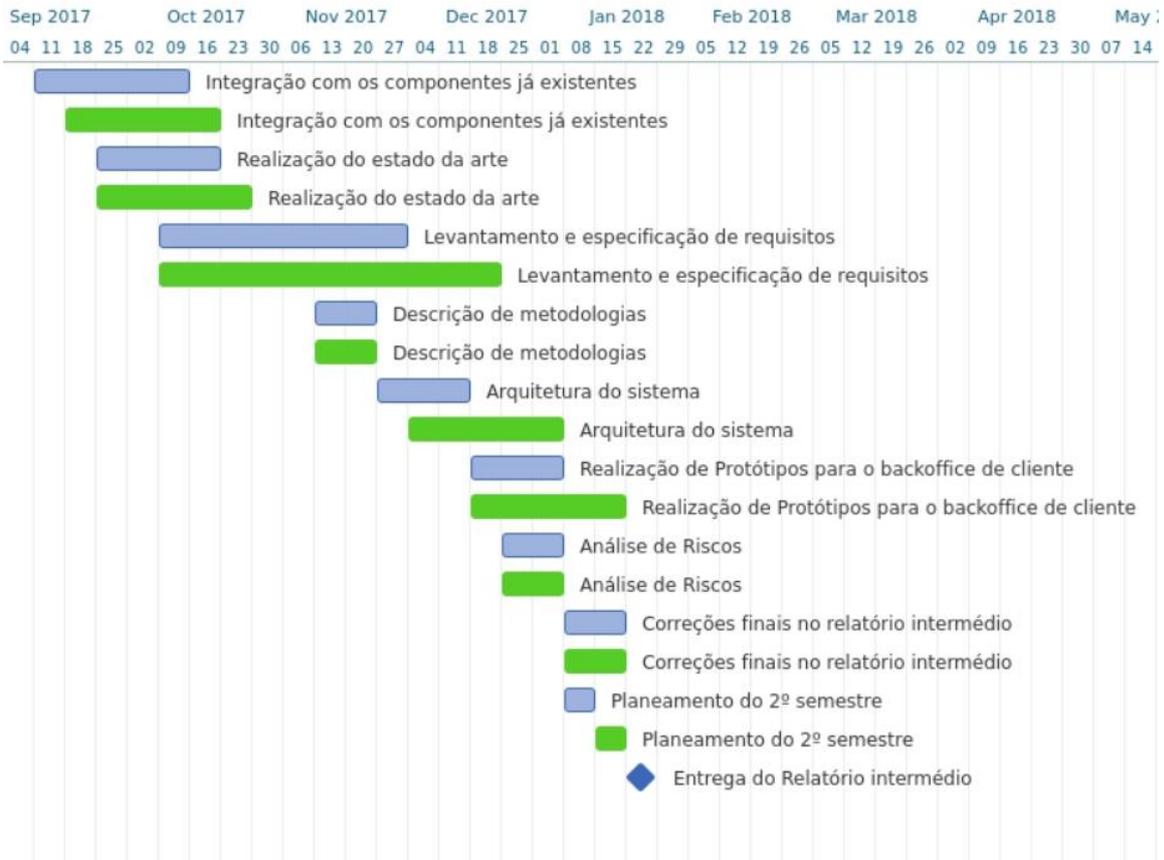


Figura B.2 - Desvio do plano inicial

Na figura B.2, é possível visualizar o desvio do tempo real despendido na realização das tarefas, em relação ao plano inicial. Verifica-se que houve pequenos atrasos na realização de algumas tarefas, no entanto foi possível concluir todo o trabalho planeado inicialmente.

No início do estágio, a aluna passou por pequenos desafios relacionados com a sua inscrição na unidade curricular, o que levou a um atraso inicial na realização das tarefas, que foi recuperado com um esforço acrescido por parte da estagiária. Também nas fases de especificação de requisitos e de arquitetura do sistema, se verificaram alguns atrasos, que são justificados, principalmente, pela necessidade de melhorias e correções no conteúdo produzido ao longo destas fases. Por fim, o atraso na realização de protótipos e no planeamento do 2º semestre veio como consequência dos desvios anteriormente referidos.

Apêndice C – Especificação de requisitos

Este apêndice completa o capítulo 4, apresentando uma análise mais detalhada aos requisitos funcionais planejados.

a) Backoffice de Cliente

Nome	Gerir Subscrição
ID	UC01
Descrição	O utilizador pretende gerir a sua subscrição, podendo realizar várias operações.
Utilizadores	Utilizador autenticado.
Pré-Condições	1. O utilizador tem uma subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador inicia sessão na aplicação; 2. O utilizador carrega no menu de utilizador existente no canto superior direito; 3. O utilizador carrega na opção “Gerir Subscrição”; 4. A aplicação redireciona o utilizador para o site de backoffice de cliente.
Fluxo Alternativo	<p>2a: O utilizador iniciou sessão com um email diferente do email de subscrição</p> <p>2a1: Não é possível visualizar a opção “Gerir Subscrição”</p>
Pós-Condições	O utilizador visualiza a página de backoffice de cliente com todas as operações possíveis.

Tabela C.1 - UC01: Backoffice de Cliente - Gerir Subscrição

Nome	Consultar dados de subscrição
ID	UC02
Descrição	O utilizador pretende consultar os dados relativos à sua subscrição.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de Backoffice;
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega na opção “Subscrição” do menu principal; 2. A aplicação mostra um submenu com várias opções relacionadas com a subscrição;

	<ol style="list-style-type: none"> 3. O utilizador carrega na opção “Dados de Subscrição” do submenu; 4. A aplicação redireciona o utilizador para a página de detalhes de subscrição.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza os detalhes de subscrição com os respetivos dados preenchidos (Nome de subscrição, email de subscrição, contacto, país, idioma, NIF, pacote subscrito, recorrência de pagamento, número de licenças subscritas).

Tabela C.2 - UC02: Backoffice de Cliente - Consultar dados de subscrição

Nome	Alterar dados de subscrição
ID	UC03
Descrição	O utilizador pretende alterar os dados que estão associados à sua subscrição.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de detalhes dos dados de subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador clica no botão “Editar”; 2. A aplicação redireciona o utilizador para o formulário dos dados de subscrição; 3. O utilizador altera os dados que pretende; 4. O utilizador carrega no botão “Guardar”; 5. A aplicação valida os dados preenchidos; 6. A aplicação informa o utilizador que os dados foram guardados com sucesso; 7. O utilizador é redirecionado para a página de gestão da subscrição.
Fluxo Alternativo	<p>4a: O utilizador carrega em “Cancelar” 4a1: A aplicação redireciona o utilizador para a página de gestão da subscrição.</p> <p>6a: Existem campos obrigatórios por preencher 6a1: A aplicação informa o utilizador que existem campos obrigatórios por preencher.</p> <p>6b: Os dados preenchidos estão inválidos 6b1: A aplicação informa o utilizador que existem campos inválidos.</p>
Pós-Condições	Os dados de subscrição são atualizados na base de dados.

Tabela C.3 - UC03: Backoffice de Cliente - Alterar dados de Subscrição

Nome	Estender a funcionalidade da subscrição atual
ID	UC04
Descrição	O utilizador pretende estender a funcionalidade da sua subscrição, podendo adquirir mais licenças, mais espaço ou funcionalidades extra.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 4. O utilizador iniciou sessão na aplicação com o email da subscrição; 5. O utilizador encontra-se na página de backoffice.
Fluxo Principal	<ol style="list-style-type: none"> 6. O utilizador carrega na opção “Serviços Adicionais” do menu principal; 7. A aplicação redireciona o utilizador para a página de Serviços Adicionais; 8. O utilizador seleciona o que pretende adquirir e quantidade (mais licenças, mais espaço); 9. A aplicação atualiza a informação disponível na página de preço a pagar, fazendo o cálculo a pagar proporcional ao tempo que faltava para terminar o mês/ ano 10. O utilizador carrega no botão “Confirmar”; 11. A aplicação valida as novas aquisições; 12. A aplicação informa o utilizador que estendeu a sua subscrição; 13. Atualiza as informações visualizadas da subscrição 14. A aplicação redireciona o utilizador para a página de gestão da subscrição.
Fluxo Alternativo	<p>7a: O utilizador não selecionou novas aquisições 7a1: A aplicação informa o utilizador que tem que selecionar o que pretende adquirir.</p> <p>7b: O utilizador selecionou que pretendia adquirir novas licenças ou mais espaço, mas não indicou a quantidade 7b1: A aplicação informa o utilizador que tem que indicar a quantidade.</p>
Pós-Condições	A subscrição é estendida e a aplicação envia um e-mail ao utilizador com informação das novas aquisições e preço mensal.

Tabela C.4 - UC04: Backoffice de Cliente - Estender funcionalidade da subscrição atual

Nome	Fazer Upgrade da subscrição
ID	UC05

Descrição	O utilizador pretende fazer upgrade da subscrição, passando a ter acesso a mais módulos/serviços (tendo um investimento superior).
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de backoffice; 3. O utilizador está a visualizar o submenu de subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Upgrade” do submenu de subscrição; 2. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 3. O utilizador carrega no botão “Confirmar”; 4. A aplicação valida a palavra-passe introduzida; 5. A aplicação mostra uma mensagem ao utilizador a informar que fez upgrade com sucesso
Fluxo Alternativo	<p>3a: O utilizador carrega no botão “Cancelar” 3a1: A caixa de diálogo é fechada e o utilizador continua a ver a página de gestão da subscrição.</p> <p>5a: A palavra-passe introduzida está errada 5a1: A aplicação informa o utilizador que a palavra-passe está errada.</p> <p>5b: A subscrição ativa já é o pacote mais alto 5b1: A aplicação informa o utilizador que não é possível fazer upgrade porque já possui o pacote mais alto.</p>
Pós-Condições	A aplicação atualiza os dados na base de dados e envia um email ao utilizador a informar das novas condições da subscrição e nova mensalidade a pagar.

Tabela C.5 - UC05: Backoffice de Cliente - Fazer upgrade da subscrição

Nome	Fazer Downgrade da subscrição
ID	UC06
Descrição	O utilizador pretende fazer downgrade da subscrição, deixando de ter acesso a alguns módulos/serviços e passando a pagar menos.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 4. O utilizador encontra-se na página de backoffice; 2. O utilizador está a visualizar o submenu de subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Downgrade” no submenu de subscrição;

	<ol style="list-style-type: none"> 2. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 3. O utilizador carrega no botão “Confirmar”; 4. A aplicação valida a palavra-passe introduzida; 5. A aplicação mostra uma mensagem ao utilizador a informar que fez downgrade com sucesso.
Fluxo Alternativo	<p>3a: O utilizador carrega no botão “Cancelar” 3a1: A caixa de diálogo é fechada e o utilizador continua a ver a página de gestão da subscrição.</p> <p>5a: A palavra-passe introduzida está errada 5a1: A aplicação informa o utilizador que a palavra-passe está errada.</p> <p>5b: A subscrição ativa já é o pacote mais baixo 5b1: A aplicação informa o utilizador que não é possível fazer downgrade porque já possui o pacote mais baixo.</p>
Pós-Condições	A aplicação atualiza os dados na base de dados e envia um email ao utilizador a informar das novas condições da subscrição e nova mensalidade a pagar.

Tabela C.6 - UC06: Backoffice de Cliente - Fazer downgrade da subscrição

Nome	Suspender Subscrição Paga
ID	UC07
Descrição	O utilizador pretende suspender a subscrição, deixando de ter acesso às funcionalidades e passando a pagar 1€ por mês, enquanto a subscrição estiver suspensa.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. Não tem o pagamento de modalidade anual escolhido ou não está em data de renovação 5. O utilizador encontra-se na página de backoffice; 6. O utilizador está a visualizar o submenu de subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Suspender” no submenu de subscrição; 2. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 3. O utilizador carrega no botão “Confirmar”; 4. A aplicação valida a palavra-passe introduzida; 5. A aplicação informa o utilizador que a subscrição foi suspensa e que passará a ter o investimento de 1€.
Fluxo Alternativo	3a: O utilizador carrega no botão “Cancelar”

	<p>3a1: A caixa de diálogo é fechada e o utilizador continua a ver a página de gestão da subscrição.</p> <p>5a: A palavra-passe introduzida está errada</p> <p>5a1: A aplicação informa o utilizador que a palavra-passe está errada.</p>
Pós-Condições	A aplicação atualiza os dados na base de dados e envia um email ao utilizador a informar que a subscrição está suspensa e que deixará de ter acesso às funcionalidades, pagando apenas 1€ por mês. Na página de gestão de subscrição passa a estar disponível apenas a opção “Retomar Subscrição”.

Tabela C.7 - UC07: Backoffice de Cliente - Suspender subscrição paga

Nome	Retomar subscrição suspensa
ID	UC08
Descrição	O utilizador tem a sua subscrição suspensa e pretende retomá-la, voltando a pagar o valor do seu serviço e voltando a ter acesso às funcionalidades.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. A subscrição está suspensa; 3. O utilizador foi redirecionado para a página de backoffice; 4. O utilizador está a visualizar o submenu de subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Retomar” no submenu de subscrição; 2. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 3. O utilizador carrega no botão “Confirmar”; 4. A aplicação valida a palavra-passe introduzida; 5. A aplicação informa o utilizador que a subscrição foi reativada e que voltará a ter acesso a todas as funcionalidades subscritas e voltará a pagar o valor normal da sua subscrição.
Fluxo Alternativo	<p>3a: O utilizador carrega no botão “Cancelar”</p> <p>3a1: A caixa de diálogo é fechada e o utilizador continua a ver a página de gestão da subscrição.</p> <p>5a: A palavra-passe introduzida está errada</p> <p>5a1: A aplicação informa o utilizador que a palavra-passe está errada.</p>
Pós-Condições	A aplicação atualiza os dados na base de dados e envia um email ao utilizador a informar que a subscrição foi reativada. O utilizador volta a ter acesso a todas as funcionalidades subscritas.

Tabela C.8 - UC08: Backoffice de Cliente - Retomar subscrição suspensa

Nome	Comprar Novas Licenças
ID	UC09
Descrição	O utilizador pretende adquirir novos serviços oferecidos pela KAMAE.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de dados da subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Licenças”; 2. A aplicação mostra uma caixa de diálogo com o número de licenças subscritas e onde o utilizador pode editar o número de licenças que pretende no seu plano; 3. O utilizador indica o número de licenças que pretende; 4. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 5. O utilizador carrega no botão “Confirmar”; 6. A aplicação valida a palavra-passe introduzida; 7. A aplicação informa o utilizador que atualizou o número de licenças.
Fluxo Alternativo	<p>5a: O utilizador carrega no botão “Cancelar” 5a1: A caixa de diálogo é fechada e o utilizador continua a ver a página de dados de subscrição.</p> <p>7a: A palavra-passe introduzida está errada 7a1: A aplicação informa o utilizador que a palavra-passe está errada.</p>
Pós-Condições	A aplicação atualiza os dados na base de dados e envia um email ao utilizador a informar que atualizou o número de licenças, indicando o novo valor a pagar.

Tabela C.9 - UC09: Backoffice de Cliente - Comprar novas licenças

Nome	Alterar Dados de Pagamento
ID	UC10
Descrição	O utilizador pretende alterar os dados de pagamento da sua subscrição
Utilizadores	Utilizador autenticado.

Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de backoffice; 3. O utilizador está a visualizar o submenu de subscrição.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Dados de Pagamento” no submenu de subscrição; 2. A aplicação redireciona o utilizador para o formulário dos dados de pagamento; 3. O utilizador altera os dados; 4. O utilizador carrega no botão “Gravar”; 5. A aplicação valida os dados alterados; 6. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 7. O utilizador carrega no botão “Confirmar”; 8. A aplicação valida a palavra-passe introduzida; 9. A aplicação informa o utilizador que os dados de pagamento foram alterados com sucesso.
Fluxo Alternativo	<p>4a: O utilizador carrega no botão “Cancelar” 4a1: A aplicação redireciona o utilizador para a página de gestão da subscrição.</p> <p>6a: Existem campos obrigatórios por preencher 6a1: A aplicação informa o utilizador que existem campos obrigatórios por preencher.</p> <p>6b: Existem campos inválidos 6b1: A aplicação informa o utilizador que existem campos inválidos.</p> <p>8a: O utilizador carrega no botão “Cancelar” 8a1: A caixa de diálogo é fechada e o utilizador continua a ver o formulário dos dados de pagamento.</p> <p>9a: A palavra-passe introduzida está errada 9a1: A aplicação informa o utilizador que a palavra-passe está errada.</p>
Pós-Condições	Os dados de pagamento são atualizados na base de dados.

Tabela C.10 - UC10: Backoffice de Cliente - Alterar dados de pagamento

Nome	Alterar dados de Faturação
ID	UC11
Descrição	O utilizador pretende alterar seus dados de faturação.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de Backoffice.

Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Faturação” no menu principal; 2. A aplicação mostra um submenu com várias opções relacionadas com a faturação; 3. O utilizador carrega em “Dados de Faturação” no submenu; 4. A aplicação redireciona o utilizador para o formulário dos dados de faturação; 5. O utilizador altera os dados; 6. O utilizador carrega no botão “Gravar”; 7. A aplicação valida os dados alterados; 8. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 9. O utilizador carrega no botão “Confirmar”; 10. A aplicação valida a palavra-passe introduzida; 11. A aplicação informa o utilizador que os dados de faturação foram alterados com sucesso.
Fluxo Alternativo	<p>6a: O utilizador carrega no botão “Cancelar” 6a1: A aplicação redireciona o utilizador para a página de gestão da subscrição.</p> <p>8a: Existem campos obrigatórios por preencher 8a1: A aplicação informa o utilizador que existem campos obrigatórios por preencher.</p> <p>8b: Existem campos inválidos 8b1: A aplicação informa o utilizador que existem campos inválidos.</p> <p>9a: O utilizador carrega no botão “Cancelar” 9a1: A caixa de diálogo é fechada e o utilizador continua a ver o formulário dos dados de pagamento.</p> <p>11a: A palavra-passe introduzida está errada 11a1: A aplicação informa o utilizador que a palavra-passe está errada.</p>
Pós-Condições	Os dados de faturação são atualizados na base de dados.

Tabela C.11 - UC11: Backoffice de Cliente - Alterar dados de faturação

Nome	Consultar espaço ocupado
ID	UC12
Descrição	O utilizador pretende consultar o espaço subscrito que já está ocupado.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de backoffice.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador visualiza um gráfico com a ocupação e espaço livre, disponível na homepage do Backoffice.

Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza um gráfico que lhe indica o espaço ocupado com cada tipo de registo e o espaço que tem ainda disponível.

Tabela C.12 - UC12: Backoffice de Cliente - Consultar espaço ocupado

Nome	Consultar registo de entrada e saída de utilizadores
ID	UC13
Descrição	O utilizador pretende consultar o registo de entrada e saída de utilizadores na plataforma.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de Backoffice.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Histórico” no menu principal; 2. A aplicação redireciona o utilizador para a página de consulta do registo de entradas e saídas de utilizadores.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador passa a visualizar uma tabela com os registos de acessos e de operações por utilizador.

Tabela C.13 - UC13: Backoffice de Cliente - Consultar registo de entrada e saída de utilizadores

Nome	Consultar conta-corrente e faturação
ID	UC14
Descrição	O utilizador pretende consultar a conta-corrente e faturação da subscrição.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação com o email da subscrição; 2. O utilizador encontra-se na página de Backoffice; 3. O utilizador está a visualizar o submenu de faturação.

Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Conta-corrente”; 2. A aplicação redireciona o utilizador para a página de consulta da conta-corrente e faturação.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador passa a visualizar uma tabela com os registos de conta-corrente e faturação por mês tendo a possibilidade de clicar nos movimentos e realizar o <i>download</i> dos documentos de faturação.

Tabela C.14 - UC14: Backoffice de Cliente - Consultar conta-corrente e faturação

Nome	Descarregar Fatura
ID	UC15
Descrição	O utilizador pretende descarregar a fatura para o seu computador.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação S4L com o email da subscrição; 2. O utilizador encontra-se na página de consulta de conta-corrente e de faturação.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Descarregar” presente na linha da tabela correspondente à fatura que pretende descarregar; 2. A aplicação inicia a transferência do ficheiro.
Fluxo Alternativo	Não tem.
Pós-Condições	O ficheiro da fatura passa a estar disponível no computador do utilizador.

Tabela C.15 - UC15: Backoffice de Cliente - Descarregar fatura

Nome	Consultar Informações de afiliado
ID	UC16
Descrição	O utilizador pretende consultar as suas informações de afiliado, verificando quem são os seus afiliados, qual o valor de comissão que já teve, qual o valor que vai receber e quando.

Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação S4L com o email da subscrição; 2. O utilizador encontra-se na página de Backoffice.
Fluxo Principal	<ol style="list-style-type: none"> 3. O utilizador carrega em “Afiliados” no menu principal; 4. A aplicação redireciona o utilizador para a página de informações de afiliação.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza quem são os seus afiliados, qual o valor de comissão que já teve, qual o valor que vai receber e quando. O utilizador pode ainda visualizar o seu link de afiliação e convidar novos possíveis afiliados.

Tabela C.16 - UC16: Backoffice de Cliente - Consultar informações de afiliado

Nome	Convidar afiliados
ID	UC17
Descrição	O utilizador pretende convidar novos possíveis afiliados, de forma a conseguir adquirir mais benefícios na sua subscrição.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação S4L com o email da subscrição; 2. O utilizador encontra-se na página de informações de afiliação.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Convidar”, presente junto ao seu link de afiliação; 2. A aplicação abre uma caixa de diálogo para o utilizador inserir os endereços de e-mail das pessoas que pretende convidar; 3. O utilizador insere os endereços de email; 4. O utilizador carrega em “Convidar”; 5. A aplicação valida os endereços de email introduzidos; 6. A aplicação informa o utilizador de que o seu link de afiliação foi enviado para os endereços indicados.

Fluxo Alternativo	<p>4a: O utilizador carrega no botão “Cancelar”</p> <p>4a1: A aplicação fecha a caixa de diálogo;</p> <p>4a2: O utilizador continua a ver as informações de afiliação.</p> <p>6a: A aplicação deteta que o utilizador introduziu endereços de email inválidos</p> <p>6a1: A aplicação informa o utilizador que existem endereços de email inválidos;</p> <p>6a2: Continua em 3.</p>
Pós-Condições	É enviado um email, com o link de afiliação do utilizador, para todos os endereços de email introduzidos por este na caixa de diálogo.

Tabela C.17 - UC17: Backoffice de Cliente - Convidar Afiliados

Nome	Fazer reset dos dados
ID	UC18
Descrição	O utilizador pretende eliminar todos os dados já inseridos por utilizadores da sua subscrição.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão na aplicação S4L com o email da subscrição; 2. O utilizador encontra-se na página de Backoffice.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em “Reset” no menu principal; 2. A aplicação informa o utilizador que todos os dados serão eliminados (à exceção dos utilizadores) e que se trata de uma operação irreversível, questionando o utilizador se deseja continuar; 3. O utilizador carrega em “Confirmar”; 4. A aplicação mostra uma caixa de diálogo para o utilizador introduzir a palavra-passe; 5. O utilizador carrega no botão “Confirmar”; 6. A aplicação valida a palavra-passe introduzida; 7. A aplicação informa o utilizador que os dados da sua subscrição foram eliminados.
Fluxo Alternativo	<p>3a: O utilizador carrega no botão “Cancelar”</p> <p>3a1: A aplicação fecha a caixa de diálogo.</p> <p>5a: O utilizador carrega no botão “Cancelar”</p>

	<p>5a1: A aplicação fecha a caixa de diálogo.</p> <p>7a: A palavra-passe introduzida está errada</p> <p>7a1: A aplicação informa o utilizador que a palavra-passe está errada.</p> <p>7b: Existem documentos de faturação já emitidos</p> <p>7b1: A aplicação informa o utilizador que não é possível fazer reset aos dados porque existe documentos de faturação emitidos.</p>
Pós-Condições	Todo o trabalho já registado no sistema é eliminado.

Tabela C.18 - UC18: Backoffice de Cliente - Fazer reset dos dados

b) Backoffice Kamae

Nome	Ver dashboard de pagamentos
ID	UC19
Descrição	O utilizador pretende visualizar o dashboard de pagamentos, onde tem indicadores e gráficos úteis para análise.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página principal do Backoffice.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador seleciona a opção de menu “Pagamentos”; 2. A aplicação redireciona o utilizador para a página de pagamentos; 3. O utilizador visualiza o dashboard de pagamentos.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza um dashboard com vários gráficos referentes a pagamentos, tendo disponível um botão para gerir pagamentos.

Tabela C.19 - UC19: Backoffice Kamae - Ver dashboard de pagamentos

Nome	Ver detalhes no dashboard de pagamentos
ID	UC20
Descrição	O utilizador pretende visualizar informação mais detalhada relativamente a um gráfico do dashboard de pagamentos.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página de dashboard de pagamentos.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega em cima do gráfico desejado; 2. A aplicação abre uma caixa de diálogo; 3. O utilizador visualiza a tabela de detalhes do gráfico escolhido.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador passa a visualizar uma tabela onde tem várias informações relativamente aos dados do gráfico clicado.

Tabela C.20 - UC20: Backoffice Kamae - Ver detalhes no dashboard de pagamentos

Nome	Listar Recorrência de pagamento
ID	UC21
Descrição	O utilizador pretende listar as recorrências de pagamento.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se no dashboard de pagamentos.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Gerir pagamentos” 2. A aplicação redireciona o utilizador para a página de gestão de pagamentos; 3. O utilizador carrega no separador “Recorrências”; 4. A vista da página é atualizada.

Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza a lista de recorrências, podendo realizar várias operações, como adicionar uma nova recorrência, editar ou eliminar uma já existente.

Tabela C.21 - UC21: Backoffice Kamae - Listar Recorrência de pagamento

Nome	Adicionar Recorrência de Pagamento
ID	UC22
Descrição	O utilizador pretende adicionar uma nova recorrência de pagamento para as subscrições do serviço.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página de gestão pagamentos; 3. O utilizador está a visualizar a lista de recorrências.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Criar Novo”; 2. A aplicação mostra uma caixa de diálogo com o formulário para preencher os campos da nova recorrência de pagamento; 3. O utilizador carrega no botão “Gravar”; 4. A aplicação valida os campos preenchidos; 5. A aplicação informa o utilizador que foi criada uma nova recorrência de pagamento.
Fluxo Alternativo	<p>4a: O utilizador carrega no botão “Cancelar”</p> <p>4a1: A aplicação fecha a caixa de diálogo;</p> <p>4a2: Continua em 2.</p> <p>7a: Existem campos obrigatórios por preencher</p> <p>7a1: A aplicação informa o utilizador que existem campos obrigatórios por preencher;</p> <p>7a2: Continua em 4.</p>
Pós-Condições	É criada uma nova recorrência de pagamento e, desde que esteja ativa, passa a ser possível criar novas subscrições com a recorrência

	criada. A lista de recorrências que o utilizador está a visualizar é atualizada.
--	--

Tabela C.22 - UC22: Backoffice Kamae - Adicionar recorrência de pagamento

Nome	Remover Recorrência de Pagamento
ID	UC23
Descrição	O utilizador pretende eliminar uma recorrência de pagamento.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE 2. O utilizador encontra-se na página de gestão de pagamentos; 3. O utilizador está a visualizar a lista de recorrências.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Eliminar” correspondente à recorrência de pagamento que pretende eliminar; 2. A aplicação mostra uma caixa de diálogo de confirmação que deseja realmente eliminar a recorrência; 3. O utilizador carrega no botão “Confirmar”; 4. A aplicação informa o utilizador que a recorrência foi eliminada; 5. A aplicação atualiza a lista de recorrências.
Fluxo Alternativo	<p>4a: O utilizador carrega no botão “Cancelar”</p> <p>4a1: A aplicação fecha a caixa de diálogo;</p> <p>4a2: Continua em 1.</p>
Pós-Condições	A recorrência é marcada como eliminada na base de dados, mas as subscrições já existentes com essa recorrência, mantém-se, apenas deixa de ser possível criar novas.

Tabela C.23 - UC23: Backoffice Kamae - Remover recorrência de pagamento

Nome	Editar Recorrência de Pagamento
ID	UC24

Descrição	O utilizador pretende alterar os dados de uma recorrência de pagamento.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página de gestão pagamentos; 3. O utilizador está a visualizar a lista de recorrências.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Editar” correspondente à recorrência de pagamento que pretende editar; 2. A aplicação mostra uma caixa de diálogo com o formulário de recorrência e com os campos preenchidos; 3. O utilizador altera os campos que pretende; 4. O utilizador carrega no botão “Gravar”; 5. A aplicação valida os campos preenchidos; 6. A aplicação informa o utilizador que a recorrência foi gravada.
Fluxo Alternativo	<p>5a: O utilizador carrega no botão “Cancelar”</p> <p>5a1: A aplicação fecha a caixa de diálogo;</p> <p>5a2: Continua em 1.</p> <p>7a: Existem campos obrigatórios por preencher</p> <p>7a1: A aplicação informa o utilizador que existem campos obrigatórios por preencher;</p> <p>7a2: Continua em 4.</p>
Pós-Condições	Os dados da recorrência de pagamento são atualizados na base de dados.

Tabela C.24 - UC24: Backoffice Kamae - Editar recorrência de pagamento

Nome	Listar Tipos de pagamento
ID	UC25
Descrição	O utilizador pretende listar os tipos de pagamento.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se no dashboard de pagamentos.

Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Gerir pagamentos”; 2. A aplicação redireciona o utilizador para a página de gestão de pagamentos; 3. O utilizador carrega no separador “Tipos de Pagamento”; 4. A vista da página é atualizada.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza a lista de tipos de pagamento, podendo realizar várias operações, como adicionar um novo tipo de pagamento, editar ou eliminar um já existente.

Tabela C.25 - UC25: Backoffice Kamae - Listar tipos de pagamento

Nome	Adicionar Tipo de Pagamento
ID	UC26
Descrição	O utilizador pretende adicionar um novo tipo de pagamento para as subscrições do serviço.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página de gestão de pagamentos; 3. O utilizador está a visualizar a lista de tipos de pagamento.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Criar Novo”; 2. A aplicação mostra uma caixa de diálogo com o formulário para preencher os campos do novo tipo de pagamento; 3. O utilizador carrega no botão “Gravar”; 4. A aplicação valida os campos preenchidos; 5. A aplicação informa o utilizador que foi criada um novo tipo de pagamento.
Fluxo Alternativo	<p>4a: O utilizador carrega no botão “Cancelar”</p> <p>4a1: A aplicação fecha a caixa de diálogo;</p> <p>4a2: Continua em 2.</p> <p>7a: Existem campos obrigatórios por preencher</p> <p>7a1: A aplicação informa o utilizador que existem campos obrigatórios por preencher;</p> <p>7a2: Continua em 4.</p>

Pós-Condições	É criado um novo tipo de pagamento e, desde que esteja ativo, passa a ser possível criar novas subscrições com o tipo de pagamento criado. A lista de tipos de pagamento que o utilizador está a visualizar é atualizada.
----------------------	---

Tabela C.26 - UC26: Backoffice Kamae - Adicionar tipo de pagamento

Nome	Remover Tipo de Pagamento
ID	UC27
Descrição	O utilizador pretende eliminar um tipo de pagamento.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página de gestão de pagamentos; 3. O utilizador está a visualizar a lista de tipos de pagamento.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Eliminar” correspondente ao tipo de pagamento que pretende eliminar; 2. A aplicação mostra uma caixa de diálogo de confirmação que deseja realmente eliminar o registo; 3. O utilizador carrega no botão “Confirmar”; 4. A aplicação informa o utilizador que o tipo de pagamento foi eliminado; 5. A aplicação atualiza a lista de tipos de pagamento.
Fluxo Alternativo	<p>4a: O utilizador carrega no botão “Cancelar”</p> <p>4a1: A aplicação fecha a caixa de diálogo;</p> <p>4a2: Continua em 1.</p>
Pós-Condições	A tipo de pagamento é marcado como eliminado na base de dados, mas as subscrições já existentes com esse tipo de pagamento, mantém-se, apenas deixa de ser possível criar novas.

Tabela C.27 – UC27: Backoffice Kamae - Remover tipo de pagamento

Nome	Editar Tipo de Pagamento
ID	UC28
Descrição	O utilizador pretende alterar os dados de um tipo de pagamento.

Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página de gestão de pagamentos; 3. O utilizador está a visualizar a lista de tipos de pagamento.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega no botão “Editar” correspondente ao tipo de pagamento que pretende editar; 2. A aplicação mostra uma caixa de diálogo com o formulário de tipo de pagamento e com os campos preenchidos; 3. O utilizador altera os campos que pretende; 4. O utilizador carrega no botão “Gravar”; 5. A aplicação valida os campos preenchidos; 6. A aplicação informa o utilizador que o tipo de pagamento foi gravado.
Fluxo Alternativo	<p>5a: O utilizador carrega no botão “Cancelar”</p> <p>5a1: A aplicação fecha a caixa de diálogo;</p> <p>5a2: Continua em 1.</p> <p>7a: Existem campos obrigatórios por preencher</p> <p>7a1: A aplicação informa o utilizador que existem campos obrigatórios por preencher;</p> <p>7a2: Continua em 4.</p>
Pós-Condições	Os dados do tipo de pagamento são atualizados na base de dados.

Tabela C.28 - UC28: Backoffice Kamae - Editar tipo de pagamento

Nome	Visualizar relatório de subscrições
ID	UC29
Descrição	O utilizador pretende visualizar um relatório com informações relativas às subscrições do serviço.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página principal do backoffice.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador carrega na opção de menu “Relatórios”; 2. O menu de relatórios expande, passando a mostrar um submenu; 3. O utilizador carrega na opção do submenu “Subscrições”;

	4. A aplicação redireciona o utilizador para a página de relatório de subscrições.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador visualiza uma tabela com informações relativas a subscrições, sendo possível realizar operações como impressão ou exportação do relatório.

Tabela C.29 - UC29: Backoffice Kamae - Visualizar relatório de subscrições

Nome	Visualizar utilizadores de uma subscrição
ID	UC30
Descrição	O utilizador pretende visualizar informações dos utilizadores de uma determinada subscrição.
Utilizadores	Utilizador autenticado.
Pré-Condições	<ol style="list-style-type: none"> 1. O utilizador iniciou sessão no Backoffice da KAMAE; 2. O utilizador encontra-se na página de relatório das subscrições.
Fluxo Principal	<ol style="list-style-type: none"> 1. O utilizador está a visualizar o relatório de subscrições; 2. O utilizador carrega em cima da linha da tabela correspondente à subscrição que deseja ver; 3. É aberta uma caixa de diálogo.
Fluxo Alternativo	Não tem.
Pós-Condições	O utilizador passa a visualizar a lista de utilizadores correspondente à subscrição em que clicou.

Tabela C.30 - UC30: Backoffice Kamae - Visualizar utilizadores de uma subscrição

Apêndice D – Arquitetura

Este apêndice surgiu com base na necessidade de reformulação à arquitetura inicialmente prevista, pois a empresa decidiu que era necessário fazer algumas alterações.

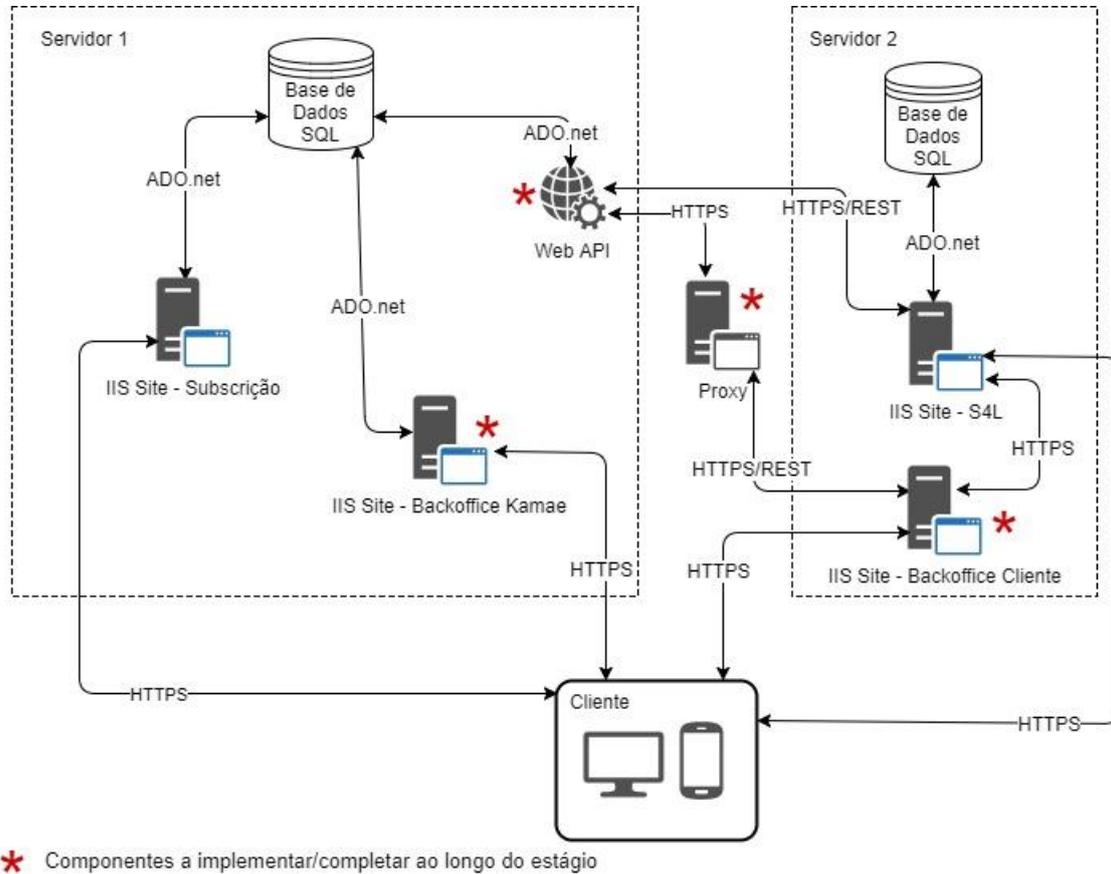


Figura D.1 - Diagrama de arquitetura do sistema (antigo)

Confrontando o diagrama da figura D.1 com o presente no capítulo de arquitetura, correspondente à arquitetura atual do sistema, é visível que o BO de Cliente deixou de estar no servidor onde está o site de frontend e passou para o servidor de BO. Esta alteração foi uma decisão da empresa e justifica-se pela forte necessidade de acesso a dados de backoffice por parte do componente movido. Foram ainda acrescentados serviços externos, que tinham ficado esquecidos no planejamento inicial, mas que são fundamentais para a realização de algumas operações no BO de cliente.

Apêndice E – Protótipos

Este apêndice surgiu como complemento ao capítulo 8, de forma a demonstrar os restantes protótipos, desenvolvidos no decorrer da primeira fase do estágio.

The screenshot shows a web browser window titled 'A Web Page' with a URL bar containing 'https://'. The page content is a 'Backoffice' interface. On the left is a dark sidebar menu with the following items: 'Subscrição', 'Faturação', 'Dados Faturação', 'Conta Corrente', 'Histórico', 'Afiliações', 'Serviços Adicionais', and 'Reset'. The main content area is titled 'Backoffice' and 'Dados de Faturação'. It contains a form with the following fields: 'Nome:' (text input with 'Nome exemplo'), 'Email:' (text input with 'email@exemplo.s4l'), 'Morada:' (text input with 'Rua x, 0000-000 Coimbra'), 'NIF:' (text input with '999999990'), 'Telefone:' (text input with '979797979'), 'Telemovel:' (text input with '979797979'), and 'País:' (dropdown menu with 'Portugal' selected). At the bottom right of the form are two buttons: 'Guardar' (with a pencil icon) and 'Cancelar' (with a circular arrow icon).

Figura E.1 - Formulário de dados de faturação

This screenshot shows the same 'Backoffice' page as Figure E.1, but with a success message overlay. The message is a white box with a grey border containing the text 'Os dados foram guardados com sucesso!' and a blue 'OK' button. The form fields and buttons are visible in the background, though slightly dimmed.

Figura E.2 - Formulário de dados de faturação - Sucesso a gravar dados alterados

A Web Page

https://

Subscrição

Dados de Subscrição

Dados de Pagamento

Upgrade

Downgrade

Suspender

Faturação

Histórico

Afiliados

Serviços Adicionais

Reset

Backoffice

Dados de Pagamento

Saldo: 2.30€ [Pagar](#)

Tipo de Pagamento

Cartão de Crédito Paypal

Nome do cartão de crédito:

Nome do cartão exemplo

Número do cartão de crédito:

4242424242424242

Validade do cartão:

Mês

Ano

CVC

4563

Recorrência:

Mensal Anual

[Guardar](#) [Cancelar](#)

Figura E.3 - Formulário de dados de pagamento

A Web Page

https://

Subscrição

Dados de Subscrição

Dados de Pagamento

Upgrade

Downgrade

Suspender

Faturação

Histórico

Afiliados

Serviços Adicionais

Reset

Backoffice

Dados de Pagamento

Tipo de Pagamento

Cartão de Crédito Paypal

Nome do cartão de crédito:

Nome do cartão exemplo

Número do cartão de crédito:

4242424242424242

Validade do cartão:

Mês

Ano

CVC

4563

Recorrência:

Mensal Anual

[Guardar](#) [Cancelar](#)

Sucesso

Os dados foram guardados com sucesso!

[OK](#)

Figura E.4 - Formulário de dados de pagamento - Sucesso a gravar

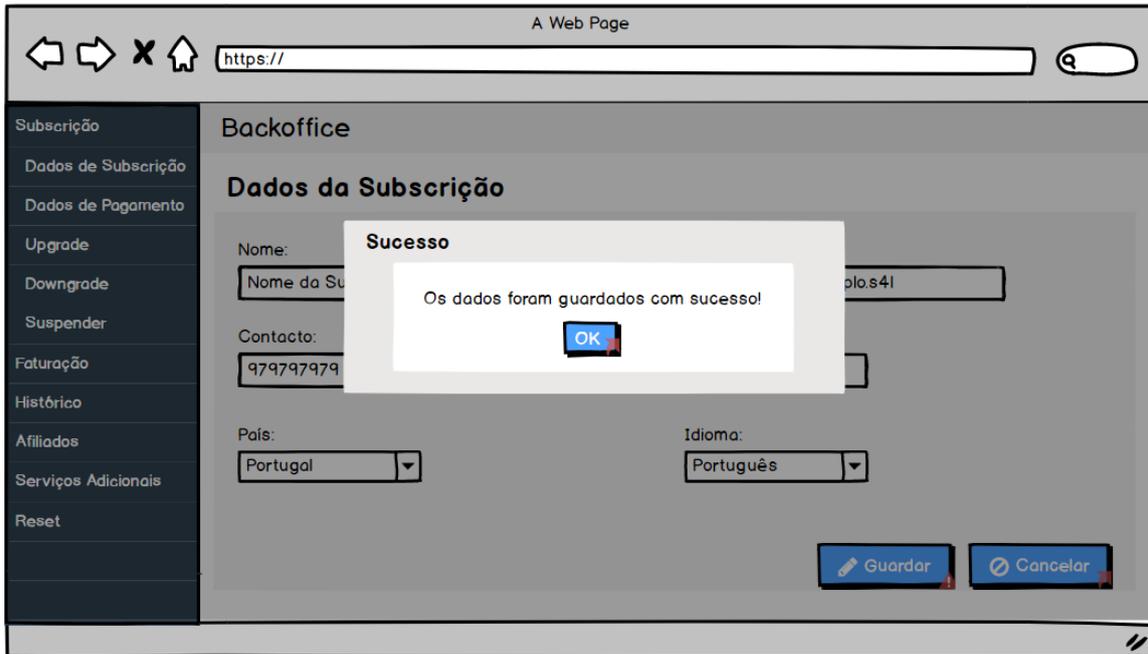


Figura E.5 - Formulário de dados de subscrição - Sucesso a gravar

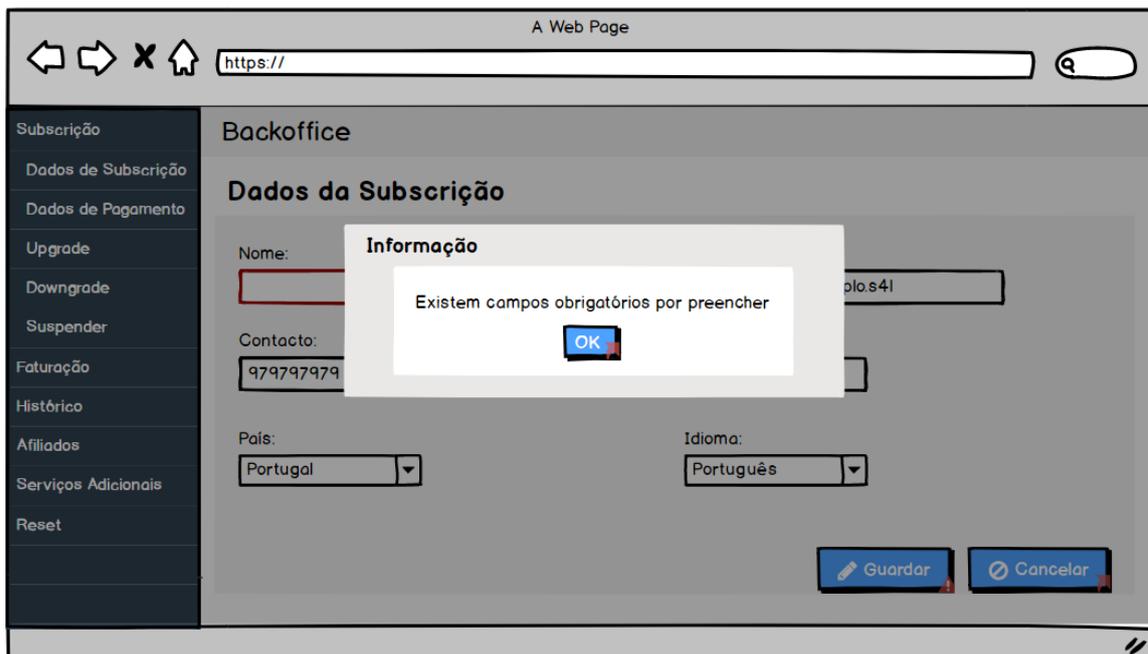


Figura E.6 - Formulário de dados de subscrição - Erro ao gravar

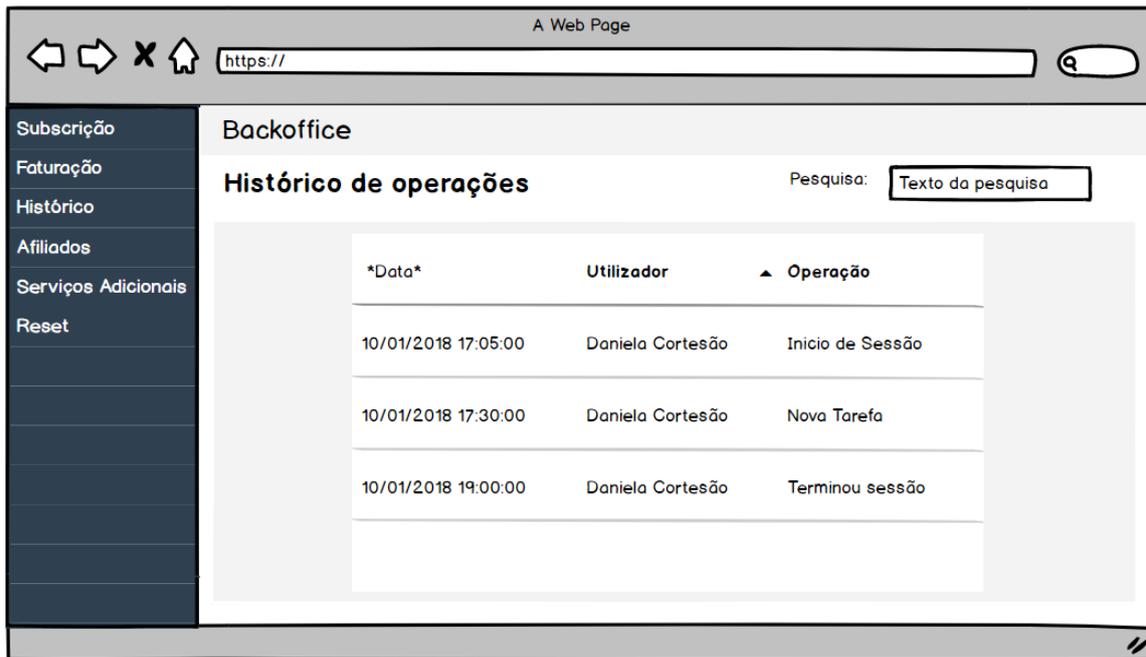


Figura E.7 - Página de histórico de operações

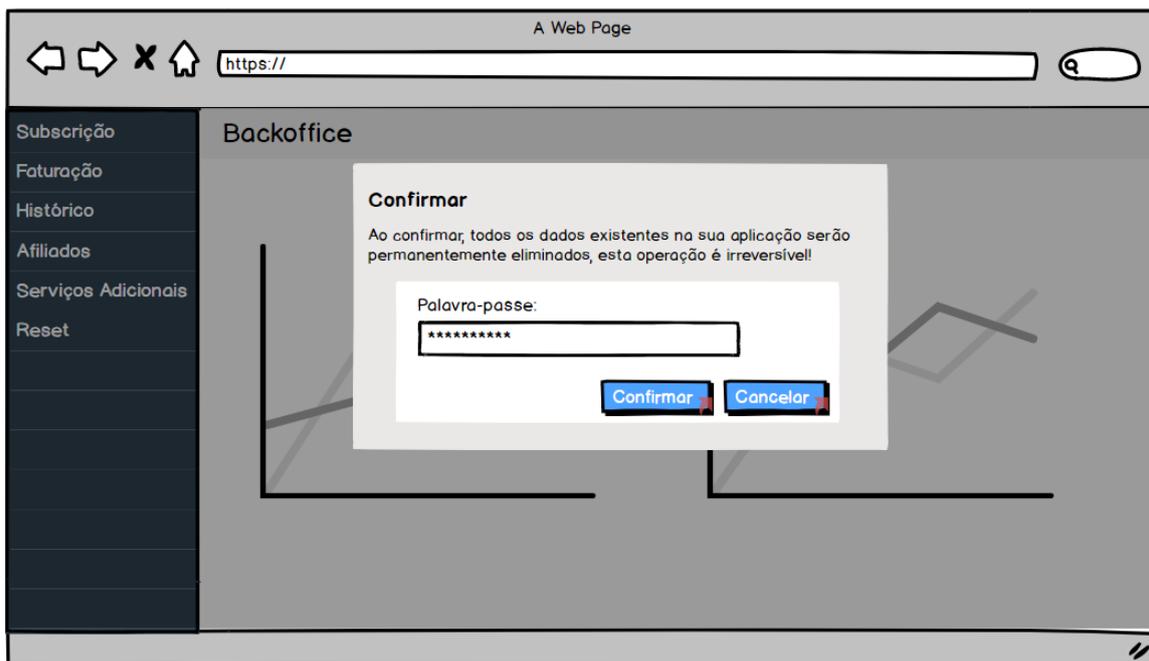


Figura E.8 - Reset dos dados de front-end

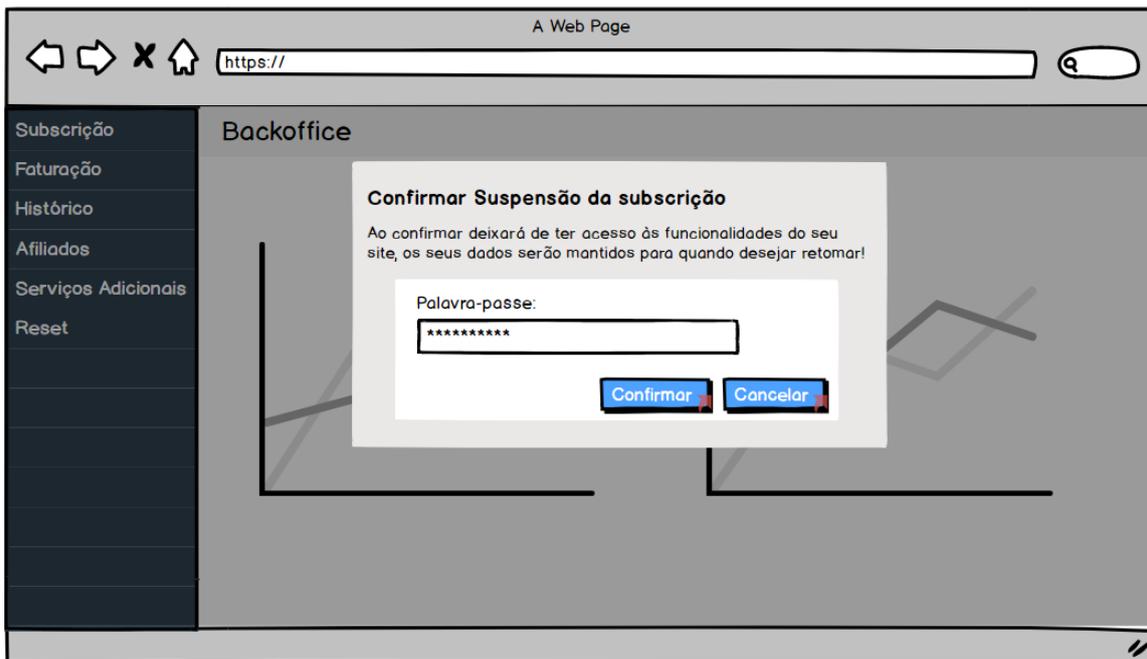


Figura E.9 - Suspende a subscrição

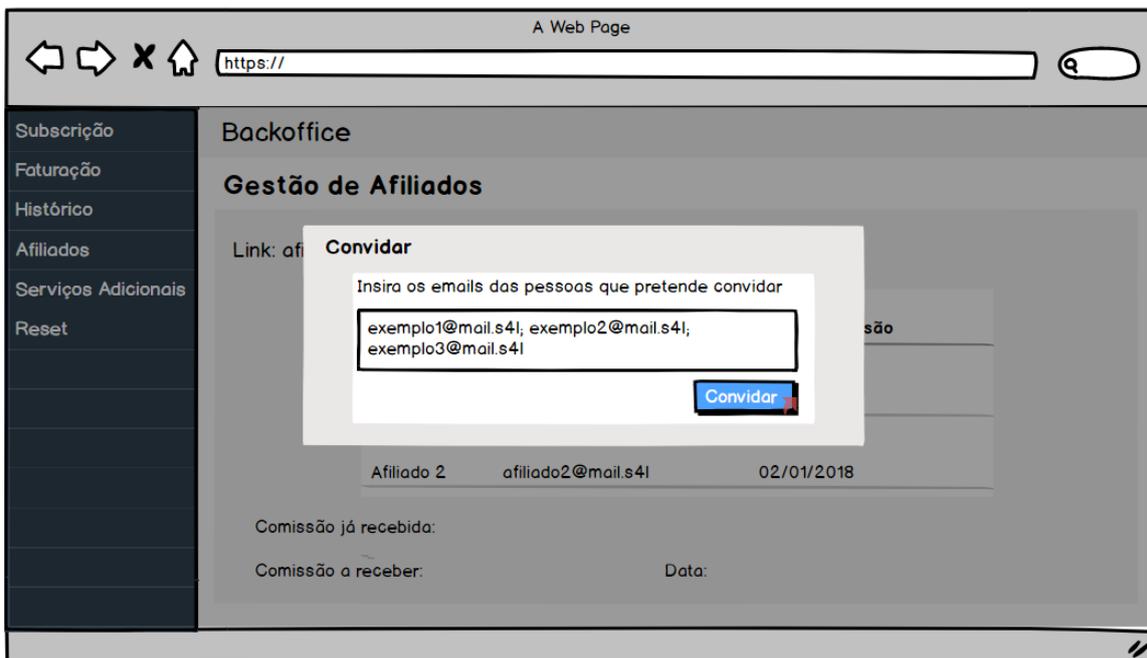


Figura E.10 - Convidar outras pessoas para subscrever o serviço

Apêndice F – Análise aos Gateways de pagamento

- **Stripe**

Esta análise foi realizada antes do início da integração do BO cliente com este meio de pagamento. Contudo, com o desenvolvimento da integração, foi possível verificar que algumas das operações poderiam ser realizadas de outro modo, reduzindo o número de chamadas necessárias à API, e facilitando um pouco a implementação.

1. Upgrade

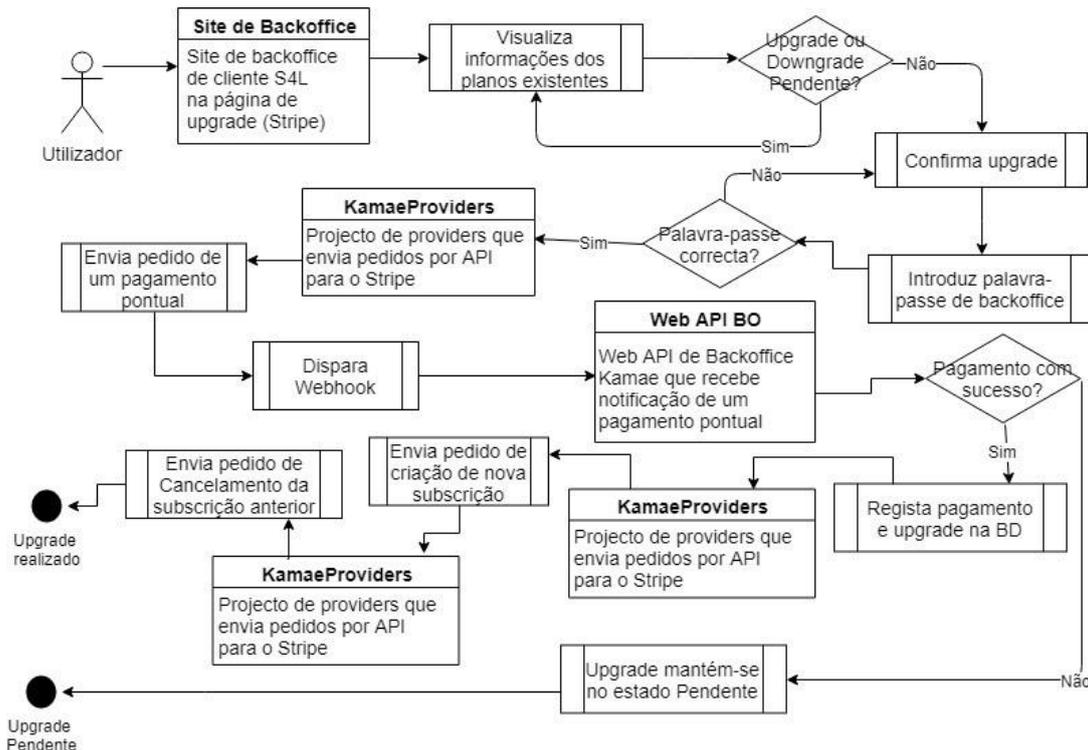


Figura F.1 - Upgrade com Stripe

Tal como foi anteriormente referido, esta análise foi feita antes do início da integração. Por esse motivo, existem algumas diferenças entre o diagrama da figura F.1, correspondente a um upgrade de subscrição, e o que foi realmente implementado. Inicialmente a estagiária pensou que seria necessário cancelar o plano ativo no momento e subscrever o novo, para realizar um upgrade. No entanto, durante o desenvolvimento desta funcionalidade, verificou-se que era possível alterar simplesmente o plano ativo no Stripe, tornando esta operação mais simples de implementar e evitando assim chamadas à API desnecessárias, assim como recebimentos de notificações na Web API, também estes desnecessários.

Os passos abaixo, demonstram o procedimento do modo que foi implementado:

- É enviado um pedido ao Stripe para atualizar o plano, substituindo o atual pelo novo;

- É enviado um pedido ao Stripe para a realização de um pagamento pontual – Corresponde à diferença de valor do plano antigo para o novo plano, nos dias que já foram pagos com o plano antigo, mas que já serão abrangidos pelo novo plano;
- Quando a Web API de backoffice recebe uma notificação (proveniente do Stripe) a indicar que o pagamento pontual foi realizado com sucesso:
 - Regista na BD o pagamento realizado e a mudança de plano para o cliente em questão – desativando o plano anterior e ativando o novo.

Caso não seja recebida nenhuma notificação, ou a notificação recebida indique pagamento falhado, o upgrade mantém-se no estado pendente, mantendo o plano anterior no estado ativo.

2. Downgrade

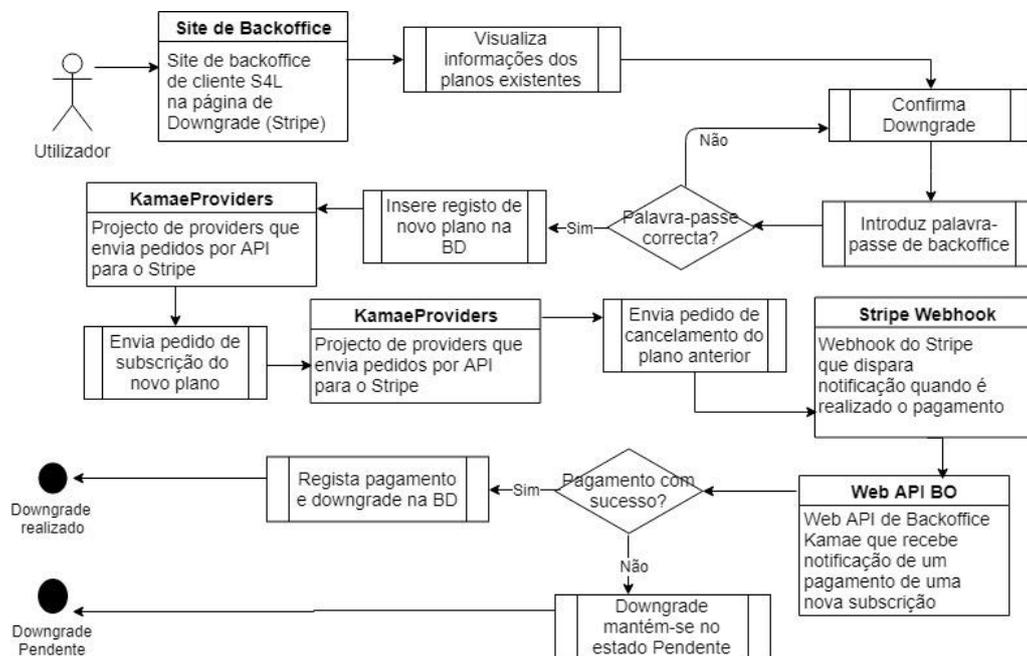


Figura F.2 - Downgrade com Stripe

Tal como no upgrade, também no caso do downgrade foram feitas algumas alterações à análise realizada e representada na figura F.2.

Procedimento implementado:

- É feito um pedido por API ao Stripe para alterar o plano subscrito, substituindo o atual pelo novo;
- A Web API recebe uma notificação de atualização da subscrição:
 - Ativa-se o novo plano na BD, no entanto este só terá início na data em que será realizado o próximo pagamento;
- O downgrade fica pendente (podendo ser cancelado) até à data do próximo pagamento;
- Caso ocorra algum erro ou algum problema no processo de downgrade, é necessário reverter a operação, voltando a mudar para o plano anterior.

3. Suspende

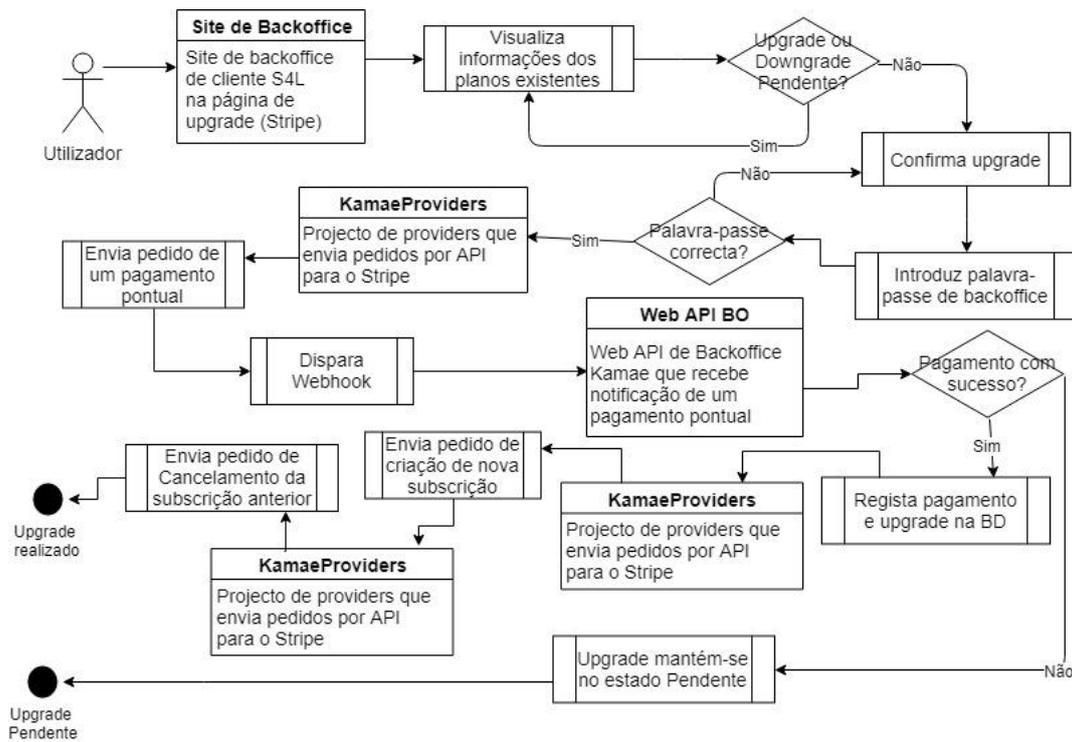


Figura F.3 - Suspende subscrição com Stripe

A inexistência de opção para suspender uma subscrição, no Stripe, obriga a ter um plano extra que corresponda a uma suspensão, de forma a garantir que irá continuar a ser cobrado um valor mensal (inferior ao de uma subscrição ativa). Também aqui, a implementação não cumpre exatamente todos os passos que tinham sido previstos na análise inicial, pois o cancelamento do plano anteriormente ativo deixou de fazer sentido, tendo em conta que a subscrição apenas é alterada, não havendo cancelamento e ativação de novos planos.

Procedimento implementado:

- Após confirmação do desejo de suspensão, é enviado um pedido para o Stripe, com o objetivo de alterar o pacote da subscrição para o correspondente a uma suspensão;
- A web API de BO recebe uma notificação, provida do webhook do Stripe, a indicar que a subscrição foi alterada com sucesso
 - A subscrição passa para o estado suspenso, no entanto encontra-se ativa até ao final do período já pago;
- Caso tenha ocorrido algum problema no processo de suspensão, é necessário reverter a operação, voltando ao plano que estava ativo anteriormente.

4. Cancelar

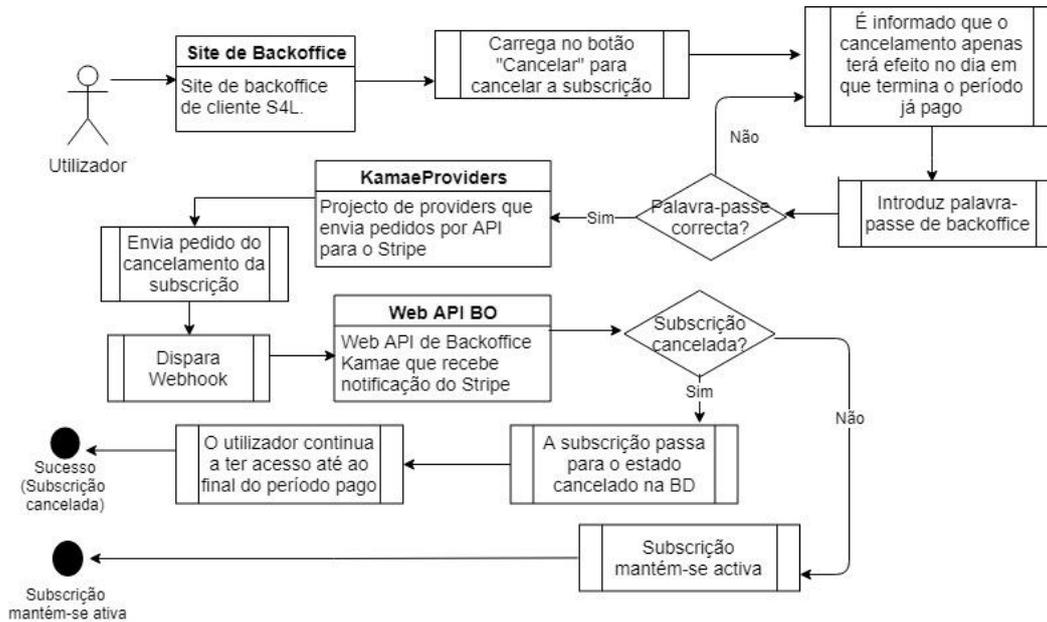


Figura F.4 - Cancelar com Stripe

No caso do cancelamento, o procedimento manteve-se tal como foi planeado na análise realizada, pois é uma operação simples que apenas implica cancelamento da subscrição no final do período já pago.

O procedimento para cancelar uma subscrição com o Stripe é:

- Após confirmação do desejo de cancelamento pelo utilizador, é enviado um pedido ao Stripe, por API, para que a subscrição seja cancelada, indicando que o cancelamento apenas deve ter efeito no final do período;
- A Web API de BO recebe uma notificação de cancelamento da subscrição, proveniente do webhook do Stripe:
 - A subscrição é atualizada na BD, passando ao estado cancelada, ficando esta alteração de estado pendente até ao final do período já pago;
 - O utilizador deixa de pagar e deixa de ter acesso ao seu site S4L assim que terminar o período já pago.
- Caso a notificação indique que a subscrição não foi cancelada, a subscrição do utilizador mantém-se ativa.

5. Cancelar Operação pendente

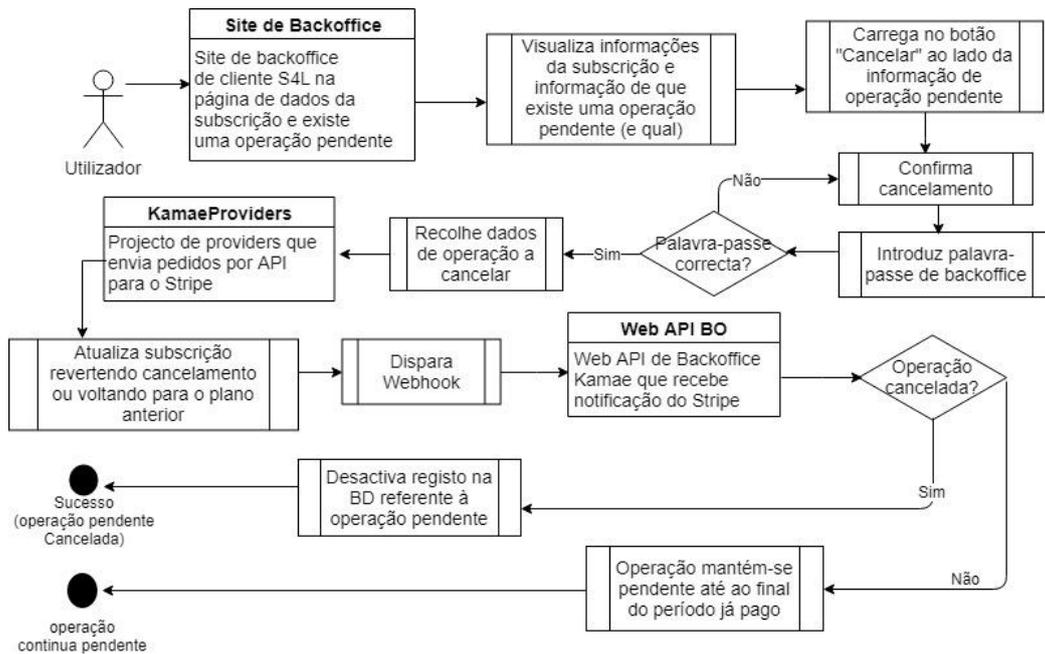


Figura F.5 - Cancelar operação pendente com Stripe

No que diz respeito ao cancelamento de operação pendente, inicialmente tinha sido pensado apenas como cancelamento de upgrades ou downgrades pendentes. Contudo, com a necessidade de permitir cancelamentos e suspensões apenas no final do período já pago, para evitar reembolsos, passou a fazer sentido que esta funcionalidade cobrisse todas essas operações.

O procedimento implementado foi o seguinte:

- O utilizador quando entra no Backoffice tem menos funcionalidades disponíveis e quando entra na página de dados de subscrição é informado de que tem uma operação pendente e que enquanto não passar ao estado ativo não pode realizar outras alterações à sua subscrição;
- Quando confirma o desejo de cancelar a operação pendente, as informações relativas à operação são obtidas da BD e é enviado um pedido ao Stripe, por API, para alterar a subscrição (seja para remover o desejo de cancelamento ou para voltar ao plano anterior, cancelando assim um upgrade, downgrade ou suspensão);
- A Web API de BO recebe uma notificação da alteração:
 - Desativa o registo relativamente à operação pendente;
 - A subscrição do utilizador mantém-se no plano e/ou estado anterior.
- O utilizador volta a ter acesso a todas as funcionalidades.

- **Paypal**

A análise ao meio de pagamento paypal foi realizada com vista a uma integração futura com o S4L.

1. Upgrade

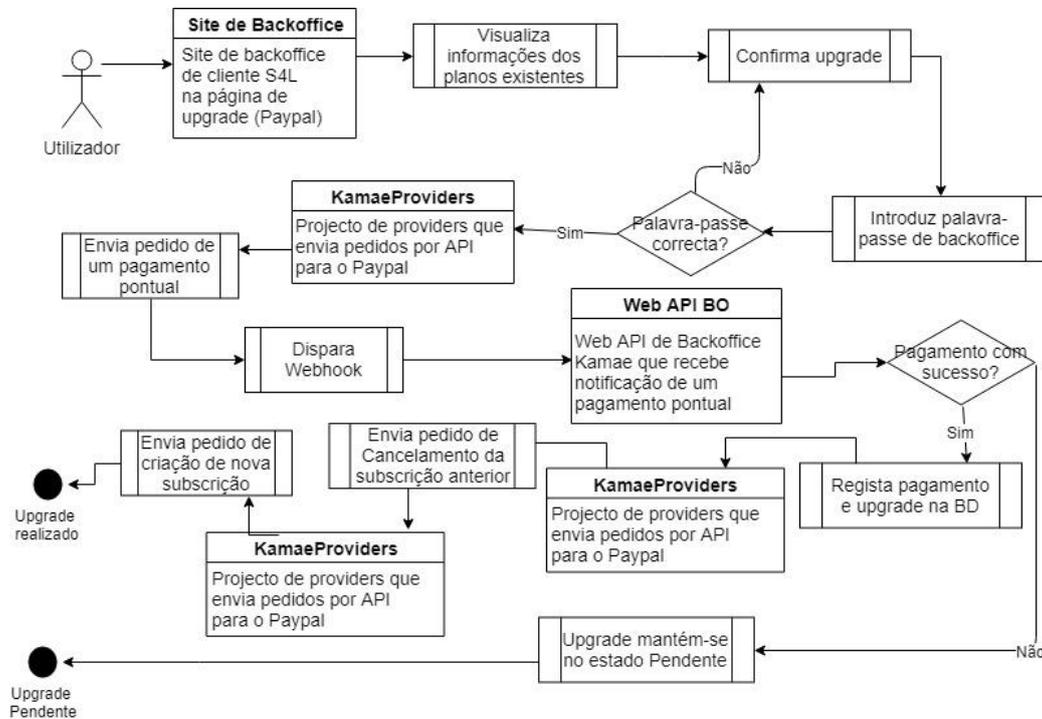


Figura F.6 - Upgrade com Paypal

O paypal não permite alterar o plano subscrito, o que leva à necessidade de cancelar o plano anterior e realizar uma nova subscrição do novo plano, para realizar a operação de upgrade.

Procedimento a implementar:

- É enviado um pedido ao Paypal para cobrar o valor da diferença entre o plano anterior e novo plano para os dias que ainda faltam para terminar o período já pago;
- A Web API de backoffice recebe uma notificação proveniente do webhook do paypal a indicar que foi realizado um pagamento:
 - O plano da subscrição é atualizado;
- É enviado um novo pedido para a API Paypal a cancelar o plano anterior;
- É enviado um pedido para a API Paypal a subscrever o novo plano;

Caso ocorra algum problema no processo de upgrade, é necessário reverter todas as alterações feitas, voltando ao plano anterior.

2. Downgrade

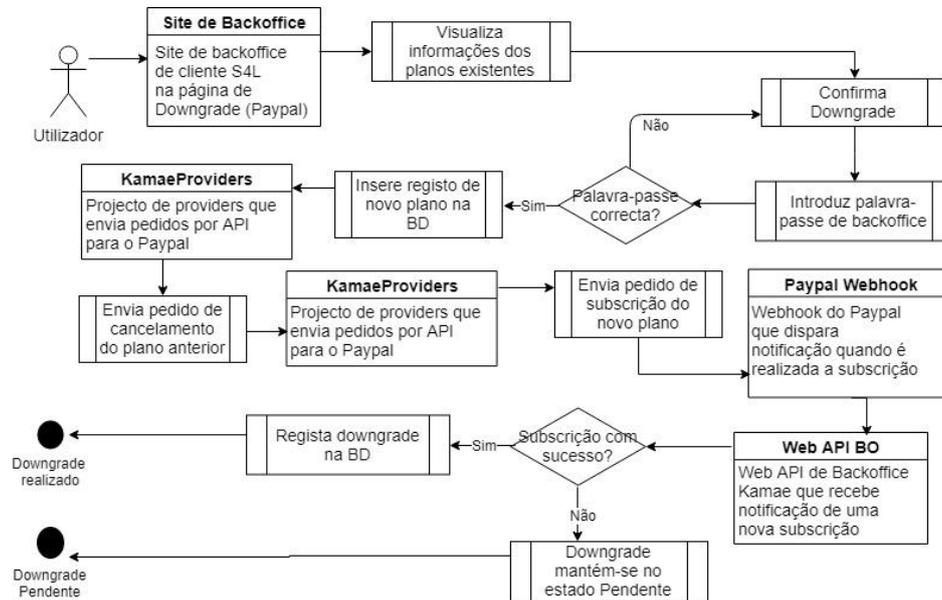


Figura F.7 - Downgrade com Paypal

Relativamente ao downgrade, funciona da mesma forma que o upgrade, havendo a necessidade de cancelar o plano anterior e subscrever o novo. Contudo, neste caso há um pequeno desafio que precisa de ser ultrapassado, pois não é possível definir a data em que o plano vai ser cancelado, sendo que o cancelamento tem efeito imediato, o que pode obrigar a um esforço adicional, caso haja necessidade de reverter esta operação.

Procedimento:

- É feito um pedido por API ao Paypal para cancelar o plano subscrito;
- É feito um pedido por API ao Paypal para subscrever o novo plano, indicando a data em que termina o período já pago como data de início desta nova subscrição;
- A Web API recebe uma notificação de atualização da subscrição:
 - Ativa-se o novo plano na BD, no entanto este só terá início na data em que será realizado o próximo pagamento;
- O downgrade fica pendente (podendo ser cancelado) até à data do próximo pagamento;

Caso ocorra algum erro ou algum problema no processo de downgrade, é necessário reverter a operação, voltando a mudar para o plano anterior.

3. Suspende

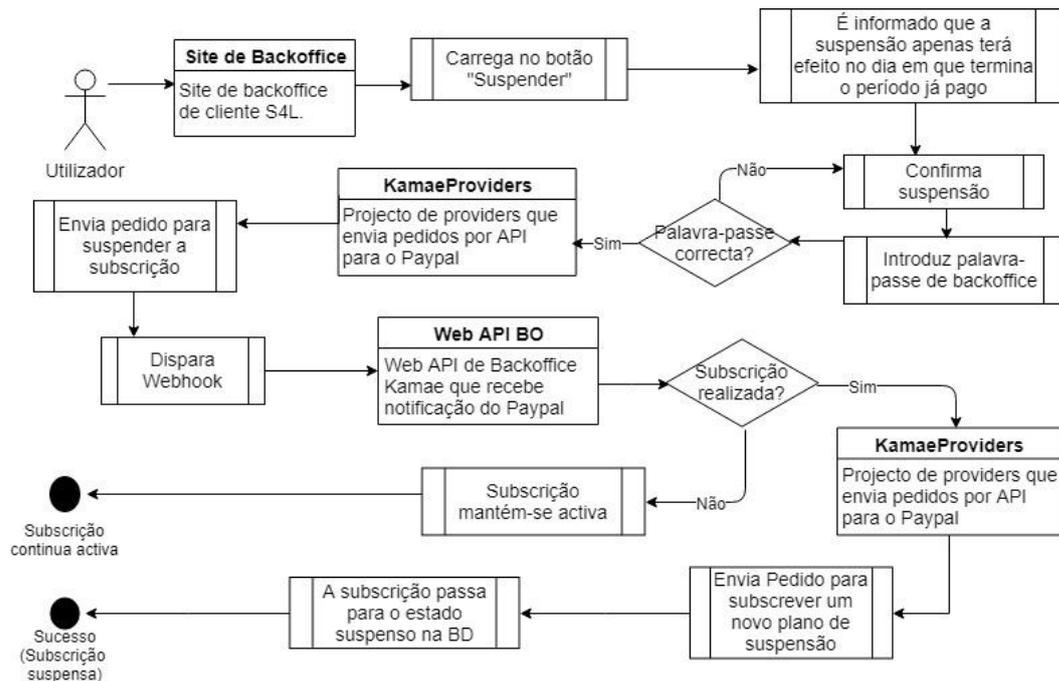


Figura F.8 - Suspende com Paypal

Ao contrário do que acontece com o Stripe, o Paypal permite suspender subscrições ativas. Contudo, apesar desta possibilidade, considerando que a KAMAE pretende que uma subscrição suspensa continue a pagar o valor de 1€, após suspender a subscrição, será necessário subscrever um novo plano, para garantir que o valor desejado será cobrado.

Procedimento a implementar:

- Após confirmação do desejo de suspensão, é enviado um pedido para o Paypal, com o objetivo de suspender a subscrição;
- A web API de BO recebe uma notificação, provinda do webhook do Paypal, a indicar que a subscrição foi suspensa
 - A subscrição passa para o estado suspenso, no entanto encontra-se ativa até ao final do período já pago;
- É feito um novo pedido à API, para subscrever um novo plano correspondente a uma subscrição suspensa;

Caso tenha ocorrido algum problema no processo de suspensão, é necessário reverter a operação, voltando ao reativar a subscrição.

4. Cancelar

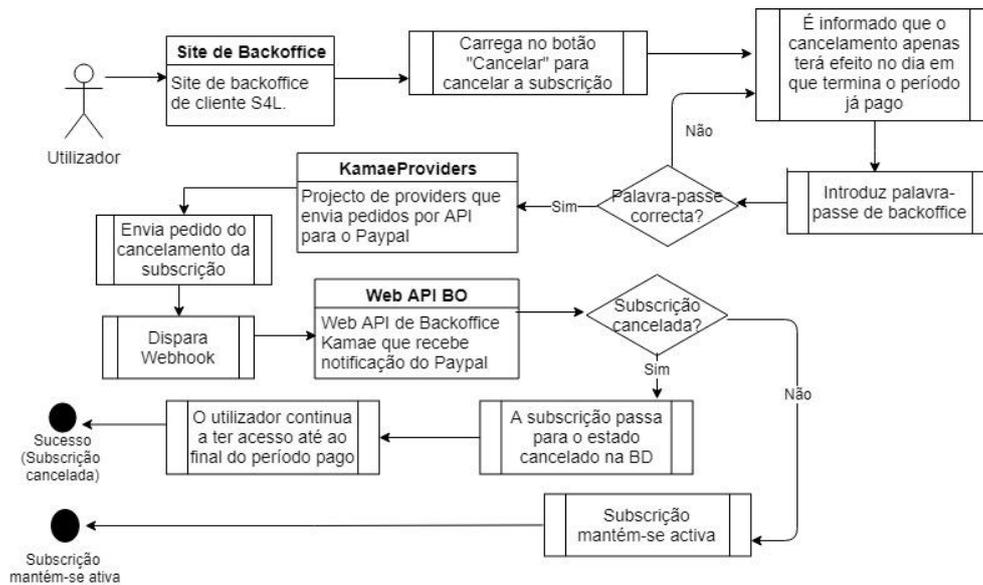


Figura F.9 - Cancelar com Paypal

No caso do cancelamento, o Paypal tem uma grande desvantagem em relação ao Stripe, pois não é possível cancelar apenas na data em que termina o período já pago, o cancelamento é imediato.

Procedimento:

- Após confirmação do desejo de cancelamento pelo utilizador, é enviado um pedido ao Paypal, por API, para que a subscrição seja cancelada;
- A Web API de BO recebe uma notificação de cancelamento da subscrição, proveniente do webhook do Paypal:
 - A subscrição é atualizada na BD, passando ao estado cancelada, ficando esta alteração de estado pendente até ao final do período já pago;
 - O utilizador deixa de pagar e deixa de ter acesso ao seu site S4L assim que terminar o período já pago.
- Caso a notificação indique que a subscrição não foi cancelada, a subscrição do utilizador mantém-se ativa.

5. Cancelar Operação pendente

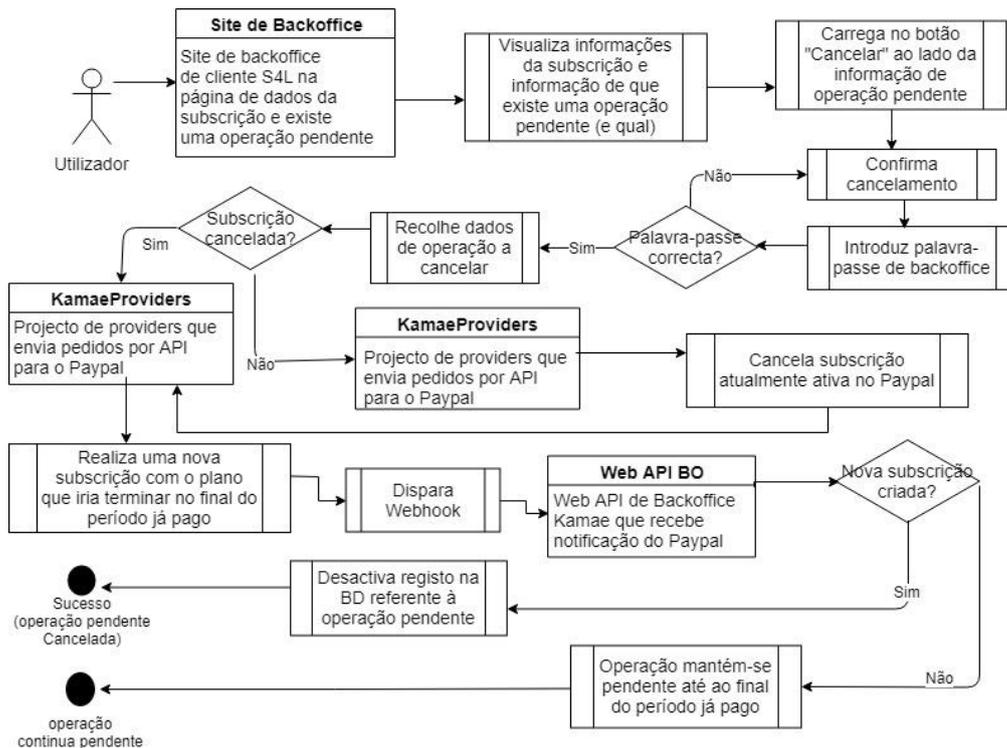


Figura F.10 - Cancelar operação pendente com Paypal

Procedimento:

- O utilizador quando entra no Backoffice tem menos funcionalidades disponíveis e quando entra na página de dados de subscrição é informado de que tem uma operação pendente e que enquanto não passar ao estado ativo não pode realizar outras alterações à sua subscrição;
- Quando confirma o desejo de cancelar a operação pendente, as informações relativas à operação são obtidas da BD:
 - Operação pendente é um cancelamento
 - É enviado um pedido ao Paypal, por API, para subscrever novamente o plano que irá terminar no final do período pago.
 - Outra operação:
 - É enviado um pedido por API para cancelar o plano atualmente subscreto;
 - É enviado um pedido ao Paypal, por API, para subscrever novamente o plano que irá terminar no final do período pago.
- A Web API de BO recebe uma notificação da nova subscrição:
 - Desativa o registo relativamente à operação pendente;
 - A subscrição do utilizador mantém-se no plano e/ou estado anterior.
- O utilizador volta a ter acesso a todas as funcionalidades.