



Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu



## **RESUMO**

O setor do turismo tem vindo a assinalar reconhecidas dinâmicas de crescimento ao longo dos últimos anos e, em conformidade com as estimativas da OMT (Organização Mundial do Turismo), prevê-se que este setor continue a crescer de forma contínua e sustentada até ao ano 2030. Em Portugal, os resultados apurados para o ano 2014 validam um total de cerca de 15 milhões de entradas de turistas internacionais e cerca de 10 mil milhões de receitas geradas.

Pretende-se com o presente projeto fornecer aos pequenos e médios fornecedores de serviços de experiências turísticas um conjunto de ferramentas que dar-lhes-ão a possibilidade de ter uma oferta online integrada, coerente, moderna e adaptável ao dinamismo que hoje se exige de um negócio que pretende singrar neste mercado, ou seja, ajustada ao perfil do novo consumidor.

Os mercados online estão hoje a ser transformados por novos modelos de negócio, em que se fornecem capacidades extraordinárias a pequenos players e a individuais. Os mercados da mobilidade door-to-door, com a Uber, do alojamento temporário, com o AirBnB, da distribuição de conteúdo musical, com o Spotify, da produção e comércio artesanal, com o Etsy, foram transformados por darem os meios e a capacidade de gerar modelos de negócio sustentáveis, autogeridos, a produtores e fornecedores de pequena dimensão. No mercado do turismo, o problema pode dividir-se, semanticamente, nos seus 3 principais momentos: pré-estadia, estadia e pós-estadia.

O presente projeto visa potenciar no primeiro momento, para operadores e fornecedores de serviços, criando modelos de negócio e ferramentas inovadoras para a gestão das suas interações e da comercialização. Em traços gerais, permitirá aos pequenos fornecedores a resolução dos seguintes problemas: visibilidade da sua oferta, criação de canais e gestão de reservas, publicidade e adequação ao target, comunicação e gestão de cliente.



## **ABSTRACT**

The tourism sector has been noted recognized growth dynamics over the past few years, according to the estimates of the WTO (World Tourism Organization), it is expected that this sector will continue to grow in a continuous and sustained way until the year 2030. In Portugal, the results reported for 2014 validate a total of about 15 million international tourist arrivals and about 10 billion revenue generated.

The aim of this project is provide for small and medium tour activities providers a set of tools that will give them the possibility of having an integrated online offering, coherent, modern and adaptable to the dynamism that today requires on business that want to succeed in this market, that is, adjusted to the new consumer profile.

The online markets are now being transformed by new business models, in which they provide extraordinary capabilities to small players and individual, then they can take advantage of strong presence of national or global brands that enhance their business. Door-to-door mobility markets, like Uber, temporary accommodation, like AirBnB, distribution of musical content, like Spotify, production and craft trade, like Etsy, they were transformed by giving the means and the ability to generate sustainable business models, self-managed, to producers and small providers. In the tourism market, the problem can be divided, semantically, in its three main stages: pre-stay, stay and post-stay.

This project aims to enhance in the first stage, to operators and service providers, creating business models and innovative tools for managing their interactions and marketing. In general terms, will enable small suppliers to solve the following problems: visibility of your offer, creating channels for booking management, advertising and suitability to the target, communication and customer management.



## **PALAVRAS CHAVE**

Turismo  
Experiências turísticas  
Microserviços  
Sistemas de Informação  
Sistema de ofertas turísticas





## **KEY WORDS**

Tourism  
Tour Activities  
Microservices  
Information Systems  
Tour Management System



## **AGRADECIMENTOS**

Gostaria de agradecer a todos os que acreditaram e contribuíram de forma direta ou indireta para concretização deste projeto. Desta forma, deixo o meu especial agradecimento ao Professor Doutor Paulo Tomé, pela sua colaboração e disponibilidade.

Agradeço ao Instituto Politécnico de Viseu e também à empresa Ubiwhere pela disponibilização de meios e informação essencial, e a todas as pessoas que dedicaram algum do seu tempo, em especial para o Bruno Silva por todo o conhecimento que me transmitiu.

À minha família e amigos, em especial á minha esposa por todo o apoio e paciência pela minha ausência.

A todos aqueles que, sabendo ou não, contribuíram para o meu crescimento e desenvolvimento enquanto pessoa, e para o crescimento da vontade de chegar mais longe.

A todos, muito obrigado!

Nuno Silva



# ÍNDICE GERAL

Índice Geral.....	xii
Índice de Figuras.....	xiv
Índice de Tabelas .....	xvi
Abreviaturas e Siglas .....	xviii
1. Introdução .....	1
1.1 Contextualização .....	1
1.2 Motivações .....	2
1.3 Objetivos .....	2
1.4 Metodologia de Investigação.....	3
1.5 Estrutura da Tese .....	4
2. Tecnologias e Plataformas existentes .....	6
2.1 Introdução.....	6
2.1.1 Avanço das tecnologias na área do turismo .....	7
2.2 Plataformas Existentes .....	10
2.2.1 Rezgo .....	10
2.2.2 Rezdy .....	11
2.2.3 Checkfront.....	12
2.2.4 ActivBookings .....	12
2.2.5 TourCMS .....	12
2.3 Tecnologias Utilizadas .....	13
2.3.1 Python .....	13
2.3.2 Django .....	13
2.3.3 Django Rest Framework .....	14
2.3.4 Swagger.....	14
3. Análise e Desenvolvimento .....	15
3.1 Análise.....	15
3.2 Desenvolvimento.....	16

3.2.1	Arquitetura lógica .....	16
4.	Serviços Implementados .....	23
4.1	Serviço de Itens .....	23
4.1.1	Categorias .....	23
4.1.2	Ofertas .....	23
4.1.3	Eventos .....	24
4.1.4	Idiomas .....	24
4.1.5	Comentários .....	24
4.2	Serviço Proxy de Ofertas Turísticas .....	24
4.3	Serviço de Calendários .....	25
4.4	Serviço de Utilizadores .....	26
4.5	Serviço de Reservas .....	27
4.6	Serviço de Multimédia .....	27
5.	Testes e Resultados obtidos .....	29
5.1	Testes Unitários e de Integração .....	29
5.2	Resultados Obtidos .....	30
6.	Conclusão e Trabalho Futuro .....	35
6.1	Trabalho Futuro .....	36
6.1.1	Desenvolvimento de marketplace para as entidades .....	36
6.1.2	Criação de serviço proxy para ofertas de hotéis .....	36
6.1.3	Criação de ferramentas de reserva e recomendação por canais sociais .....	36
	Referências .....	37
	Anexo 1 – Modelo de Dados do Serviço de Itens .....	40
	Anexo 2 – Modelo de Dados do Serviço de Utilizadores .....	41
	Anexo 3 – Modelo de Dados do Serviço de Reservas .....	42

## ÍNDICE DE FIGURAS

Figura 1 – Docker Containers vs. VMs.....	9
Figura 2 - Arquitetura geral da plataforma .....	16
Figura 3 - Arquitetura da camada Cliente .....	17
Figura 4 - Arquitetura da camada Proxy .....	18
Figura 5 - Arquitetura da camada Servidor.....	19
Figura 6 – Descrição do processo de Autenticação .....	27
Figura 7 – Endpoint para criação de categorias das ofertas .....	30
Figura 8 – Endpoint para criação de oferta .....	31
Figura 9 – Endpoint para remover e alterar uma oferta .....	32
Figura 10 – Página Inicial do site.....	33
Figura 11 – Área de gestão de ofertas.....	33
Figura 12 – Página de criação de novos produtos.....	34
Figura 13 – Modelo de Dados do Serviço de Itens .....	40
Figura 14 – Modelo de Dados do Serviço de Utilizadores .....	41
Figura 15 – Modelo de Dados do Serviço de Reservas .....	42





## **ÍNDICE DE TABELAS**

Tabela 1 – Tecnologias utilizadas nas plataformas abordadas .....	10
---	----



## ABREVIATURAS E SIGLAS

REST	Representational State Transfer
SGBD	Sistema de Gestão de Base de Dados
SQL	Structured Query Language
NoSQL	Not Only SQL
B2C	Business-To-Customer
B2B	Business-To- Business
API	Application Programming Interface
RFC	Request for Comments
CMS	Content Management System
DEV	Development
TEST	Tests
QA	Quality Assurance
PROD	Production
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
ORM	Object-Relational Mapping
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
MS	Microservice
VM	Virtual Machine





# **1. Introdução**

Neste capítulo pretende-se fazer uma abordagem geral, onde se descreve todas as informações necessárias para que o leitor se possa enquadrar plenamente no âmbito do trabalho. Serão também referenciados pontos relevantes para a avaliação e validação do trabalho desenvolvido, como os objetivos a alcançar e as contribuições que se esperam obter. Por fim, irá ser fornecida uma visão sobre a estrutura deste documento e o seu significado.

## **1.1 Contextualização**

O setor do turismo tem 13% do número de empresas em Portugal, 5% do volume de negócios global e 10% do número de pessoas ao serviço em sociedades não financeiras. Na última década o peso de cada um dos indicadores acima referidos aumentou, em especial o número de pessoas ao serviço, que teve um aumento em dois pontos percentuais. O volume de negócios em 2013 teve um crescimento de 3%, que é resultado da inversão do comportamento da procura interna. No entanto, metade das empresas deste setor ainda apresentam resultados operacionais negativos, resultantes da forte dependência de capital alheio, em especial de empréstimos bancários (Menezes, 2014).

Apesar deste setor ter cada vez mais peso na economia portuguesa, urge a necessidade de criar inovações que estimulem o crescimento sustentável deste setor, promovendo a capacidade de desenvolver, ou pelo menos, requalificar as atividades e infraestruturas existentes.

### 1.2 Motivações

Perante os fatores apresentados anteriormente o que motivou este projeto foi a análise das vantagens para o setor do turismo, uma vez que as necessidades identificadas são cada vez mais preponderantes quer para os turistas como para as empresas do setor.

Do ponto de vista do turista e, portanto, do mercado, este projeto visa criar meios e mecanismos mais convenientes, confiáveis e transparentes para que estes efetuem a reserva das suas experiências turísticas. Pretende-se promover ainda a criação de novos mecanismos de comunicação e conhecimento contínuo entre promotores e turistas, através destes mesmos canais.

Se a reserva online é já um processo relativamente simples de executar tecnologicamente, assiste-se hoje em dia à sua transformação, com o alargamento dos canais de compra às plataformas sociais e de mensagens instantâneas. Assiste-se ainda ao surgimento de novos métodos de pagamento e autorização, distribuídos, baseados nos princípios da segurança distribuída das transações e em moedas completamente digitais, como o Bitcoin, e que potenciam a troca de serviços online. Estes ramos de investigação são particularmente relevantes se considerarmos uma geração de compradores cada vez mais jovem, com maior poder de compra e capacidade de mobilidade temporária, o que se reflete no perfil rejuvenescido e tecnologicamente apto dos visitantes atuais e futuros.

### 1.3 Objetivos

De acordo com as necessidades acima identificadas, o projeto visa criar ferramentas que exponenciam a capacidade, competitividade e escalabilidade dos negócios dos Agentes de Animação Turística. Desta forma, identificam-se os seguintes objetivos:

- Estudo de critérios tecnológicos que os Agentes de Animação Turística devem seguir para distribuir os seus produtos

Pretende-se com o atual projeto criar uma plataforma de distribuição atrativa globalmente, ainda que focada no fornecimento de experiências turísticas de base regional/local. Para o efeito, propõe-se encetar um conjunto de requisitos e critérios basilares que configurem a singularidade das experiências turísticas a estruturar, bem como, garantir que as mesmas sejam distintivas face às demais ofertas disponíveis no mercado. Em síntese, a plataforma a desenvolver que integrará as ofertas turísticas customizadas à medida, definirá quais os Agentes de Animação Turística que poderão distribuir os seus produtos (experiências turísticas singulares) utilizando a plataforma de distribuição digital. Pretende-se que esta seja

parte da oferta de soluções tecnológicas dos copromotores para entidades ligadas ao Turismo, que assim façam parte da rede criada para potenciar o comércio de experiências locais.

- Criação de ferramentas de agregação e disponibilização de ofertas

Utilização de standards da indústria para pesquisa, agregação e disponibilização da oferta em canais de distribuição externos, com vista à extensão e expansão da visibilidade dos promotores (Agentes de Animação Turística) e de cada experiência turística em concreto, permitindo uma progressiva monitorização dos perfis de ofertas que geram maiores dinâmicas de procura. A ambição da plataforma passará invariavelmente pela capacidade de disponibilizar as experiências turísticas junto dos mercados externos, diretamente ao consumidor (B2C) e através de outros players da distribuição turística (B2B).

Promover a disponibilização de pacotes de visitas integrados e conjugados com ofertas distintas do mesmo promotor. Desta forma fornece-se a capacidade de associar vários serviços relacionados ou não entre si, com possibilidade de potenciar o negócio nas diferentes áreas de ação.

- Criação de ferramentas de reserva e recomendação por canais sociais

Disponibilização imediata de canais de reserva com base em mecanismos inovadores, como interações em redes sociais ou plataformas de messaging, assim viabilizando as transações em novos meios de interação e em tempo real. A facilidade de reserva/ compra de experiências turísticas previamente ou no destino propriamente dito contribui de forma decisiva para aumentar as permanências médias no destino, melhorar os índices de satisfação e, por consequência, exponenciar o potencial de recomendação que se gera através das plataformas digitais (redes sociais).

## **1.4 Metodologia de Investigação**

Os métodos e as técnicas de investigação têm como objetivo o estudo com base nos métodos, sustentados por referentes teóricos, que enquadram o plano de trabalho de investigação e sugerem os procedimentos de recolha e tratamento de informação consistentes com a própria investigação.

Tendo em conta que o teor deste projeto é mais técnico do que teórico, como metodologia de investigação será utilizada a pesquisa do género exploratória, explicativa, bibliográfica e documental:



- Tendo em conta o estudo dos objetivos, a pesquisa será uma pesquisa exploratória onde se vai tentar uma familiarização mais próxima com o que existe atualmente e tentar construir hipóteses das melhores maneiras de extrair e categorizar os vários aspetos do estudo. Será também uma pesquisa explicativa onde se vai apresentar argumentos de sistemas e ambientes existentes e tentar explicar.
- Do ponto de vista dos procedimentos técnicos, será uma pesquisa bibliográfica e documental, onde se vai recorrer a material já publicado e material que ainda não recebeu tratamento analítico.

### 1.5 Estrutura da Tese

Neste subcapítulo descreve-se a organização deste documento e sintetizam-se cada um dos 6 capítulos que constituem este documento.

Neste primeiro capítulo, intitulado de “**Introdução**”, é inicialmente fornecido ao leitor uma contextualização do desenvolvimento da área do turismo em Portugal. De seguida, é possível aferir quais os objetivos que se pretendem alcançar com este trabalho e quais as motivações pessoais e académicas que o sustentam. Termina com a estrutura da tese, que tem como objetivo orientar a leitura deste documento, conforme os interesses de diferentes tipos de leitores.

No segundo capítulo, designado de “**Tecnologias e Plataformas existentes**”, são analisadas as principais tendências tanto a nível das tecnologias como dos sistemas existentes. A continuação é feita através da enumeração de características particulares na área das novas tecnologias. Também é realizada uma comparação dos vários softwares disponíveis no mercado. O capítulo encerra com alguns aspetos tecnológicos, métodos e ferramentas utilizadas para o desenvolvimento do projeto.

O terceiro capítulo, denominado de “**Desenvolvimento**”, aborda-se a análise realizada aos conceitos e componentes utilizados na arquitetura do sistema.

O quarto capítulo, intitulado de “**Serviços Implementados**”, é constituído pelos detalhes e características de cada serviço que compõem todo o sistema desenvolvido. Também é abordado alguns desafios que foram surgindo durante o desenvolvimento e definição de cada serviço, assim como a justificação de algumas escolhas mais importantes nos módulos de cada serviço.

No quinto capítulo, intitulado de “**Testes e Resultados Obtidos**”, descreve-se os vários tipos de testes que foram abordados durante o desenvolvimento. É também avaliado o impacto do desenvolvimento perante os objetivos delineados inicialmente.

O sexto e ultimo capítulo, denominado de “**Conclusão e Trabalho Futuro**”, é feito uma reflexão sobre o trabalho feito sobre como este projeto vai evoluir no futuro.

## 2. Tecnologias e Plataformas existentes

O aparecimento da Internet fez surgir novas oportunidades que permitiram o desenvolvimento de um canal direto entre os clientes e os fornecedores. No início dos anos 2000, com o aumento do uso de internet para consulta de destinos turísticos começaram a surgir os primeiros sistemas de reserva turística. Atualmente, existem várias plataformas para procura e reserva de atrações turísticas.

Este capítulo resume o estado da arte de plataformas de reserva turística e das tendências tecnológicas para esta área de negócio, que tem vindo a evoluir cada vez mais.

### 2.1 Introdução

Tecnologicamente, o mercado das experiências turísticas ainda está numa fase ascendente da chamada *sharing economy*. Com *players* por todo o mundo, não existem claros vencedores globais ou mesmo internacionais, que tenham encontrado viabilidade e sustentabilidade no seu modelo de negócio. No entanto o modelo mais comum é a disponibilização de vários tipos de ofertas, possibilitando a reserva e o pagamento online. Com maior viabilidade na redução de custos em termos de exigência de mão de obra em escritórios focados na reserva das experiências turísticas.

As tecnologias de informação têm desempenhado um papel importante no crescimento e desenvolvimento da indústria do turismo em geral. Nos primeiros anos de disponibilização de turismo em massa (de 1950 e 1970), os sistemas de informação foram principalmente utilizados como forma de apoio nos setores dos transportes, hotelaria e serviços de alimentação (Sezgin & Yolal, 2007).

Atualmente, as plataformas através da Internet tornaram-se uma das ferramentas de comunicação mais importantes para turistas, assim como as inovações tecnológicas tem tido um papel preponderante na estrutura da indústria do turismo. O turismo é um negócio baseado em informação, estando muitas vezes dependente da forma como é disponibilizada. Os produtos turísticos são fornecidos á base de “confiança”, ou seja, uma avaliação prévia abrangente de suas qualidades. Esta avaliação é tomada a partir de um modelo abstrato do produto, com base em informações obtidas através de múltiplos meios, como a televisão, folhetos, ou na Web (Werthner & Ricci, 2004).

Nesta era da informação, existe uma crença generalizada de que a eficácia organizacional está intimamente relacionada com a capacidade da organização para usar a informação oportuna, precisa e relevante na tomada de decisões de negócios. As exigências obrigam à integração de sistemas de informação, facilitando a interatividade com o mercado. Desta forma, ajuda a aumentar a satisfação do consumidor, tendo fácil acesso a mais informações, e uma maior escolha. A redução de burocracias liberta eficazmente o tempo de atendimento, a prestação de novos serviços e facilita a integração de tarefas operacionais e serviços personalizados.

### 2.1.1 Avanço das tecnologias na área do turismo

Os avanços das tecnologias são, indiscutivelmente, o fator de mudança mais importante na área do turismo. Destacam-se algumas tecnologias que mais contribuíram para o avanço multidimensional nesta área, entre elas distinguem-se a **Capacidade de Computação**, **Base de Dados**, o **Uso de Padrões**, o **Registo de eventos para repetição**, a **Facilidade e Flexibilidade de Deploy**.

Atualmente observa-se que maior parte das aplicações exigem cada vez mais uma grande quantidade de processamento de dados, aplicações de mapeamento genético, computação gráfica, previsões metrológicas e aplicações que funcionam como serviços web (API). Para tentar solucionar essa exigência, foram criados supercomputadores. No entanto, nem sempre a sua utilização é viável, devido aos elevados custos de aquisição e manutenção que a sua utilização implica. Uma alternativa à sua utilização e que permite concentrar um grande poder computacional com custos inferiores, é a utilização de clusters (B´TEN, 2014).

A computação em cluster mostra-se maior parte das vezes como uma solução viável porque os nós podem até mesmo ser compostos por computadores simples, como PC’s de desempenho mediano. Juntos, eles configuram um sistema de processamento com capacidade suficiente para dar conta de determinadas aplicações que, se fossem atendidas por supercomputadores ou servidores sofisticados, exigiriam investimentos muito maiores. Para o utilizador a aplicação de cluster aparenta ser um único sistema, porque o processamento de múltiplas tarefas em simultâneo é efetuado de forma transparente.

Com a necessidade de grandes quantidades de processamento de dados por parte de grandes empresas web, surgiu o aparecimento do NoSQL que está relacionado com o crescimento de projetos criados pela Google(BigTable), Facebook(Cassandra) e Amazon(Dynamo) (Gomes, 2011).

As ultimas tendências na área dos sistemas de gestão de base de dados, baseiam-se maioritariamente na mudança da tradicional Structured Query Language (SQL) para o alto desempenho do NoSQL (not SQL). O NoSQL abrange todas as bases de dados que não seguem os princípios das bases de dados relacionais. Não é apenas uma tecnologia, mas sim um conjunto de tecnologias e diversos conceitos de armazenamento e manipulação de dados.

O uso de padrões é visto como uma forma de distribuição de conhecimento, uma maneira de aumentar a reutilização de boas práticas e como uma forma de facilitar a comunicação sobre as soluções existentes.

Dois organismos de normalização, Open Travel Alliance e Open AXIS, têm vindo a desenvolver padrões de XML com o objetivo de suportar o acesso a conteúdo distribuído (Street, 2000). O foco principal destes organismos é a criação de estruturas de mensagens eletrônicas de modo a facilitar a comunicação entre sistemas heterogêneos na indústria global do turismo.

Estes padrões facilitam a construção de sistemas de software em que a sua finalidade seja trabalhar em conjunto com outros sistemas. Por outro lado, também ajudam analistas a especificar outros sistemas semelhantes para a indústria do turismo.

A definição de eventos para repetição é um cenário bastante normal na gestão de atividades turísticas. No entanto, existe um grau elevado de complexidade por de trás desta funcionalidade que implica a resolução de datas e horas num determinado período de tempo.

A chave que permite absorver toda esta complexidade é determinado como regras recursivas (RRULEs), que consiste num formato standard definido por RFC 2445, section 4.8.5.4. Este formato encapsula uma ou várias regras (p.e. “todas as Terças-feiras às 10h”). Estas regras são processadas e por fim é retornado uma lista com os vários timeslots (Jennie Lees, 2015).

Por outro lado, também permite adicionar exceções em conjunto com as regras, por exemplo, supondo que é pretendido um evento que ocorra á segunda-feira e terça-feira às 10h durante o período de um mês, existe uma segunda-feira nesse período que não é pretendido haver esse evento, então adiciona-se a exceção para esse dia em específico.

Com o crescimento da capacidade de computação em nuvem e a virtualização, tem havido um numero crescente de máquinas que um administrador de sistemas precisa de gerir. A

facilidade de deploy e gestão de máquinas é um aspeto que tem vindo a ser cada vez mais importante no processo de desenvolvimento de software. É importante que o nível de complexidade seja o mais baixo possível no momento de se efetuar a instalação e gestão de uma plataforma, e que esta não se torne um pesadelo cada vez que se pretende fazer uma nova instalação. Com esta necessidade foi abordado uma plataforma de virtualização de ambientes chamada Docker.

O foco principal de Docker é automatizar o processo de deployment de aplicações dentro de containers. Os containers de aplicação são definidos de forma a empacotar e correr um simples processo, enquanto que os containers de sistema são definidos para correr vários processos, semelhante às máquinas virtuais. Com Docker as aplicações ficam completamente independentes e portáteis, isto é, não necessitam de quais queres configurações adicionais ou outras aplicações externas para correr. É possível mover um container de um host para outro e o Docker garante que vai funcionar da mesma forma (Steven J. Vaughan-Nichols, 2014).

Ao contrário de uma máquina virtual, um container não precisa arrancar um kernel de sistema, logo os containers podem ser criados em menos de um segundo. Esta capacidade é uma mais valia em relação a outras formas de virtualização. Numa máquina física é possível conter inúmeros containers a correr de forma isolada e independente. É ideal para sistemas que são sujeitos às várias fases de desenvolvimento (dev / test / qa / prod).

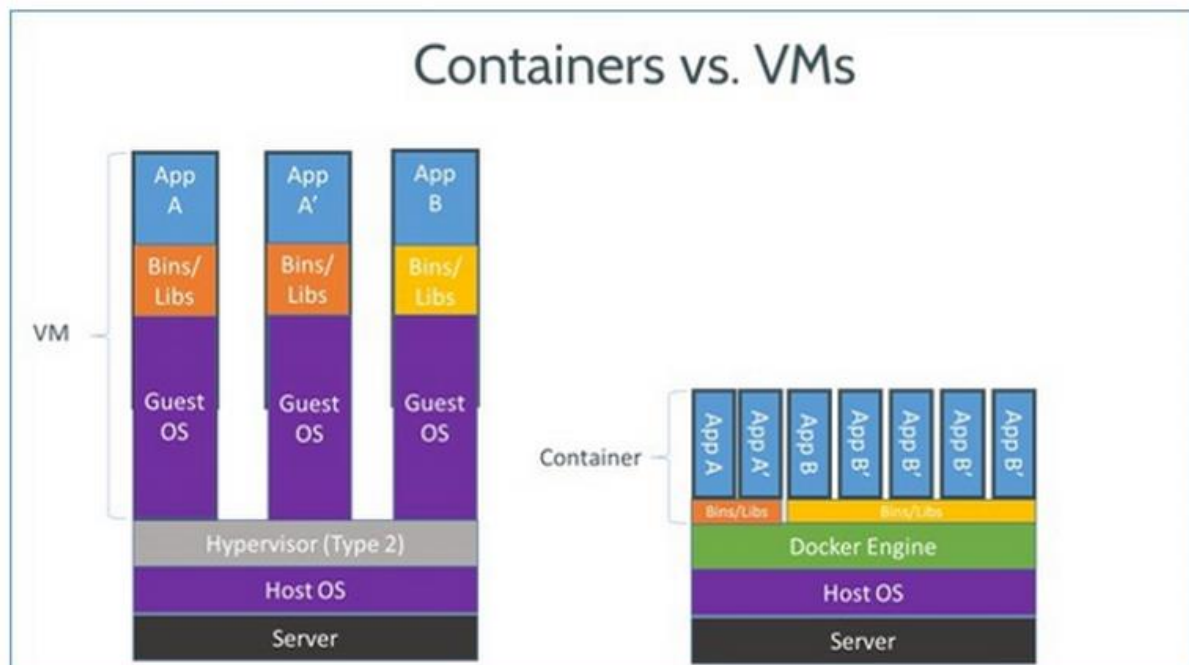


Figura 1 – Docker Containers vs. VMs

O crescimento da ferramenta Docker tem sido bastante acentuado, assim como de toda a sua comunidade. Cada vez mais tem sido utilizado no processo de desenvolvimento de software pelas empresas.

## 2.2 Plataformas Existentes

Este subcapítulo representa um dos aspetos mais relevantes para o desenvolvimento de um produto. Será feita uma análise de soluções próximas ao que se pretende. Com esta análise é possível definir características essenciais, pontos estratégicos a melhorar, entre outros atributos que poderão acrescentar valor no final. Cada sistema enumerado a baixo é apresentado com conjunto de características.

Como resultado de um trabalho de pesquisa, foram encontrados vários sistemas que tem com objetivo facilitar a gestão e disponibilização de ofertas turísticas. De uma lista bastante abrangente foram selecionadas algumas plataformas mais interessantes e que mais se enquadravam com este projeto: **Rezgo**, **Rezdy**, **Checkfront**, **ActivBookings** e o **TourCMS**. Maior parte destes sistemas disponibilizam uma API B2B com uma documentação bastante composta.

Component \ System	Rezdy	Rezgo	TourCMS	Checkfront
Architecture	-	Monolithic	-	-
API	REST	REST	REST	REST
Response Formatting	JSON / XML	XML	XML	JSON
Authentication	API Key	API Key	-	OAuth2

Tabela 1 – Tecnologias utilizadas nas plataformas abordadas

No quadro acima refere-se uma comparação de algumas tecnologias abordadas por cada um dos sistemas referenciados.

### 2.2.1 Rezgo

O Rezgo é considerado uma das maiores e mais maduras plataformas de reserva de atividades turísticas. Esta plataforma tem sido desenvolvida com base numa comunidade de operadores turísticos que partilham o seu conhecimento e experiência. Comunidade essa já referenciada anteriormente, o Open Travel Alliance (Stephen Joyce, 2011). Em 2006 foi lançada a primeira versão lançada, ainda beta.

A plataforma Rezgo é especialmente designada a operadores turísticos. Permite aos operadores vender as suas atividades através da Internet e receber as reservas em tempo real. É composto por uma plataforma de gestão, onde os operadores podem realizar todo o seu

inventário de atividades. Permite vários tipos de utilizadores administrativos. Disponibiliza vários widgets javascript para fácil integração em vários sites. Permite a integração com várias gateways de pagamento.

Os aspetos mais interessantes desta plataforma consistem na facilidade com que se realiza tarefas que de outra forma o custo de realização seria elevado. O registo de toda atividade dos utilizadores, a forma simples com que se consegue seguir a ocupação dos espaços numa atividade e a criação de regras, como as horas de abertura, mínimo e máximo de lugares disponíveis, tipos de preços, descontos e ofertas especiais, ou quando é necessário definir preços diferentes durante um dia da semana ou no fim de semana.

### **2.2.2 Rezdy**

O Rezdy é um sistema B2B de reserva e gestão de atividades para operadores turísticos. O sistema é composto por uma API muito bem documentada, e também por uma plataforma que permite aos operadores realizarem toda a gestão relacionada com as atividades. Em 2012 foi lançada a primeira versão (Chris Motola, 2015).

A plataforma de gestão é composta por uma interface muito simples e apelativa, onde não falha nenhum detalhe ou opção para o utilizador. Cada componente torna a experiência de utilização muito fácil. Permite selecionar uma lista de ações mais prioritárias, em que se encontram facilmente acessíveis ao utilizador, facilitando a sua interação com os vários componentes do sistema.

Para uma melhor definição e gestão de atividades a plataforma Rezdy tem como componente um calendário que permite facilmente manipular, ver e filtrar sessões de várias formas. Cada atividade listada é composta por uma cor própria, onde permite ver detalhes da atividade como a disponibilidade, os preços ou o produto. Selecionando um determinado período de tempo no calendário é possível criar uma nova sessão para uma atividade.

Para reserva das atividades o foco principal do Rezdy é a integração com sites existentes ou páginas do Facebook. Widgets com listagem das atividades, apenas contento uma imagem e uma pequena descrição com um link para o detalhe completo. O processo de reserva definido é muito simples: os clientes escolhem a atividade, a data, o numero de reservas e se pretendem oferecer como presente.

Outros aspetos importantes nesta plataforma a referenciar são: a gestão de conteúdos em várias línguas, facilidade em adicionar cupões de oferta e vouchers e a integração com várias gateways de pagamento.



### **2.2.3 Checkfront**

Checkfront é um sistema de reservas online focado em atividades turísticas e hotéis. Em 2009 foi lançada a primeira versão. Conforme os outros sistemas, este também é composto por uma plataforma de gestão de ofertas. Integração com sites existentes e CMS's através de plugins é um dos focos principais deste sistema. Com estes plugins é possível adicionar de forma simples um componente para a reserva de uma oferta, seja ela uma atividade ou um hotel (Chris Motola, 2016).

Uma particularidade interessante no calendário desta plataforma é a existência de um calendário para os clientes e outro para as ofertas. Com isto, é possível verificar todas as reservas de um determinado cliente, tendo uma visão geral sobre os eventos que irão ocorrer. Outros aspetos que se destacam nesta plataforma são: incrementação / decrementação dinâmica do preço de uma oferta para um determinado período de tempo (p.e. subida de preço 20% no mês de dezembro), ativação de itens apenas num determinado período de tempo, definição de regras e políticas de reserva para um período de tempo específico (p.e. definir quantidade mínima de 2 em todas as sextas do mês de maio), mapa da localização dos clientes, permite analisar as zonas onde são feitas as reservas e por fim a gestão dos conteúdos em várias línguas.

### **2.2.4 ActivBookings**

O ActivBookings é um portal online português que agrupa ofertas de operadores que fornecem experiências turísticas. Disponibiliza serviço para operadores (animação turística) e agentes (pontos de venda), sendo que nos últimos se incluem hotéis, agências de viagens, postos de aluguer de automóveis, postos de informação ou lojas.

Os aspetos interessantes destacam-se como a possibilidade de visualizar oferta disponível num mapa, a apresentação do tempo mínimo requerido para efetuar uma reserva e a diferenciação de acordo com o tipo de experiência (programada, etc).

### **2.2.5 TourCMS**

O TourCMS é uma plataforma web para operadores turísticos que permite aos operadores gerir ofertas, datas, preços e disponibilidades. Desenvolvido sobre um sistema de reservas existente utilizado por mais de 60 operadores especialistas em atividades turísticas (Kevin May, 2009).

Os aspetos menos positivos desta plataforma consideram-se ao nível da interface do utilizador. A interface é bastante podre em relação aos outros sistemas referenciados anteriormente, contento alguns erros de usabilidade e de funcionalidades perante uma análise muito por alto aos componentes presentes.

## 2.3 Tecnologias Utilizadas

Para desenvolvimento deste projeto foram definidas tecnologias open-source, com base no índice de experiencia em cada uma das seguintes tecnologias.

### 2.3.1 Python

Para desenvolvimento do sistema em termos tecnológicos optou-se por utilizar Python como linguagem de programação. Python é uma linguagem incrivelmente poderosa, é utilizada por empresas como Google, Dropbox, Spotify e Netflix. É possível verificar que Python é utilizado numa grande variedade de aplicações. Existem uma grande comunidade open source, em que vários developers contribuem diariamente criando novas bibliotecas. Estas bibliotecas são muito importantes para que seja possível reaproveitar nos projetos em desenvolvimento e não haver a necessidade de fazer algo que já foi feito por alguém, ou seja, “reinventar a roda”.

### 2.3.2 Django

Como framework de desenvolvimento utilizou-se Django, uma framework que permite fazer aplicações complicadas de uma forma simples e prática. Django tem bastantes pontos vantajosos em relação a outras frameworks mais comuns, no entanto optou-se por enumerar três pontos cruciais desta poderosa framework:

- **Django ORM:** O ORM é uma poderosa ferramenta que permite a criação da base de dados e a execução de queries, baseando-se apenas em modelos definidos como objects.
- **Django Admin:** O painel de administração do Django é muito mais que um pedaço de código gerado a partir de um modelo, permite fazer múltiplas customizações e sem grandes esforços pode ser adaptado como painel de gestão de um website.
- **Comunidade:** Assim como já foi referenciado que a partilha de bibliotecas é um dos aspetos fundamentais de Python, em Django ainda se aplica mais este fator. Django oferece o número enorme de bibliotecas disponíveis, com uma boa documentação e de fácil integração nos projetos em desenvolvimento.

### **2.3.3 Django Rest Framework**

A Django Rest Framework permite escrever apis REST de forma rápida e flexível, tirando proveito dos mais novos recursos do ecossistema Django. Web browseable API, OAuth1/OAuth2, api de serialização, altamente customizável, ótima documentação são algumas das características do Django REST framework. Com todas essas funções, tornou-se uma das principais frameworks REST da comunidade Django.

### **2.3.4 Swagger**

A Framework Swagger permite numa API REST visualizar a documentação, de forma a facilitar a sua interpretação. Toda a documentação é gerada com base no código da API, e permite fazer customizações em toda a interface. Demonstra uma interface simples e apelativa, em que permite interagir com a API e fornecer aos utilizadores uma ideia geral de como a API responde aos pedidos com vários parâmetros e opções.

## **3. Análise e Desenvolvimento**

Neste capítulo vai ser detalhado a análise efetuada para o desenvolvimento do projeto, incluindo a arquitetura do sistema. Serão apresentados também alguns aspetos tecnológicos, métodos e ferramentas utilizadas para o desenvolvimento.

### **3.1 Análise**

Devido à complexidade do sistema que se pretende desenvolver optou-se por fazer uma análise focalizada na separação em módulos, de forma a facilitar o isolamento e reutilização dos componentes.

Na indústria de desenvolvimento de software é bastante comum os sistemas desenvolvidos serem baseados numa arquitetura monolítica. Nesta arquitetura, são desenvolvidos, testados, empacotados e disponibilizados como um serviço único. No entanto, internamente estes sistemas são compostos por vários serviços, componentes, etc.

Nos sistemas monolíticos quanto maior a sua complexidade mais difícil para se compreender o funcionamento, por sua vez se torna bastante complicado fazer qualquer tipo de alterações. Também neste tipo de aplicações é considerado um obstáculo adicionar novos elementos à equipa de desenvolvimento, devido ao elevado custo até conseguir integrar numa equipa e perceber como funciona uma aplicação nesta arquitetura.

Após uma análise às arquiteturas de sistemas desenvolvidos atualmente, chegou-se a um conceito. Conceito conhecido por Microserviços, tem sido bastante aprofundado na arquitetura de novas aplicações. Este conceito está gradualmente a mudar a forma como são

desenvolvidas aplicações web (Shahir Daya Nguyen Van Duy, Kameswara Eati, Carlos M Ferreira, Dejan Glozic, Vasfi Gucer, Manav Gupta, Sunil Joshi, Valerie Lampkin, Marcelo martins, Shishir Narain, 2015).

## 3.2 Desenvolvimento

O presente subcapítulo tem como objetivo apresentar os caminhos percorridos para chegar aos objetivos propostos, e também apresentar todas as especificações técnicas necessárias para a compreensão de toda a conceção do sistema.

### 3.2.1 Arquitetura lógica

A arquitetura lógica do sistema, baseia-se numa arquitetura constituída por 3 camadas, com separação de privilégios. A primeira camada, a camada Cliente, inclui os dois módulos de frontend, um destinado às entidades e o outro destinado ao cliente final, assim como os clientes B2B e B2C. Esta camada comunica com a camada Proxy, que expõe um conjunto de API REST, que dão acesso aos recursos e funcionalidades disponibilizadas pela camada Servidor. Esta arquitetura é representada na figura 1.

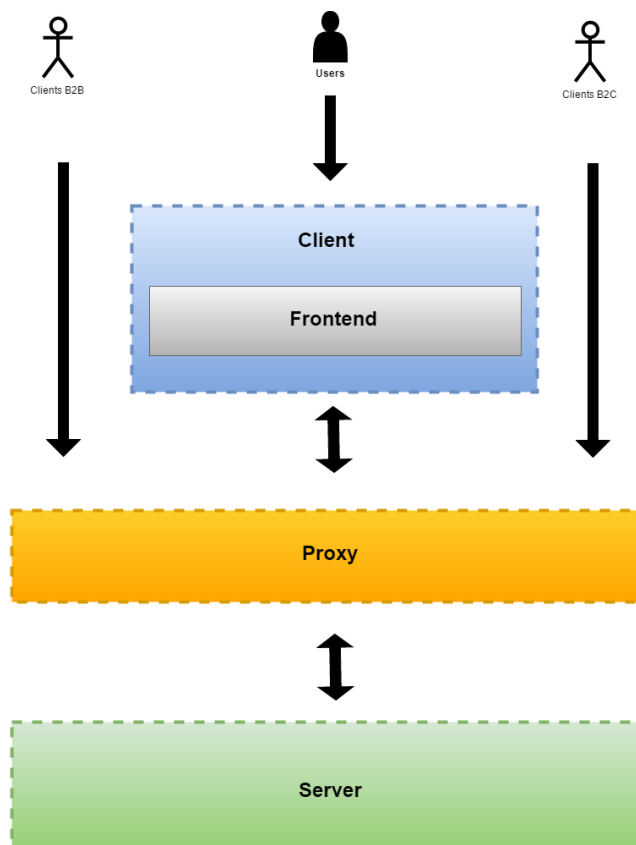


Figura 2 - Arquitetura geral da plataforma

A camada do Cliente inclui os dois módulos de frontend, implementados em AngularJS, os clientes B2C e todos os clientes B2B, que podem incluir plataformas externas que se alimentem dos dados produzidos por este serviço. Os componentes desta camada acedem às funcionalidades da plataforma através da camada Proxy. A arquitetura da camada Cliente é representada na figura 2.

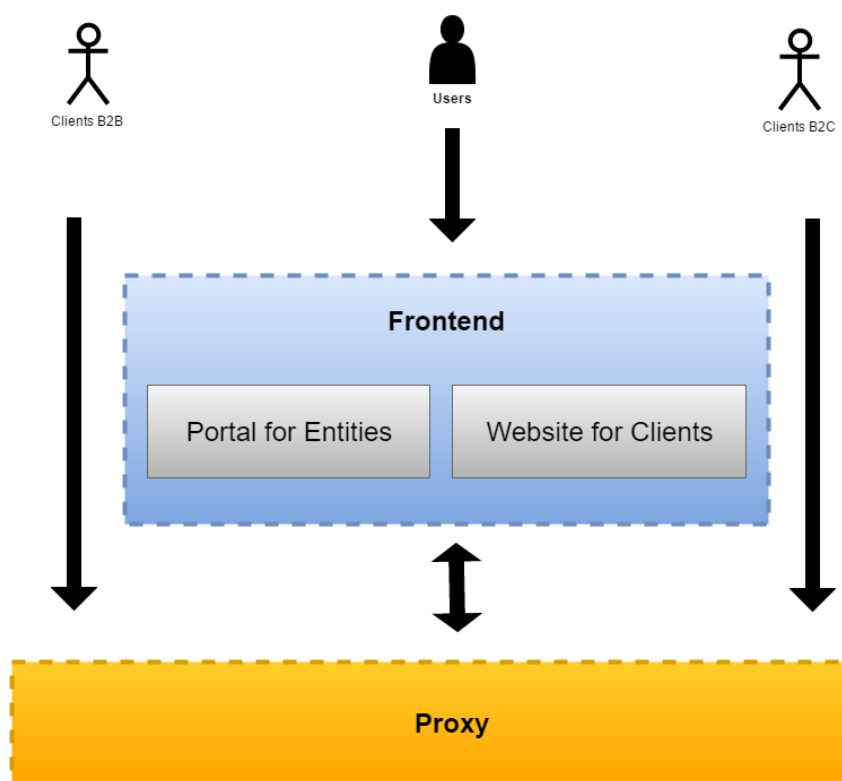


Figura 3 - Arquitetura da camada Cliente

A camada Proxy corresponde à camada intermédia entre as camadas Cliente e Servidor, e serve de canal de comunicação entre ambas. A necessidade desta camada intermédia, deve-se, primeiramente, ao fato da plataforma se destinar a várias finalidades distintas, como servir clientes B2B, clientes B2C ou os módulos de frontend. Cada um destes clientes requer acesso a diferentes funcionalidades e recursos, e a exposição de uma API única, indiferenciada e monolítica, resultaria numa interface de comunicação rígida e pouco conveniente para os consumidores da API. Com esta camada intermédia, é possível criar pontos específicos de comunicação, à medida das necessidades dos clientes.

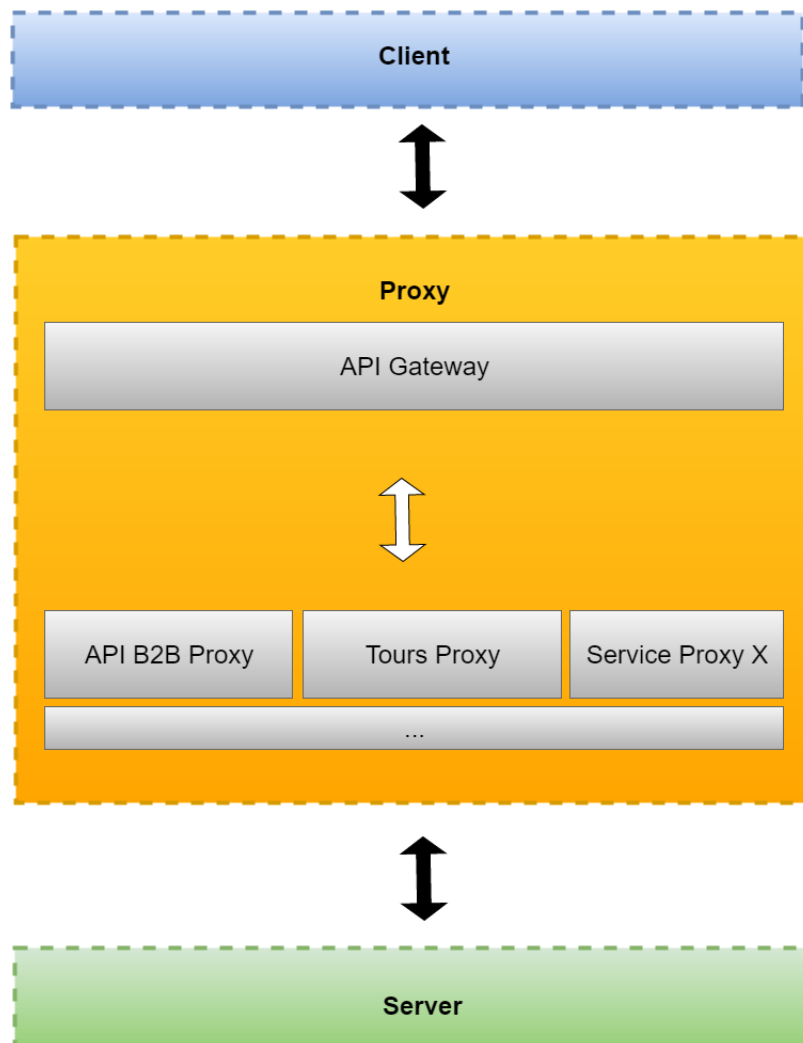


Figura 4 - Arquitetura da camada Proxy

Esta camada de proxy é composta por um conjunto de serviços independentes, sendo que, cada um deles expõe uma interface de comunicação, e é responsável pela adaptação dos pedidos que recebe através dessa interface às funcionalidades disponibilizadas pelo servidor, bem como pela adaptação das respostas recebidas do servidor ao cliente que executou o pedido.

Desta forma, consegue-se uma arquitetura modular e flexível, que permite, além de responder ao principal desafio, escalar facilmente a plataforma para suportar novas categorias de clientes ou expor novas funcionalidades.

A camada do Servidor contém toda a implementação da lógica de negócio da plataforma, que expõe aos seus clientes através da camada Proxy. A sua arquitetura é baseada no paradigma de microserviços, que resulta numa estrutura modular, flexível e escalável. Desta forma, a

complexidade do sistema é minimizada, facilitando assim, a gestão dos processos de desenvolvimento, manutenção e instalação.

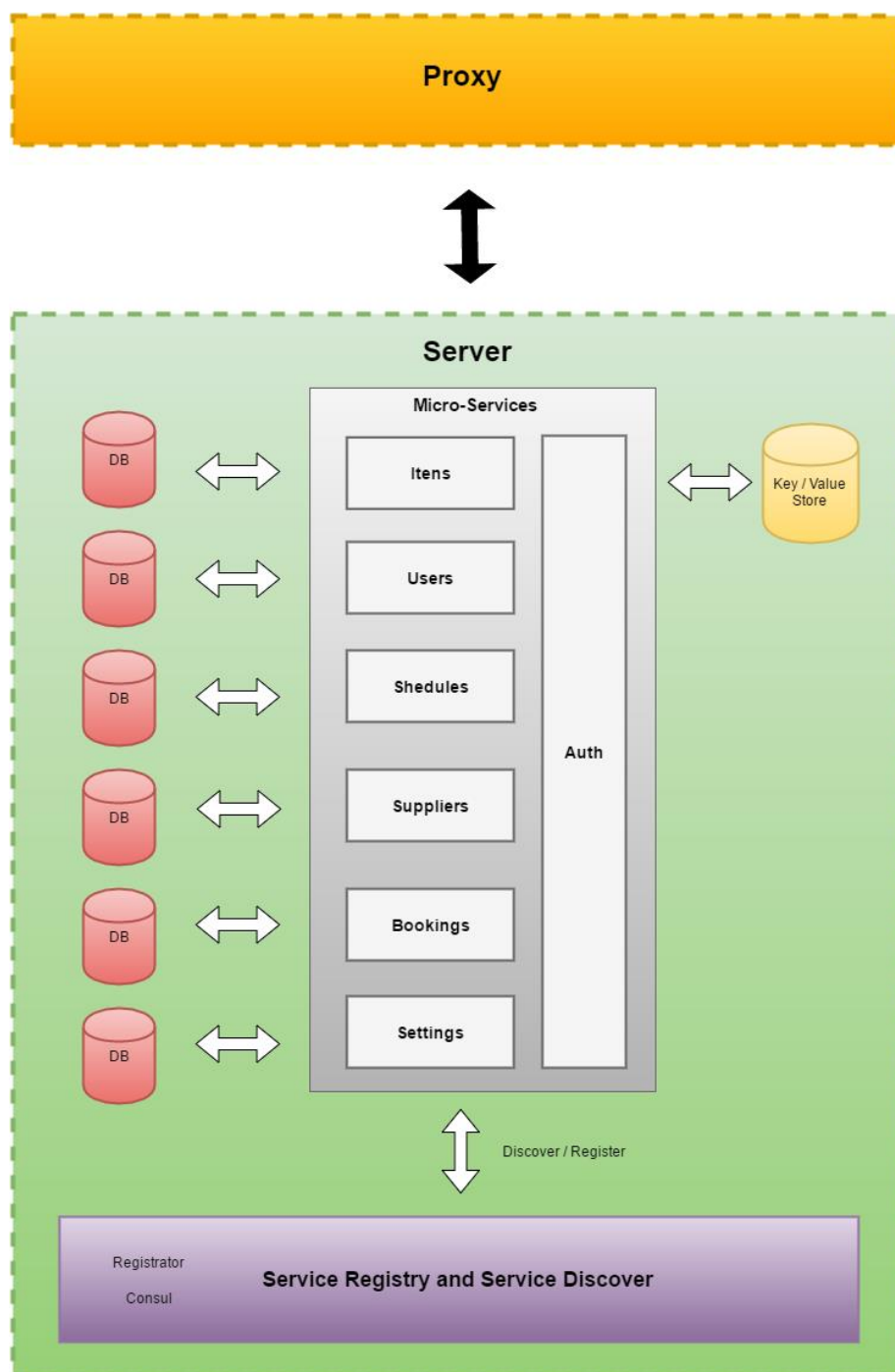


Figura 5 - Arquitetura da camada Servidor

A arquitetura em microserviços surgiu ao longo dos últimos anos para descrever uma forma particular de conceber aplicações de software, como um conjunto de serviços que funcionam de forma independente uns dos outros. Cada serviço executa o seu próprio processo que depende das responsabilidades atribuídas, muitas das vezes, uma API de recursos HTTP.



Podem ser implementados serviços em diferentes linguagens de programação e usam diferentes tecnologias de armazenamento de dados (Sebastián Peyrott, 2015).

A própria independência é a particularidade que torna a programação de serviços cada vez mais atrativos entre as equipas de desenvolvimento. Pode-se ter diferentes equipas a trabalhar de forma isolada. Por outro lado, permite de forma favorável às equipas:

- **Liberdade para escolher as ferramentas certas:** É possível desenvolver um serviço numa tecnologia e outros serviços em tecnologias diferentes, isto permite tirar partido das capacidades de várias equipas e de várias tecnologias para módulos distintos dentro da mesma aplicação.
- **Isolamento de falhas:** Existem limites claros das responsabilidades de cada serviço, desta forma isolando o comportamento de cada micro serviço e chamadas isoladas.
- **Maior facilidade na manutenção do código:** Ao isolar os serviços, e como se cada um deles fosse uma aplicação diferente, logo a quantidade de código é muito menor do que numa aplicação onde estariam todos os serviços como um só. Desta forma, estando o código separado por serviços e módulos diferentes a sua manutenção é bastante facilitada.
- **Flexibilidade:** Com utilização de serviços a qualquer momento do desenvolvimento a arquitetura da aplicação é bastante flexível, ao ponto de ser possível alterar facilmente processos e componentes que supostamente seria muito complicado numa arquitetura monolítica.
- **Escalabilidade:** Uma vez que cada serviços funciona de forma independente do resto da aplicação, escalar uma instancia de micro serviço particular é bastante simples. No caso de um serviço se tornar um estrangulamento devido á sua execução lenta, pode-se executar várias instancias em máquinas diferentes para processar os elementos de dados em paralelo. Em sistemas monolíticos a escalabilidade não é de todo trivial. Por exemplo se num módulo existir uma parte de código interna que seja mais lenta, não há nenhuma forma de fazer essa peça individual de execução de código mais rápida.

Uma das principais ferramentas utilizadas nesta aplicação, e que funciona como base para toda a infraestrutura é o Docker. Na documentação online Docker é definido como uma plataforma aberta para o desenvolvimento, transporte e execução de aplicações. O Docker permite criar máquinas virtuais, denominadas como containers e colocar aplicações dentro desses containers que ficarão completamente isoladas, ou seja, pode-se colocar esse container em qualquer servidor que vai logo funcionar.

De forma simplificada, podemos associar Docker containers a máquinas virtuais muito “leves” (na realidade é muito mais que isso). Por outro lado, as máquinas virtuais demoram bastante tempo a ligar comparado com o Docker container que normalmente liga em segundos. Cada container pode conter o seu próprio Sistema Operativo, ter os seus próprios ficheiros, correr os seus próprios processos e muito mais.

Os benefícios de utilizar containers nas aplicações são vários e cada vez mais a comunidade vai arranjado formas inovativas de os utilizar em diversos modos. Por outro lado, uma das grandes vantagens está relacionado com a portabilidade e facilidade de instalação de ambientes.

Um dos principais princípios de Docker é garantir que independentemente das diferenças entre os ambientes de desenvolvimento, testes e produção, os containers / aplicação vão correr sempre da mesma forma.

De uma forma geral verifica-se vários padrões da arquitetura em microserviços com determinadas regras de forma a garantir que a aplicação funciona como esperado. Padrões estes que são identificados como **Registo e Descoberta de Serviços**, **API Gateway**, **Comunicação entre Microserviços** e **Base de Dados** (Martin Fowler, 2014).

Registo e descoberta de serviços é considerado o modulo responsável por procurar todas as instancias que estão a correr de cada serviço e registá-los num container independente, onde será permitido a pesquisa pelos serviços ativos.

Para a descoberta dos serviços é utilizada como a ferramenta o Consul que é descrito no GitHub como "uma ferramenta para descoberta, monitoramento e configuração de serviços". Ele oferece um sistema distribuído, altamente disponível para registrar serviços, armazenar configuração compartilhada e manter uma visualização precisa de vários datacenters. Disponibiliza uma API HTTP que permite às aplicações registarem-se automaticamente cada vez que são inicializadas, ou então fazer consultas sobre o estado de um determinado serviço (ou uma lista de vários serviços).

A ferramenta intermediária que verifica, analisa e regista os containers/serviços que estão a correr no Consul é identificada como Registrator. Quando um container é inicializado o registrator é notificado sobre este novo container pelo Docker, e automaticamente este regista no Consul com todos os detalhes do serviço. Por outro lado, quando um container é parado ou deixa de funcionar o registrator deteta e remove-o automaticamente.

O módulo da API Gateway é o ponto de entrada para todos os clientes. Será responsável pelo roteamento e balanceamento de carga dos pedidos. Reencaminha todos os pedidos para o serviço indicado. Para fazer este roteamento é utilizado o Nginx como proxy com um ficheiro

de configuração gerado através do consul-template. O consul-template é uma biblioteca que faz consultas ao Consul sobre um ou vários serviços, e escreve a informação desses serviços para um ficheiro baseado num template onde substitui variáveis por informação relacionada com os serviços, como por exemplo o IP e Porta.

A comunicação entre microserviços é considerada o maior desafio nesta arquitetura e por consequência também na aplicação a desenvolver. Em cada serviço são definidos casos de uso, e cada serviço comunica com os outros serviços pedindo para executar determinadas tarefas dependendo das responsabilidades de cada um.

A forma mais simples para efetuar esta comunicação entre os serviços seria através de pedidos HTTP, mas o problema maior com esta abordagem são os atrasos na comunicação. Contudo procurou-se uma solução mais viável e encontrou-se gRPC, um meio de comunicação desenvolvido e bastante utilizado pela Google. É considerado pela Google como uma "forma de comunicação de baixa latência para criar sistemas massivamente distribuídos que abrangem centros de dados, bem como aplicativos móveis, comunicações em tempo real, dispositivos IoT e APIs". Cada serviço é composto por uma ou várias interfaces que representam cada a mensagem. Estas interfaces são programadas utilizando Protocol Buffers IDL, em que o princípio deste protocolo é através de um ficheiro com uma determinada estrutura ser possível gerar o código para o cliente e para servidor mesmo que estes estejam em linguagens diferentes.

Em termos de base dados, cada serviço terá uma base de dados independente do resto da aplicação e com sua própria estrutura. A relação entre diferentes bases de dados é feita apenas por id's externos sem qualquer conhecimento de onde provem cada um destes id's, sendo apenas uma referencia para um determinado registo noutra base de dados. Com esta abordagem qualquer base de dados resiste facilmente com alterações na estrutura sem grandes proporções em toda a base de dados do sistema.

Neste projeto irão ser utilizados dois tipos de base dados em diferentes serviços, base de dados relacionais e não relacionais. Em serviços onde as transações de dados são reduzidas optou-se por usar uma base dados relacional como Postgres SQL. Por outro lado, para serviços em que existe registo de grandes volumes de dados optou-se por uma abordagem não relacional (NoSQL), em que o motor de base de dados é MongoDB.

## **4. Serviços Implementados**

Todo o sistema é composto por vários serviços definidos cada um como um componente independente. Tendo em conta que, cada um destes serviços conhecem as responsabilidades de cada um, eles comunicam entre si.

### **4.1 Serviço de Itens**

O serviço de itens é o serviço responsável por toda a gestão de ofertas presentes na plataforma, assim como das entidades que se relacionam diretamente. Considera-se como entidades deste serviço as tabelas que representam os módulos necessários para a gestão das ofertas, tais como, categorias de ofertas, as ofertas, os eventos presentes para cada oferta, os idiomas que compõe cada oferta e por fim os comentários. A base de dados deste serviço tem como SGBD o PostgreSQL e está estruturada com os seguintes módulos.

#### **4.1.1 Categorias**

As categorias permitem que as ofertas sejam agrupadas por determinadas características definidas por uma categoria. Desta forma, consegue-se de forma simples aplicar filtros sobre um indeterminado número de ofertas.

#### **4.1.2 Ofertas**

O módulo de ofertas é o mais importante módulo deste serviço, maior parte dos restantes módulos dependem deste. Contém toda a informação diretamente relacionada com as ofertas,

tais como localização, títulos, descrições, quantidade máxima de participantes, entre outros dados.

### 4.1.3 Eventos

Este módulo é composto pela gestão dos dados relacionados com a identificação temporal, ou seja, as horas a que uma determinada oferta ocorrerá, e também é composta por a definição de preços. Cada oferta poderá ser composta por vários eventos, por exemplo se uma oferta ocorrer três vezes por dia em diferentes horas, terá que ter três eventos diferentes. Em cada evento é registado a hora de início e a hora de fim. Por outro lado, o registo de datas é responsabilidade de outro serviço, o serviço de Calendário. Os preços para um determinado evento são definidos com base no tipo de participante.

### 4.1.4 Idiomas

Cada oferta permite ter associado uma ou vários idiomas para a tradução de conteúdos em alguns campos específicos, estas línguas são previamente configuradas para todo o sistema.

Durante o desenvolvimento deste módulo abordaram-se várias opções para a gravação nas diferentes línguas, dos conteúdos na base de dados. A primeira opção passava por criar uma tabela isolada, em que seria gravada uma referência para o parâmetro específico (p. e. o nome do atributo), depois era persistido no mesmo registo a língua associada e o conteúdo pretendido. A segunda opção, que foi a que acabou por ser implementada, consiste em gravar os dados num campo com estrutura JSON, em que é composto pela chave da língua e o conteúdo correspondente. Desta forma aproveitou-se um módulo adicionado recentemente ao Postgres, que permite efetuar operações de pesquisa sobre os conteúdos contidos dentro de estruturas JSON.

### 4.1.5 Comentários

O módulo de comentários é diretamente relacionado com as ofertas em que cada oferta pode conter vários comentários.

## 4.2 Serviço Proxy de Ofertas Turísticas

Este serviço funciona como Proxy para o serviço de Itens, em que o seu principal objetivo é acrescentar alguns campos específicos das ofertas turísticas e criar operações específicas sobre estes campos. Optou-se por desenvolver este serviço de forma a permitir deixar o

serviço de itens o mais genérico possível, possibilitando adaptar-se a várias áreas de negócio dentro do turismo, e não ficar tão focalizado apenas nas ofertas turísticas.

Durante o desenvolvimento deste serviço foram aparecendo alguns desafios, em que foi necessário mudar algumas abordagens inicialmente definidas através de conceitos teóricos. A comunicação entre o serviço de proxy e o serviço de itens tinha sido pensada para utilizar gRPC / Protobuf, no entanto ao efetuar-se os testes funcionais deparou-se com um problema na mensagem enviada entre os serviços. Com Protobuf é necessário criar toda a estrutura da mensagem com seus atributos, assim como a definição do tipo de cada atributo. Está definido na documentação do Protobuf, que em situações que é enviado o valor por defeito de um atributo (p.e. Booleano é falso, inteiro é zero), esse atributo não é enviado na mensagem. A integração desta situação no serviço a desenvolver representa uma falha, tendo em conta que existem cenários onde é necessário o envio de todos atributos, como por exemplo na listagem de ofertas quando são enviados os filtros. Mesmo que seja enviado com o valor por defeito é necessário que seja enviado para aplicar o filtro na lista. Perante esta falha optou-se por mudar toda a comunicação para REST.

Este serviço não tem qualquer tipo de persistência de dados, logo não está ligado a nenhum SGBD.

### 4.3 Serviço de Calendários

De forma a que os tempos disponíveis para as reservas de ofertas fiquem centralizados criou-se o serviço de calendários. Este serviço tem como principais funções, criar sessões para uma determinada oferta e gerar blocos de disponibilidade para a sessão correspondente. Os blocos de disponibilidade são gerados através de regras de repetição de eventos (RFC-5545). Este formato de dados é utilizado para representação e partilha de informação de calendários tais como eventos, listas de tarefas, informação de disponibilidade, protocolos ou serviços de calendarização independentes. Consiste em gerar uma ou várias linhas de código sendo baseado em determinadas regras sobre o evento que se pretende criar. Regras essas que podem ser por exemplo os dias que se repete, as horas, o dia final do evento, se tem alguns dias de exclusão, entre outras inúmeras regras possíveis.

A primeira abordagem para a geração dos tempos de disponibilidade de uma oferta passava por gravar na base de dados apenas o código da regra, e cada vez que eram pedidos os tempos de disponibilidade, o sistema gerava em tempo de execução todos os blocos. Para se verificar que esta alternativa seria viável futuramente realizou-se uma prova de conceito, que consistia em calcular o tempo que demoraria a processar um pedido, tendo em conta que seria uma oferta com um mês de disponibilidade. Verificou-se que o serviço demoraria uma média de 2 segundos por pedido, o que está bastante abaixo do que se pretende, até porque quando se

interligar com os outros serviços irá afetar o tempo de resposta deles. Após esta análise cuidada aos tempos de processamento chegou-se à conclusão que, a melhor solução seria gerar e gravar todos os tempos de disponibilidade inicialmente quando é criada a sessão para uma determinada oferta.

Com a abordagem de gravar todos os tempos de disponibilidade das sessões, terá que haver maior suporte em pesquisas para grandes quantidades de dados. Devido ao elevado número de dados que este serviço terá que gerar, teve que se optar por abordar um SGBD baseado em NoSQL. O sistema que se optou por utilizar foi o MongoDB.

### 4.4 Serviço de Utilizadores

O serviço de utilizadores desempenha funções relacionadas com toda a gestão de utilizadores do sistema. É composto por dois tipos de utilizadores, utilizadores de backoffice e clientes do sistema.

Os utilizadores de backoffice foram criados com intuito de administração de toda a informação relacionada com a entidade a que pertence. Permite-lhes realizar funções ligadas a entidade como, a gestão de ofertas, a gestão de todos os clientes, assim como a visualização das reservas efetuadas pelos clientes.

Os clientes do sistema são os utilizadores finais, isto é, são os utilizadores para o qual se destina e se focaliza o desenvolvimento desta plataforma. É possível com este tipo de utilizador realizar um conjunto de tarefas relacionadas com as ofertas turísticas. De uma forma mais alto nível, permite pesquisar ofertas disponíveis numa determinada entidade, consultar informação de uma entidade específica, consultar a disponibilidade de uma oferta para uma data ou um determinado período de tempo e efetuar as respetivas reservas.

Outro módulo incluído neste serviço com um elevado grau de importância no sistema, é a autenticação dos utilizadores. A nível tecnológico a autenticação é realizada com base na autenticação por Token, através de uma tecnologia que surgiu recentemente, JSON Web Tokens (JWT). Esta tecnologia é representada por um standard (RFC 7519) que define uma forma de transmitir informação através de objetos JSON. A informação enviada é assinada por um código secreto para garantir a autenticidade da mesma (auth0.com, 2015). Desta forma é gerado um Token com base em informação específica para o sistema, como o tempo de expiração, assim como o id e o tipo de utilizador autenticado.

Após o processo de autenticação do utilizador é necessário que, o serviço de utilizadores verifique a autenticidade do token em todos os pedidos efetuados para qualquer um dos outros

serviços. Desta forma, a necessidade de processamento e consultas à base de dados, foi a principal razão por se ter optado por esta tecnologia.

Como se pode verificar na imagem em baixo o processo de autenticação é bastante simples. Inicialmente, o utilizador faz o pedido para autenticação e o serviço devolve o token de sessão. Depois, em cada pedido realizado ao sistema é necessário enviar o token devolvido e o serviço de autenticação, neste caso é o serviço de utilizadores, verifica a autenticidade do token e devolve a resposta de sucesso ou insucesso à ação requisitada.

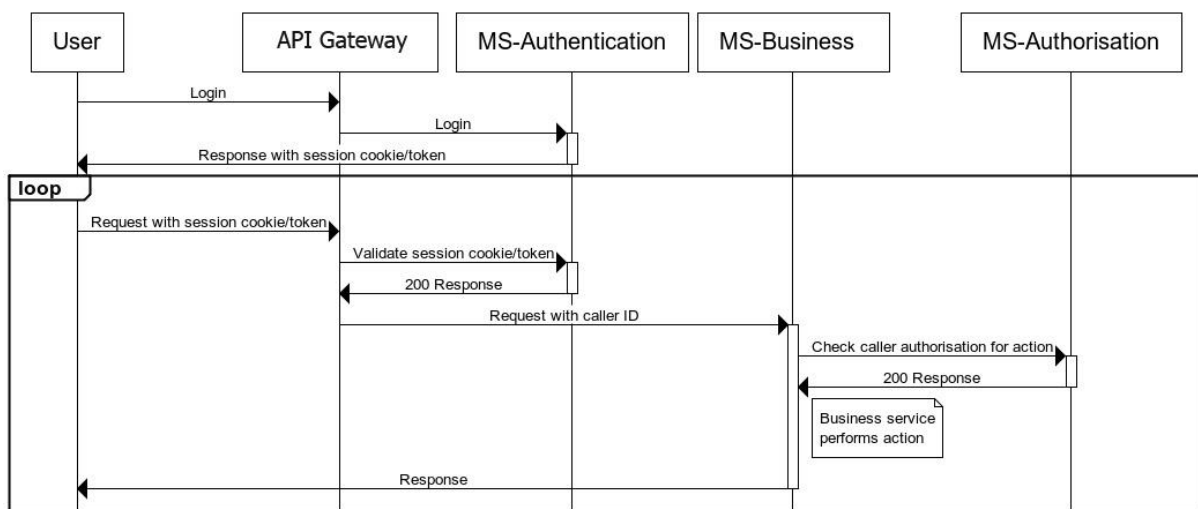


Figura 6 – Descrição do processo de Autenticação

## 4.5 Serviço de Reservas

Este serviço tem como finalidade a integração de toda a gestão de reservas no sistema. Perante as funcionalidades pretendidas para este serviço, é possível verificar que necessita dos outros serviços para que seja possível concluir as suas responsabilidades com sucesso. Isto é, para se realizar uma determinada tarefa existe um conjunto de comunicações com os outros serviços. Por exemplo, no processo de registo de uma reserva é necessário consultar o serviço de itens para verificar se a oferta existe no sistema, também é preciso consultar os tempos de disponibilidade da oferta no serviço de calendários e também é exigido ao serviço de utilizadores a informação do utilizador que pretende efetuar a reserva.

## 4.6 Serviço de Multimédia

Este serviço é considerado o mais simples de todo o sistema, isto porque as suas funções são de certa forma menos complexas do que qualquer outro serviço. Tem como principal função receber ficheiros multimédia, registá-los na base de dados e grava-los no sistema de ficheiros.



Por outro lado, tem como responsabilidade disponibilizar os ficheiros para download a quando é feito o pedido correspondente.

## **5. Testes e Resultados obtidos**

A fase de testes é uma das tarefas mais importantes do desenvolvimento de software, isto porque, tem como objetivo validar e apresentar evidências de confiabilidade contribuindo para uma maior qualidade do software.

Este capítulo tem por objetivo detalhar os testes realizados no sistema durante o desenvolvimento, assim como os resultados obtidos.

### **5.1 Testes Unitários e de Integração**

Durante o desenvolvimento foram criados testes unitários em cada serviço, que consistem principalmente em testar cada componente individual. Estes testes garantem que cada módulo funciona como esperado, executando cada operação disponível. Atualmente, tem tendência a serem compostos por vários níveis, desde que maior parte dos módulos tem vindo a ser feitos por pequenos componentes.

Foram também desenvolvidos testes de integração, de forma a testar as funcionalidades a um nível mais alto, já com os vários serviços interligados entre si. O objetivo destes testes é garantir que determinadas funcionalidades funcionem corretamente quando são integradas com os outros serviços. Por vezes um componente encontra-se a funcionar corretamente quando é testado de forma unitária, no entanto quando é integrado com outros componentes a funcionalidade em si não funciona de forma esperada. Desta forma o principal objetivo dos testes integração passa por cobrir essas falhas.

## 5.2 Resultados Obtidos

De forma a poder-se avaliar todo o estudo efetuado e todo trabalho desenvolvido durante este projeto, pretende-se neste subcapítulo enumerar os objetivos propostos e analisar com base nos resultados obtidos após a conclusão deste projeto. Desta forma, nos parágrafos seguintes são enumerados os objetivos propostos inicialmente.

Relativamente ao objetivo principal proposto no início do projeto **Criação de ferramentas de agregação e disponibilização de ofertas**, o resultado obtido passa por um sistema de backend, desenvolvido no intuito deste projeto e, uma plataforma de frontend desenvolvida em parceria com a empresa Ubiwhere.

Antes de ser inicializado o desenvolvimento da plataforma foi realizada uma análise das várias tecnologias utilizadas na área de turismo e das tendências tecnológicas para os próximos anos nesta área. Com base na realização deste estudo foi possível definir os critérios utilizados para o desenvolvimento do projeto.

O sistema de backend é composto por um conjunto de endpoints para uma ou várias funcionalidades específicas. É possível verificar nas seguintes imagens alguns exemplos de endpoints definidos.

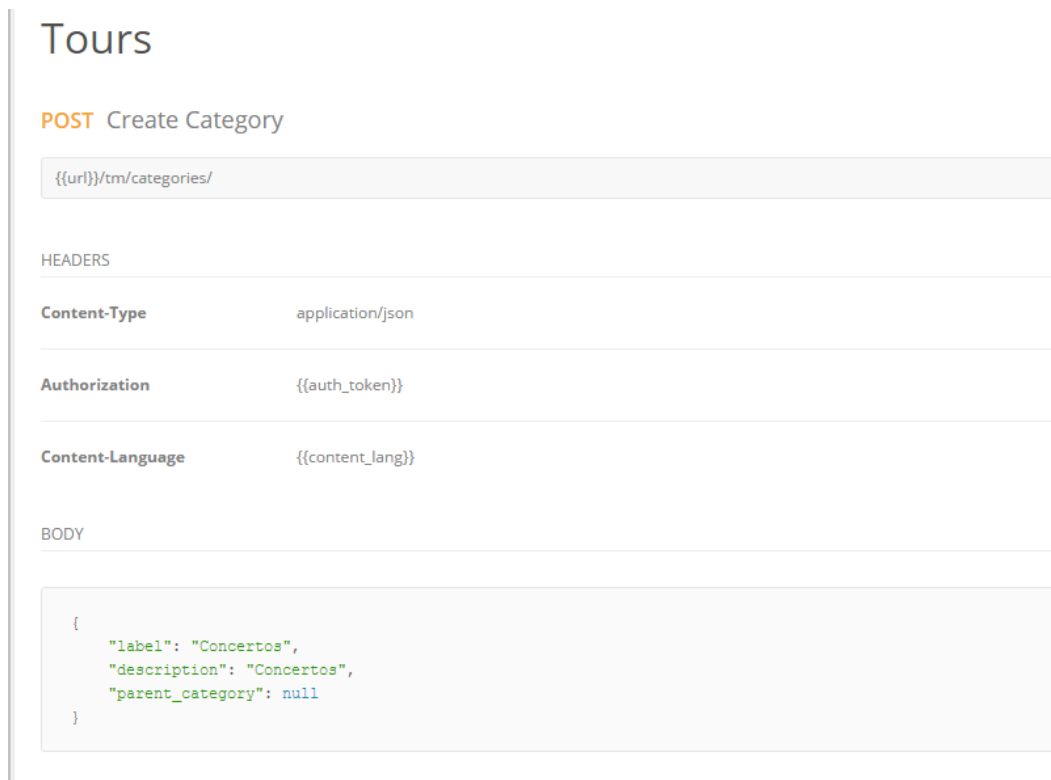


Figura 7 – Endpoint para criação de categorias das ofertas

A imagem anterior está relacionada com a criação das categorias de ofertas, no entanto, pode-se analisar alguns pontos importantes que são comuns a todos os pedidos:

- Verifica-se o seguinte o url “`{{url}}/tm/categories/`” em que `{{url}}` é o endereço do serviço, depois o `/tm/` é identificação do serviço que se está a utilizar e por fim, `/categories/` é módulo utilizado no serviço.
- Nos headers do pedido estão representadas duas variáveis, `{{auth_token}}` e `{{content_lang}}`. A variável `{{auth_token}}` é o token de autenticação do utilizador e a variável `{{content_lang}}` é língua estabelecida para os conteúdos enviados para o sistema.

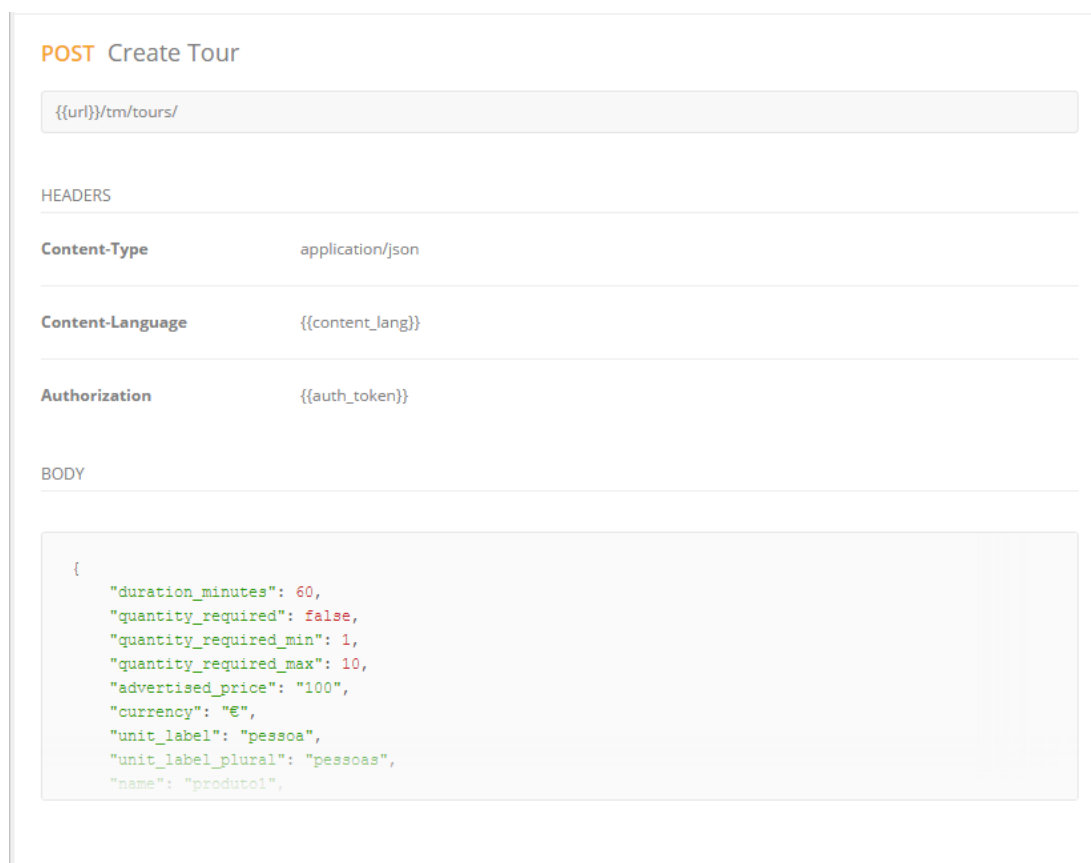


Figura 8 – Endpoint para criação de oferta

A imagem anterior e as que se seguem estão relacionadas com as operações disponíveis no módulo de ofertas no serviço das ofertas turísticas.

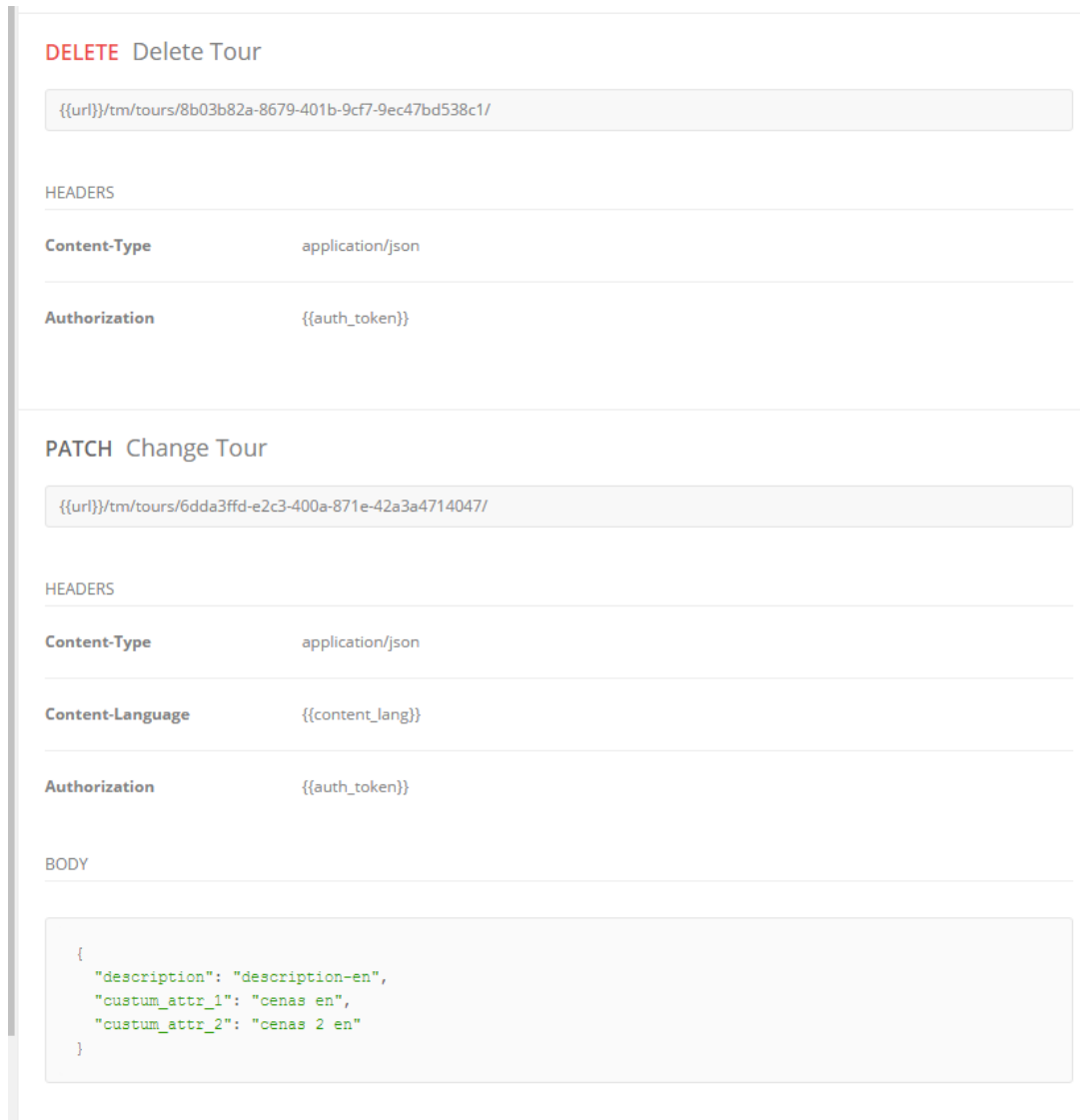


Figura 9 – Endpoint para remover e alterar uma oferta

Relativamente á plataforma de frontend é possível visualizar nas imagens a baixo algumas das funcionalidades implementadas.



Figura 10 – Página Inicial do site

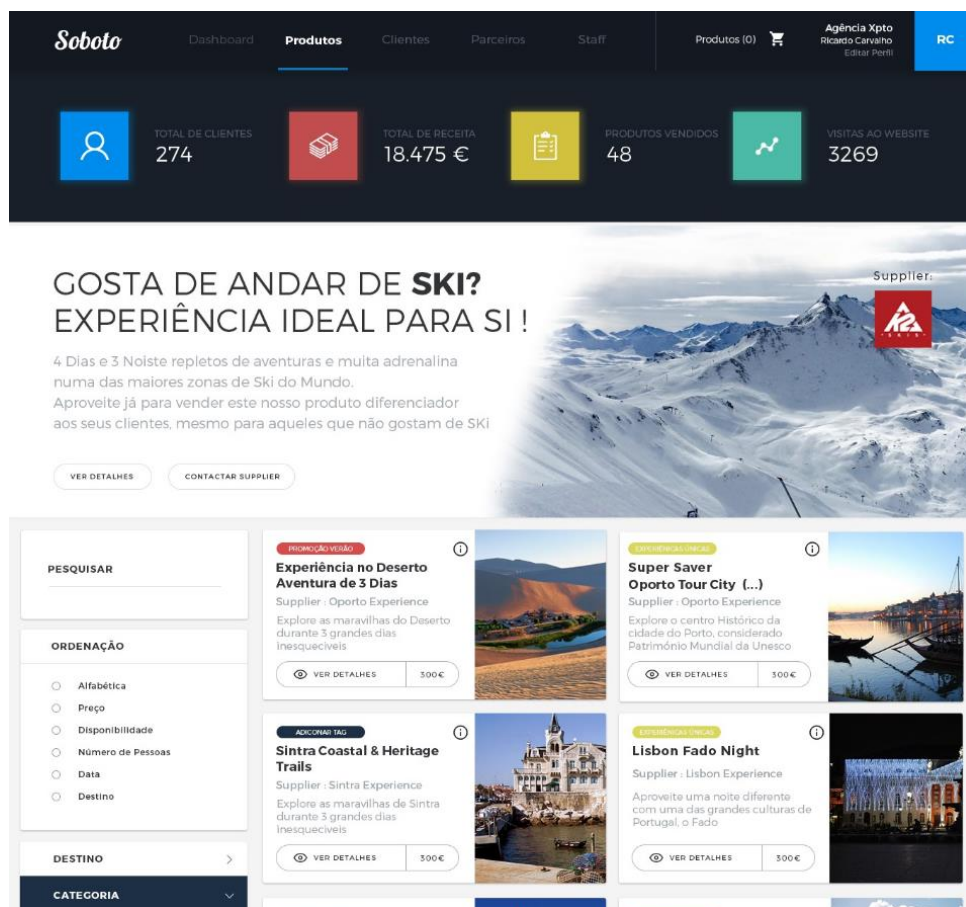


Figura 11 – Área de gestão de ofertas

The screenshot displays the 'Produtos' (Products) section of the Soboto dashboard. The top navigation bar includes 'Dashboard', 'Produtos' (active), 'Clientes', 'Parceiros', and 'Staff'. A user profile icon and a shopping cart icon are also present. Below the navigation bar, a summary row shows four key metrics: 'TOTAL DE CLIENTES' (274), 'TOTAL DE RECEITA' (18.475€), 'PRODUTOS VENDIDOS' (48), and 'VISITAS AO WEBSITE' (3269). The main content area is titled 'DETALHES DO PRODUTO' and features a sidebar with a list of product attributes: 'Nome do produto', 'Código do produto', 'Categoria do produto', 'Descrição pequena do produto', 'Descrição longa do produto', 'Destaques do produto', 'O que inclui o produto', 'O que não está incluído no produto', 'Informações adicionais', and 'Política de cancelamento'. The main form area contains four input fields: 'NOME DO PRODUTO\*' (with placeholder 'Escreva o nome do produto'), 'CÓDIGO DO PRODUTO\*' (with placeholder 'Associe um código ao produto'), 'CATEGORIA DO PRODUTO\*' (with placeholder 'Associe uma ou mais categorias ao produto'), and 'DESCRIÇÃO PEQUENA DO PRODUTO\*' (with placeholder 'Resumo do produto'). The language is set to 'Portuguese'.

Figura 12 – Página de criação de novos produtos

Relativamente ao objetivo **Criação de ferramentas de reserva e recomendação por canais sociais**, não foi possível abordar este tema, devido á falta de meios para se poder avançar. Isto é, como a falta de tempo não permitiu terminar os objetivos anteriores, e este objetivo dependia da conclusão dos anteriores, não houve possibilidades de passar para quais queres funcionalidades de meios de reserva e recomendações por canais sociais. No entanto, irá ser referenciado no capítulo seguinte como trabalhos futuros a desenvolver para este sistema.

## 6. Conclusão e Trabalho Futuro

Neste capítulo vão ser referenciadas as principais conclusões retiradas do trabalho realizado e vai ser também enumerados alguns pontos a abordar para o futuro desenvolvimento do sistema realizado neste projeto.

No planeamento do projeto definiram-se duas fases para a elaboração deste projeto, **Estudo da Solução e Planeamento** e **Desenvolvimento e conclusão do relatório**. Em cada um destas fases foram envolvidas várias tarefas, envolvendo Estudos Técnicos, Especificações detalhadas, Aquisição de novos conhecimentos, Desenvolvimento tecnológico, Construção de Protótipos e Testes. Para cada uma das tarefas concluídas foi-se realizando um relatório com estudo das competências e conhecimentos adquiridos com objetivo de facilitar a realização deste relatório final.

Durante este projeto foram aparecendo algumas dificuldades, nomeadamente no capítulo do estudo das plataformas existentes e na análise de desenvolvimento. No estudo das plataformas existente deve-se à escassez de informação relacionada com os aspetos de cada sistema analisado. Em relação à análise de desenvolvimento, optou-se pela pesquisa e revisão de literatura nesta área o que acabou por provocar alguns atrasos nalgumas tarefas, mas que se tornou muito gratificante pela aquisição de novos conhecimentos e conceitos numa área que tem crescido muito nos últimos anos.

Na fase de desenvolvimento foi-se registando alguns desafios interessantes, inicialmente realizou-se uma análise da arquitetura de microserviços, e estudando-se as tecnologias utilizadas nesta arquitetura e ao mesmo tempo foi surgindo um enorme fascínio e motivação por toda esta arquitetura.



## **6.1 Trabalho Futuro**

O desenvolvimento deste sistema não termina com a entrega oficial desta Dissertação, visto que o sistema em si ainda se encontra em fase de desenvolvimento. Desta forma, os serviços desenvolvidos foram já preparados para lidar com alterações e, teve-se sempre em consideração a possibilidade de adição de novos componentes e módulos. Posteriormente também deverá ser necessário a inclusão de mais serviços no sistema, estando a arquitetura preparada para isso.

Encontram-se já estabelecidos alguns pontos importantes para o futuro deste sistema, entre eles destacam-se:

### **6.1.1 Desenvolvimento de marketplace para as entidades**

O desenvolvimento de um Marketplace irá permitir a venda das ofertas registados no sistema por revendedores, isto é, as entidades de ofertas turísticas terão a possibilidade de vender as suas ofertas noutros sites de revendedores. O sistema será preparado para que as plataformas dos revendedores possam comunicar com o sistema disponibilize métodos para facilitar a venda das ofertas.

### **6.1.2 Criação de serviço proxy para ofertas de hotéis**

Encontra-se já delineado a criação de um novo serviço para adaptação de ofertas de hotéis. Será um serviço semelhante ao já desenvolvido de ofertas turísticas.

### **6.1.3 Criação de ferramentas de reserva e recomendação por canais sociais**

Conforme já referido anteriormente esta funcionalidade era prevista para ser integrada no sistema e também neste relatório. Desta forma, ficará como das funcionalidades mais importantes a desenvolver no futuro deste sistema.

## REFERÊNCIAS

- auth0.com. (2015). Get Started with JSON Web Tokens - Auth0. Retrieved July 8, 2016, from <https://auth0.com/learn/json-web-tokens/>
- B'TEN, B. T. E. N.-. (2014). Estudo de atratividade e potencialidade nos clusters – Agroalimentar, Economia do Mar, Cidades Inteligentes, Tecnologias de Produção e Turismo, 118.
- Chris Motola. (2015). Rezdy Review 2016. Retrieved April 14, 2016, from <https://www.merchantmaverick.com/reviews/rezdy-review/>
- Chris Motola. (2016). Checkfront Review 2016. Retrieved April 16, 2016, from <https://www.merchantmaverick.com/reviews/checkfront-review/>
- Gomes, P. F. L. (2011). Migração de aplicações legadas para bases de dados NoSQL. *Universidade Do Minho, 1*, 144. <http://doi.org/10.1017/CBO9781107415324.004>
- Jennie Lees. (2015). The deceptively complex world of RRULEs. Retrieved August 1, 2016, from <https://nylas.com/blog/rrules/>
- Kevin May. (2009). TourCMS builds Marketplace for niche tour operators. Retrieved April 16, 2016, from <https://www.tnooz.com/article/tourcms-builds-marketplace-for-niche-tour-operators/>
- Martin Fowler, J. L. (2014). Microservices. Retrieved July 16, 2016, from <http://martinfowler.com/articles/microservices.html>
- Menezes, P. (2014). Caracterização das empresas do setor do turismo em Portugal.
- Sebastián Peyrott. (2015). An Introduction to Microservices, Part 1. Retrieved July 15, 2016, from <https://auth0.com/blog/an-introduction-to-microservices-part-1/>
- Sezgin, E., & Yolal, M. (2007). Golden Age of Mass Tourism : Its History and Development.
- Shahir Daya Nguyen Van Duy, Kameswara Eati, Carlos M Ferreira, Dejan Glozic, Vasfi Gucer, Manav Gupta, Sunil Joshi, Valerie Lampkin, Marcelo martins, Shishir Narain, R. V. (2015). Microservices from Theory to Practice Creating Applications in IBM Bluemix Using the Microservices Approach. *IBM*, 170.
- Stephen Joyce. (2011). Welcome to a new world of standards in travel technology. Retrieved April 18, 2016, from <https://www.tnooz.com/article/welcome-to-a-new-world-of-standards-in-travel-technology/>
- Steven J. Vaughan-Nichols. (2014). What is Docker and why is it so darn popular? | ZDNet. Retrieved July 15, 2016, from <http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popula>

Street, J. C. (2000). OpenTravel Alliance Message Specifications.

Werthner, H., & Ricci, F. (2004). E-Commerce and Tourism. *Communications of the Acm*, 47(12), 101–105. <http://doi.org/10.1145/1035134.1035141>



ANEXO 1 – MODELO DE DADOS DO SERVIÇO DE ITENS

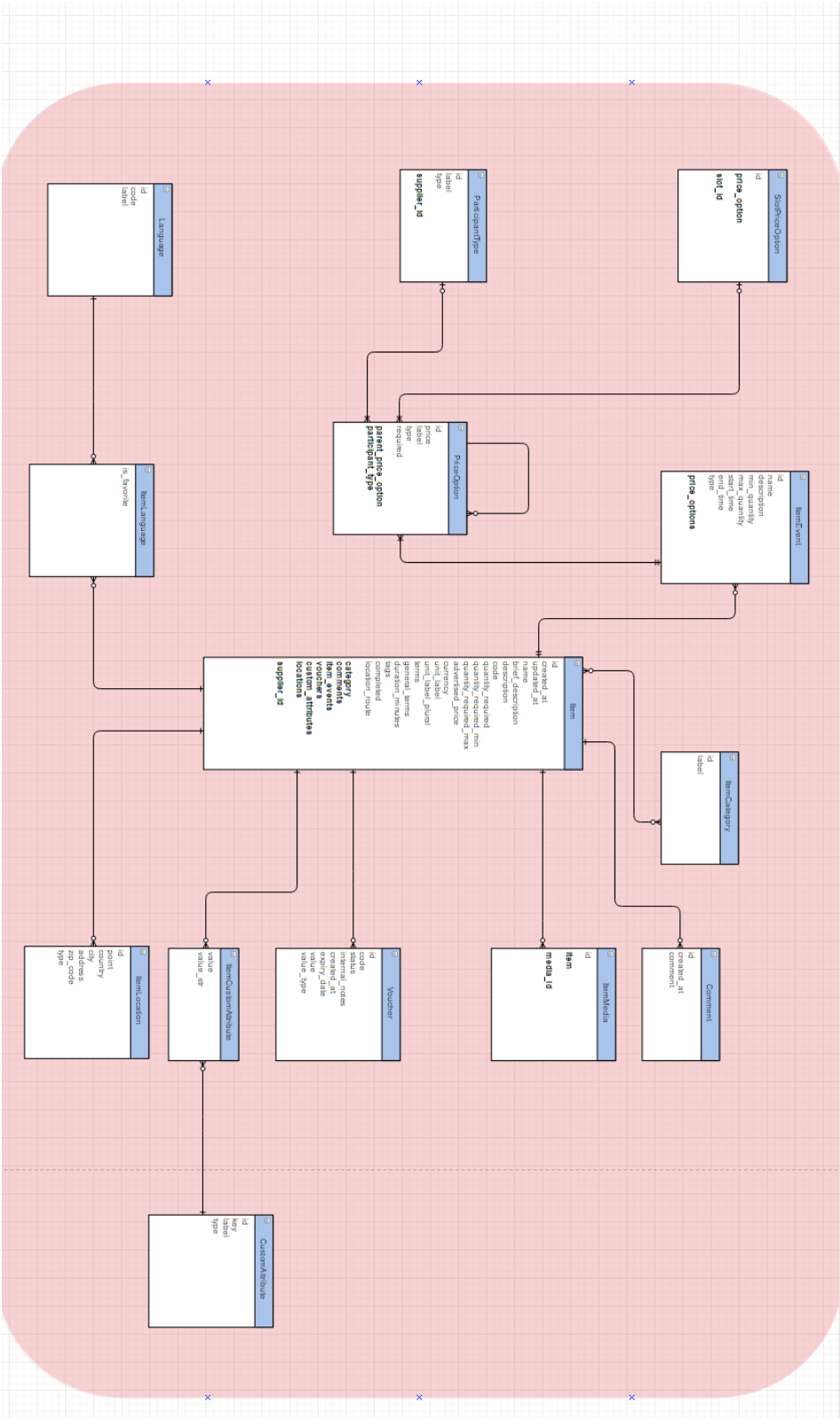


Figura 13 – Modelo de Dados do Serviço de Itens

## ANEXO 2 – MODELO DE DADOS DO SERVIÇO DE UTILIZADORES

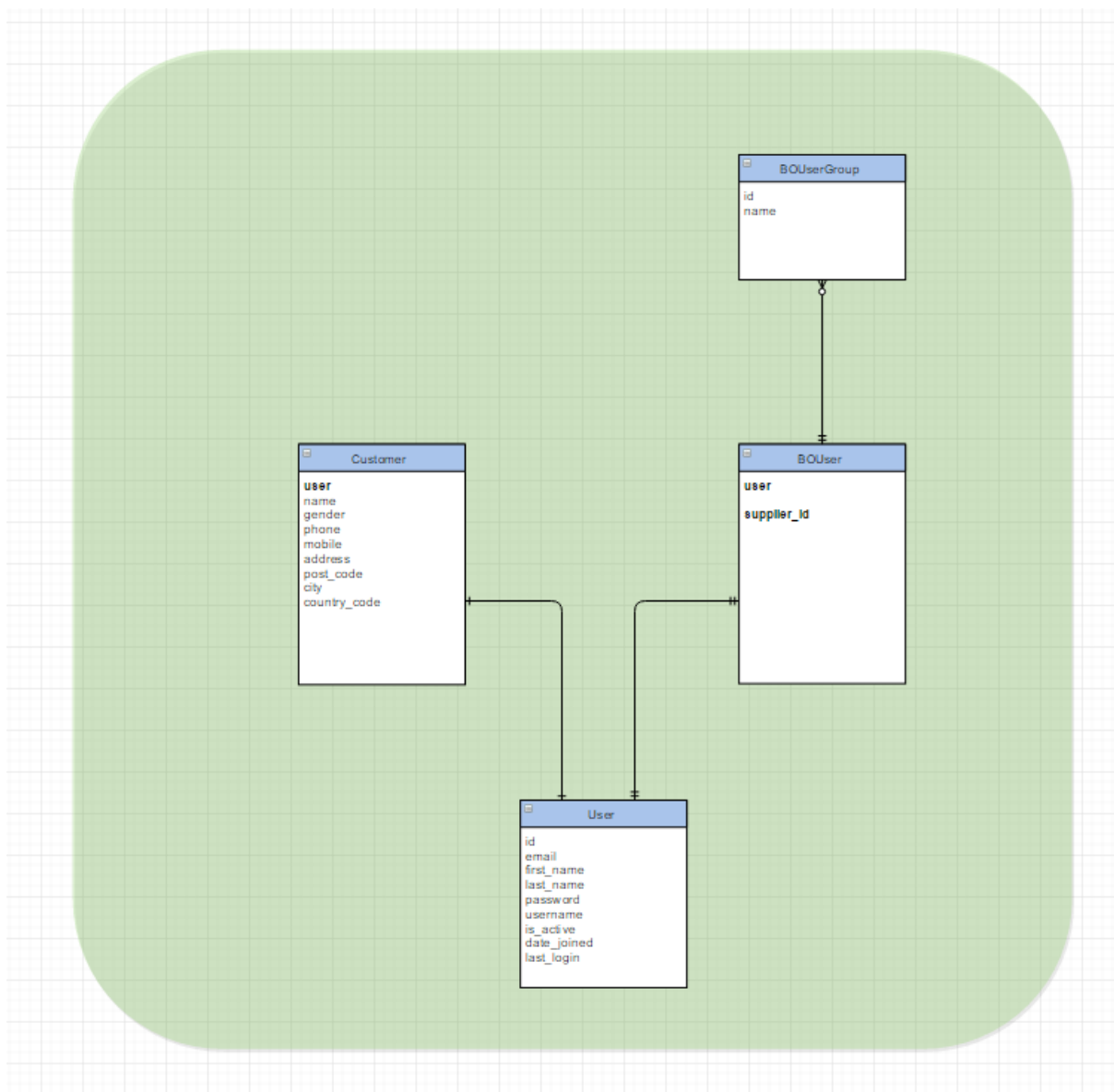


Figura 14 – Modelo de Dados do Serviço de Utilizadores

## ANEXO 3 – MODELO DE DADOS DO SERVIÇO DE RESERVAS

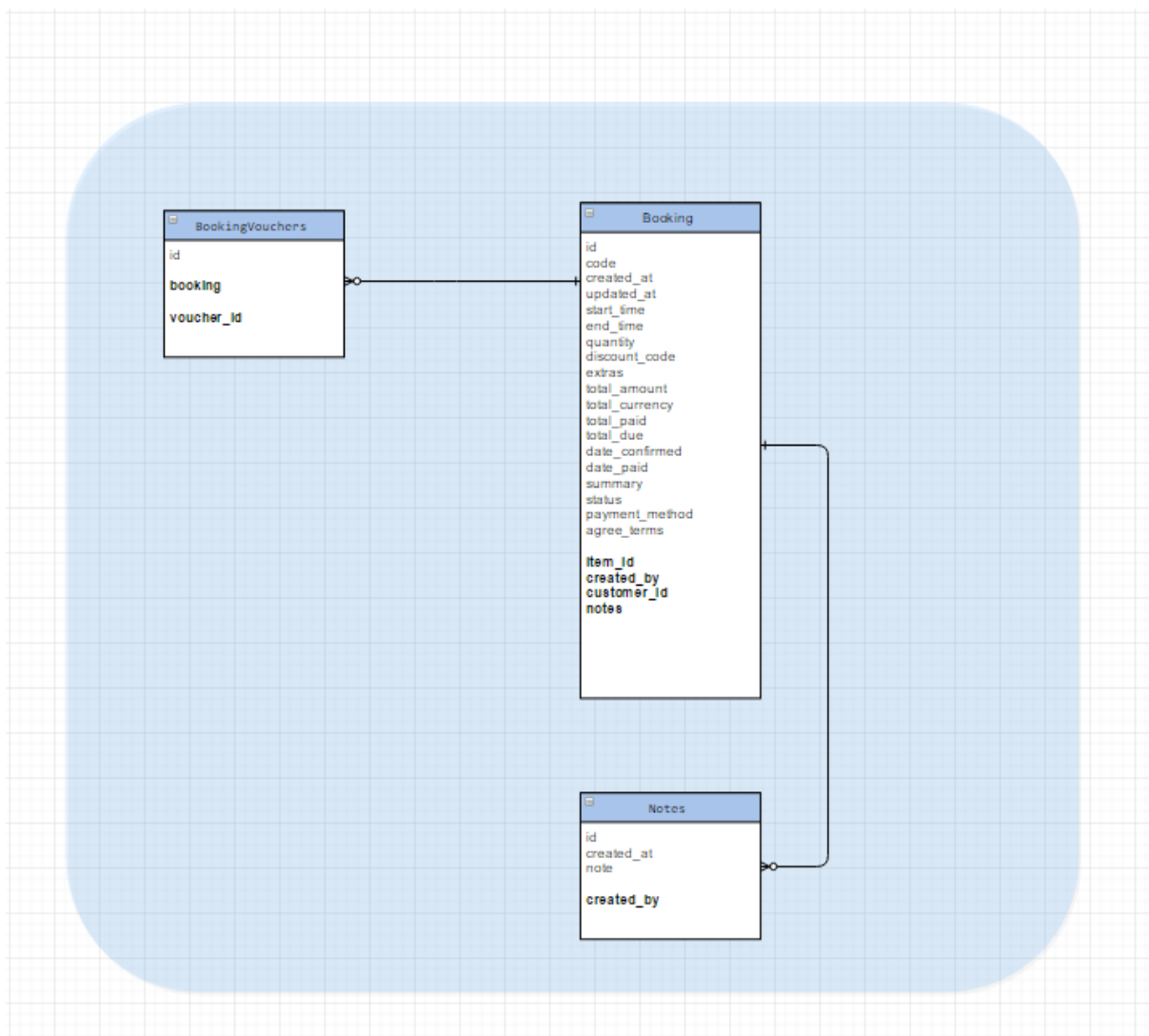


Figura 15 – Modelo de Dados do Serviço de Reservas