



UNIVERSIDADE
LUSÓFONA

BestRide

Trabalho Final de curso

Relatório Final

Nome do Aluno: Diogo Cerqueira a22002160

Nome do Aluno: Erick Pina a22006182

Nome do Orientador: Rui Ribeiro

Trabalho Final de Curso | LEI | 10/06/2023

www.ulusofona.pt

Direitos de cópia

(*BestRide*), Copyright de (Diogo Cerqueira, Erick Pina), ULHT.

A Escola de Comunicação, Arquitetura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

O BestRide tem como objetivo revolucionar o mercado de animação turística, especialmente no setor dos Tuk-Tuks. Atualmente, enfrentamos diversos desafios, como a falta de uma plataforma digital abrangente que ofereça detalhes completos sobre as viagens disponíveis. Isso resulta em dificuldades na busca por motoristas, modificações de preços e incerteza em relação à quantidade de passeios diários realizados pelos motoristas.

O mercado de animação turística é um tanto antiquado, e a BestRide busca levar esse setor para um novo nível tecnológico. Nosso objetivo é otimizar o tempo e o dinheiro tanto para os profissionais quanto para os clientes, além de envolver a comunidade acadêmica no desenvolvimento e na continuidade dessa aplicação inovadora.

O projeto BestRide já estava em desenvolvimento, e tivemos acesso a propostas anteriores de Trabalhos Finais de Curso (TFC). Nos trabalhos anteriores, foram utilizados os serviços da Cloud Amazon AWS. No entanto, devido à saída da cliente, os serviços foram cancelados e o nosso objetivo principal mudou. Sendo este, o processo de migração dos serviços previamente utilizados para os serviços oferecidos pela Microsoft Azure Cloud. Essa transição permitiu que aproveitássemos ao máximo as vantagens e as capacidades da plataforma Azure para melhor atender às necessidades do BestRide.

Abstract

BestRide aims to revolutionize the tourist animation market, particularly in the Tuk-Tuk sector. Currently, we face several challenges, such as the lack of a comprehensive digital platform that provides detailed information about available trips. This results in difficulties in finding drivers, price fluctuations, and uncertainty about the number of daily tours conducted by the drivers.

The tourist animation market is somewhat outdated, and BestRide aims to take this sector to a new technological level. Our goal is to optimize time and money for both professionals and clients while involving the academic community in the development and continuity of this innovative application.

The BestRide project was already in development, and we had access to previous Final Course Work (FCW) proposals. Our task is to continue this project because we believe it has tremendous potential. In previous work, Amazon AWS Cloud services were used. However, due to the client's departure, the services have been canceled and our main objective has changed. This involves migrating from the previously used services to those offered by Microsoft Azure Cloud. This transition allows us to make the most of the advantages and capabilities of the Azure platform to better meet the needs of BestRide.

Índice

Resumo	iii
Abstract.....	iv
Índice	v
Lista de Figuras	vii
Lista de Tabelas	viii
1 Identificação do Problema.....	11
2 Viabilidade e Pertinência	13
3 Benchmarking	14
3.1 Walkbox Self-Guided Tours	14
3.2 TripAdvisor.....	15
4 Engenharia.....	17
4.1 Levantamento e análise dos Requisitos	17
4.1.1 Requisitos Funcionais:.....	18
4.1.2 Requisitos Não-Funcionais:.....	23
4.2 Diagramas de Casos de Uso.....	30
4.3 Diagramas de Atividades	32
4.4 Modelos Relevantes	35
4.4.1 Diagrama Entidade - Relação	35
4.4.2 Diagrama de Classes.....	36
4.4.3 Diagramas de Sequência.....	37
4.5 Estrutura (<i>Mockups</i> , <i>StoryBoards</i> e Mapa Aplicacional)	38
4.5.1 Mockups	39
4.5.2 StoryBoard.....	47
4.5.3 Mapa Aplicacional.....	48
5 Solução Proposta.....	49
5.1 Introdução	49
5.2 Tecnologias e ferramentas utilizadas	50
5.2.1 Microsoft Azure.....	50
5.2.2 Docker	53
5.2.3 Ionic	53

5.2.4	Stripe.....	54
5.2.5	ArcGIS WebService	55
5.2.6	Django	55
5.2.7	MySQL.....	55
5.3	Arquitetura e Implementação.....	56
5.4	Abrangência.....	63
6	Plano de testes e validação.....	65
7	Método e Planeamento	66
7.1	Método de trabalho	66
7.2	Cumprimento dos calendários propostos	66
7.3	Avaliação geral do planeamento.....	67
8	Resultados	13
9	Manual Técnico da aplicação.....	22
	Download das Ferramentas Necessárias para Correr front-end Localmente.....	22
	Correr o Projeto	22
	Download das Ferramentas Necessárias para Correr back-end Localmente.....	22
10	Agradecimentos.....	24
	Bibliografia.....	25
	Glossário.....	26
	Anexos.....	27
	Testes	27
	Guião de Tarefas	27
	Questionário de Satisfação	29
	Compromisso de Confidencialidade	30

Lista de Figuras

Figura 1 – Walkbox	14
Figura 2 - TripAdvisor	15
Figura 3 - Diagrama de Caso de Uso App Turista	31
Figura 4 - Diagrama de Caso de Uso App Driver	32
Figura 5 - Diagrama de Atividade Reservar Tour	33
Figura 6 - Diagrama de Atividade Criar Tour	34
Figura 7 - Diagrama Entidade – Relação	35
Figura 8 - Diagrama de Classes	36
Figura 9 - Diagrama de Sequência Reservar Tour	37
Figura 10 - Diagrama de Sequência Criar Tour	38
Figura 11 - Mockup Página Login	39
Figura 12 - Mockup Página de Registo	40
Figura 13 - Mockup Página Inicial	41
Figura 14 - Mockup Detalhes do Tour	42
Figura 15 - Mockup Página de Reservas	43
Figura 16 - Mockup Página Conta do Usuário	44
Figura 17 - Mockup Página de Definições	45
Figura 18 - Mockup Página Reservar Tour	46
Figura 19 – StoryBoard	47
Figura 20 - Mapa Aplicacional BestRide	48
Figura 21 - Função de upload de imagens Azure Blob-Storage	51
Figura 22 - Função de upload de imagens para a AWS do TFC anterior	51
Figura 23 - Função de translate da aplicação	52
Figura 24 - Função de translate com AWS TFC anterior	53
Figura 25 - Docker	53
Figura 26 - Arquitetura de uma aplicação Ionic	54
Figura 27 - Stripe	55
Figura 28 - Logo django / python	55
Figura 29 – MySQL	56
Figura 30 - Arquitetura da aplicação	56
Figura 31 - Pipelines no azure	60
Figura 32 - Container registry Azure	60
Figura 33 - Container instance Azure	60
Figura 34 - Inserção docker image back-end na container instance	61
Figura 35 - Serviços criado no Azure	62
Figura 36 - Serviços criado no azure - 2	62
Figura 37 - Nova arquitetura no Azure	63
Figura 38 - Calendário de Planeamento	12
Figura 39 - Função de registo de utilizador	14
Figura 40 - Nova implementação do mapa do tour	15
Figura 41 - Comparação login anterior com novo login	16
Figura 42 - Comparação ecrã de registo anterior com o novo ecrã de registo	17
Figura 43 - Comparação ecrã de recuperação de conta	17
Figura 44 - Comparação ecrã de tours	18
Figura 45 - Ecrã de detalhes	19
Figura 46 - Função que retorna tours de acordo com a distância	20
Figura 47 - Calendário de Planeamento Antigo	31
Figura 48 - Calendário de Planeamento Antigo	31

Lista de Tabelas

Tabela 1 - Comparação das Aplicações.....	16
Tabela 2 - Tabela de Requisitos da Migração	25
Tabela 3 - Tabela Requisitos	28
Tabela 4 - Requisitos não implementados / parcialmente implementados	30
Tabela 5 - Amazon Aws vs Microsoft Azure	57

1 Identificação do Problema

A ideia inicial da ferramenta BestRide [3] é resolver problemas no mercado dos Tuk-Tuks. Os Tuk-Tuks têm um papel muito importante na sociedade por conta do seu trabalho com o turismo nos países. O turismo tem uma grande influência nas questões socioculturais e económicas de Portugal, trazendo muitos turistas com vontade de conhecer as grandes áreas de interesse histórico, espalhadas por todas as cidades portuguesas, e assim movimentando o capital do país.

Os Tuk-Tuks têm o papel de oferecer serviços aos turistas auxiliando-os a conhecer os pontos históricos de cada cidade a partir de um roteiro turístico, além de explicar a história de cada local para os seus clientes. Porém, como referido na secção do resumo, esta área dos tuk-tuks ainda tem muitos problemas e precisa de melhorias para ter ainda mais destaque no mercado. Alguns dos problemas que os profissionais enfrentam são:

- Há uma falta de um sistema de informação para a gestão do negócio dos Tuk-Tuks;
- Os preços médios não são facilmente conhecidos, podendo ser muito voláteis entre uma empresa e outra, fazendo que haja alguma investigação por parte dos clientes de forma a perceber quais as empresas cujo preço é mais fiável.
- O facto de haver poucos lugares para as viaturas torna-se muito incómodo, tanto para motoristas quanto para clientes.
- As empresas de Tuk-tuks têm um controlo limitado no que diz respeito á questão de pagamentos dos roteiros e do número real de viagens feitas por dia.
- Não existe uma organização dos roteiros de forma concreta tanto por parte dos motoristas quanto por parte das empresas, o que indica então uma baixa procura pela área.

A ferramenta BestRide vem para resolver estes problemas e melhorar a área do mercado dos Tuk-Tuk's. A aplicação está dividida em duas partes, sendo uma para os turistas (clientes) e outra para os profissionais (motoristas). Com isso, conseguiremos trazer tecnologia e informação para um mercado que está carente disso, tencionando melhorar e facilitar o trabalho dos motoristas e fazendo com que os mesmos ganhem mais visibilidade, assim tendo mais clientes à procura de serviços. Na parte dos clientes, queremos facilitar a busca por turismo, fazendo com que os mesmos tenham todas as informações dos roteiros que desejam fazer e consigam facilmente encontrar um profissional que possa realizar o seu “tour”.

Além dos desafios enfrentados no mercado de Tuk-Tuks, encontramos um obstáculo durante a metade do desenvolvimento do nosso Trabalho Final de Curso (TFC). Os serviços que utilizávamos para dar suporte à nossa aplicação, fornecidos pela Amazon AWS, foram cancelados com a saída da cliente. Como resultado, tivemos que interromper o uso desses serviços e iniciar a migração da aplicação para a plataforma Microsoft Azure

Cloud. Para isso, tivemos que realizar uma análise tecnológica para entender melhor qual o rumo que iríamos tomar. Com isso, seria necessário a realização de uma análise da nova arquitetura do projeto, entendimento dos novos serviços a serem utilizados, análise de prioridades de desenvolvimento como desenvolvimento do back-end em primeiro lugar juntamente com a base de dados e posteriormente o front-end, e por último a realização do planejamento. Todo essa estratégia inicial foi crucial para a conclusão do desenvolvimento e realização do objetivo da mudança tecnológica (migração), e assim, a conclusão do projeto.

No que diz respeito ao momento atual do projeto, conseguimos realizar toda a migração entre as duas plataformas de Cloud (AWS para Azure). Dentro do Azure, conseguimos criar a nossa Base de dados e as VMs que dão suporte ao nosso back-end e as duas aplicações (Turista e Driver). Além disso, conseguimos utilizar os serviços da plataforma para realizar as funções essenciais para o funcionamento das nossas aplicações, entre eles temos Azure Translator e Azure Blob Storage. Gostaríamos de ter utilizado o serviço Active Directory B2C porém não conseguimos implementá-lo da forma correta, para a realização da autenticação de utilizadores. Com isso, implementamos a autenticação da nossa aplicação através da base de dados validando todas as informações necessárias.

2 Viabilidade e Pertinência

O Tuk-Tuk é um negócio em crescimento em Portugal. Cada vez mais, em várias cidades do país, pessoas procuram por trabalhadores que possam realizar animações turísticas e ajudá-los a conhecer as principais áreas de interesse histórico. Com isso, a empresa BestRide visa melhorar um negócio com um potencial muito grande, solucionando problemas que as empresas de Tuk-Tuk enfrentam e facilitando o trabalho dos seus profissionais.

Neste mercado existem poucas soluções e iniciativas de melhorias na área dos Tuk-Tuks. Com isso, podemos afirmar que este não é apenas um projeto académico, tendo como objetivo a continuação após a conclusão do TFC. O projeto surge com a finalidade de solucionar os problemas mencionados na seção anterior, dar prosseguimento a aplicação e fazer com que a mesma funcione com os novos serviços utilizados após a migração, assim, gerando benefícios económicos tanto para as empresas de animação turística quanto para os seus trabalhadores e clientes, mostrando então que este projeto se torna uma necessidade para os mesmos.

A antiga cliente da BestRide, Sónia Gomes, trabalhou durante muitos anos na área dos Tuk-Tuks em Portugal e por isso, tem um conhecimento elevado e ideias muito pertinentes para a evolução e melhoria do negócio. Ideias como um curso para animadores turísticos e também passeios turísticos realizados virtualmente, mostram que este projeto tem uma grande capacidade de crescimento e evolução nesta área.

Segundo a cliente, o projeto BestRide já é um produto de interesse comercial pela câmara de Sintra e pela Polícia local. Esta colaboração dos órgãos públicos seria muito vantajosa para o projeto, principalmente porque seria possível criar áreas de estacionamento próprio de viaturas de Tuk-Tuk, facilitando assim o trabalho dos profissionais e o encontro com os clientes.

Em conclusão, este projeto possui um grande potencial no mercado-alvo e demonstra uma notável capacidade de crescimento. Como mencionado anteriormente, o nosso trabalho é uma continuação de um projeto anterior. Portanto, o nosso objetivo inicial era aprimorar ainda mais o trabalho realizado pelos ex-alunos, corrigindo erros e implementando novas funcionalidades na aplicação. Contudo, o âmbito do nosso projeto foi alterado e passou a ser migrar corretamente todos os serviços e funcionalidades da aplicação da AWS para o Azure. Obtivemos resultados positivos nesse sentido, conseguindo migrar com sucesso a nossa aplicação para a nuvem do Azure e reformular o backend utilizando estes serviços. Com isso, conseguimos implementar todas as funções necessárias e criar novas para melhorar ainda mais a lógica do projeto. Em relação ao frontend, conseguimos fazer algumas alterações de design e resolver alguns bugs e erros anteriores. No entanto, como o nosso objetivo principal se tornou a migração e o desenvolvimento completo do backend da aplicação, infelizmente, não conseguimos efetuar muitas alterações no frontend.

3 Benchmarking

Depois de uma análise de mercado, verificamos que a aplicação BestRide vem para inovar e ser única, pois não existe nenhuma aplicação para os Tuk-Tuk's no mercado, o que é uma vantagem para a empresa. Existem algumas aplicações como “TripAdvisor” e “Walkbox” por exemplo, porém, estas aplicações apenas se limitam a mostrar os pontos de interesse das cidades do país, ao contrário do BestRide que, para além de mostrar os pontos de interesse através de um tour guiado, tem um profissional que conhece o local e a sua história e assim pode explicar mais sobre esses pontos ao cliente durante a viagem.

Explicando um pouco sobre as aplicações citadas:

3.1 Walkbox Self-Guided Tours

A aplicação Walkbox é uma app que tem como objetivo mostrar as principais áreas de interesse histórico existentes em Portugal. Nesta aplicação podemos escolher qual região do país queremos visitar e nos será apresentado todos os roteiros para se fazer naquele local. Além disso podemos acessar fotos, história, mapa, audio tour e mais informações sobre o ponto de interesse.



Figura 1 – Walkbox

A Walkbox é uma aplicação que orienta os seus utilizadores, dando-lhes os roteiros de viagem e as informações sobre os locais de interesse na região. A inovação na aplicação BestRide é que o utilizador, além de ter informações dos roteiros, também poderá contratar um motorista de Tuk-Tuk para conhecer mais os lugares nos quais o cliente pretende visitar.

3.2 TripAdvisor

O TripAdvisor é uma plataforma de avaliações que permite que o usuário possa saber os melhores preços para tudo o que precisa em sua viagem. É a concentração de toda informação que o viajante precisa em um só lugar, como: preços de hotéis, pacotes de viagem, programação cultural, pontos de interesse local, entre outras.

O TripAdvisor é considerado o maior site de viagens do mundo e pode ser acessado pelo navegador (plataforma web), ou pelo telemóvel, através da aplicação.

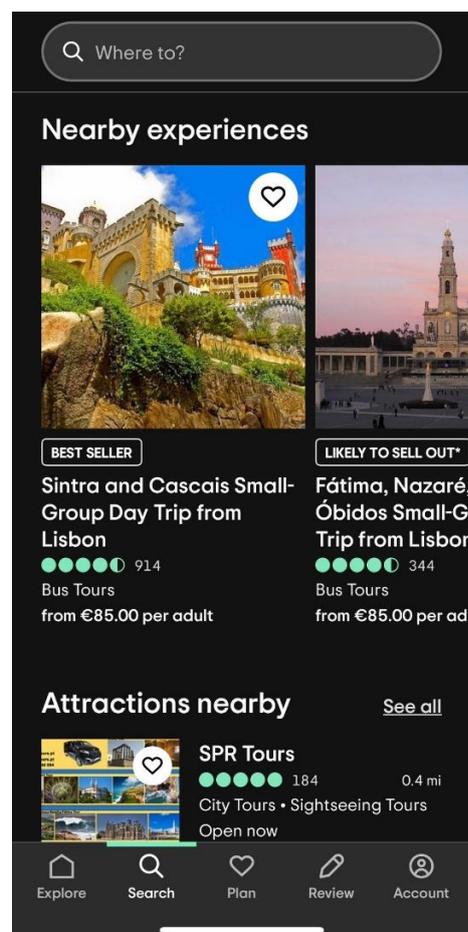


Figura 2 - TripAdvisor

Com isso, novamente afirmamos que não podemos dizer que esta plataforma é uma concorrência direta para o BestRide, pois tem objetivos distintos em relação ao nosso projeto. Contudo, esta é uma ótima plataforma para busca de informações sobre viagens e para a realização de turismo e, sem dúvidas, é uma inspiração para o nosso negócio.

Para além destas aplicações, temos algumas empresas de Tuk-Tuk's que gerem o seu negócio através de websites. Estes websites tem todas as informações sobre o tour que será realizado caso o cliente se sinta interessado. Listamos aqui alguns dos websites que pesquisamos:

- <https://www.lisbon-tuk-tours.com>
- <https://tukdreams.pt>
- <https://www.tukonme.pt>

Portanto, a ideia de negócio da BestRide é inovadora e única no mercado até agora. As empresas que pesquisamos e listamos anteriormente são uma ótima fonte de inspiração para o aperfeiçoamento do nosso negócio, visto que nosso objetivo é sempre continuar inovando a aplicação para tentar melhorar cada vez mais a experiência de uso da mesma, tanto para o cliente quanto para os profissionais da área.

Concluindo iremos apresentar uma tabela de comparação das funcionalidades da nossa aplicação em relação às aplicações citadas acima:

	Informações Turísticas	Tours Guiados	Reservas de Tuk-Tuk's	Orientador Turístico
BestRide	✓	✓	✓	✓
WalkBox	✓	✓	X	X
TripAdvisor	✓	X	✓	X

Tabela 1 - Comparação das Aplicações

4 Engenharia

A especificação da engenharia da aplicação é um processo crítico para garantir que a nossa aplicação seja desenvolvida de forma eficiente. Começamos este processo com o levantamento e análise dos requisitos funcionais e não funcionais, o que nos permite compreender as necessidades dos utilizadores e os objetivos da aplicação. Em seguida, utilizamos ferramentas como diagramas de caso de uso, diagramas de atividade, diagramas de entidade-relação, modelo de classes para modelar o sistema e suas funcionalidades.

Mesmo após a mudança estratégica do projeto, decidimos realizar também os mockups e storyboards da aplicação, que nos permitem visualizar a interface do utilizador e garantir que ela seja intuitiva e fácil de usar. Tomamos essa decisão após conversas com o nosso orientador, pois, além do desenvolvimento do back-end, que se tornou nosso principal objetivo, também gostaríamos de arrumar e melhorar o Front-end da aplicação e, com isso, desenvolvemos mockups para nos orientar em relação ao design que gostaríamos. Já tínhamos um design inicial da aplicação realizada pelos alunos que iniciaram este projeto, porém, após reuniões com a cliente, decidimos reestruturar algumas páginas da aplicação. A diferença entre o design antigo e o novo será mostrado neste tópico. Tudo isso é crucial para garantir que a aplicação atenda às necessidades dos utilizadores e seja fácil de manter e escalar ao longo do tempo.

4.1 Levantamento e análise dos Requisitos

O projeto BestRide, como já dito anteriormente, é uma continuação da realização de um TFC antigo iniciado no ano letivo de 2021/2022. Com isso, sabemos que os colegas que iniciaram o projeto também desenvolveram requisitos e por isso alguns dos nossos requisitos iniciais, sobre funcionalidades padrão da aplicação, são iguais aos deles. Porém, ao ter contato com a cliente durante reuniões, tivemos ainda mais ideias de novas funcionalidades e mudanças para a aplicação. Iremos citar quais os novos requisitos criados por nós. Vale ressaltar que, ao perdermos todos os serviços da cloud anteriormente utilizada, foi necessário o redesenvolvimento do nosso back-end (Django), por isso, tivemos que fazer novos requisitos para a aplicação, que, por mais que alguns foram implementados anteriormente, não foram realizados requisitos para tais funcionalidades. Esses requisitos, juntamente com os requisitos de migração, são os que denominamos como novos requisitos e que gostaríamos de implementar durante e após a realização da migração. Com isso, desenvolvemos alguns requisitos tanto para a aplicação do Turista (AOT) quanto para a aplicação do *Driver* (AOD):

4.1.1 Requisitos Funcionais:

Backlog: BestRide AOT1 Tema: Aplicação Turista
Como Turista Eu quero criar uma conta Para que possa ter acesso á totalidade da aplicação

Backlog: BestRide AOT2 Tema: Aplicação Turista
Como Turista Eu quero efetuar login Para que possa ter acesso á totalidade da aplicação

Backlog: BestRide AOT3 Tema: Aplicação Turista
Como Turista Eu quero criar conta com o Google Para que a criação da conta seja mais rápida

Backlog: BestRide AOT4 Tema: Aplicação Turista
Como Turista Eu quero criar conta com o Facebook Para que a criação da conta seja mais rápida

Backlog: BestRide AOT5 Tema: Aplicação Turista
Como Turista Eu quero aceder ao ecrã Tours Para que possa ver os Tours mais populares e mais perto de mim

Backlog: BestRide AOT6 Tema: Aplicação Turista
Como Turista Eu quero filtrar os Tours Para que possa ver os Tours que correspondam ao meu filtro

Backlog: BestRide AOT7 Tema: Aplicação Turista
Como Turista Eu quero Reservar e Agendar um Tour Para que possa realizar o Tour

Backlog: BestRide AOT8 Tema: Aplicação Turista
Como Turista Eu quero atualizar os meus dados de pagamento Para que possa efetuar uma reserva

Backlog: BestRide **AOT9**
Tema: Aplicação Turista

Como **Turista**
Eu quero **ver os Tours sem criar uma conta**
Para que **seja mais prático**

Backlog: BestRide **AOT10**
Tema: Aplicação Turista

Como **Turista**
Eu quero **avaliar um Tour**
Para que **os outros Turistas tenham um feedback do Tour**

Backlog: BestRide **AOT11**
Tema: Aplicação Turista

Como **Turista**
Eu quero **ser capaz de alterar a password**
Para que **possa atualizar os dados da minha conta**

Backlog: BestRide **AOT12**
Tema: Aplicação Turista

Como **Turista**
Eu quero **poder alterar o email ou telemovel associado a conta**
Para que **os meus dados estejam atualizados**

Backlog: BestRide **AOT13**
Tema: Aplicação Turista

Como **Turista**
Eu quero **poder recuperar a minha password**
Para que **possa fazer login na conta**

Backlog: BestRide **AOT14**
Tema: Aplicação Turista

Como **Turista**
Eu quero **visualizar os meus dados pessoais**
Para que **mante-los atualizados**

Backlog: BestRide **AOT15**
Tema: Aplicação Turista

Como **Turista**
Eu quero **poder alterar o idioma**
Para que **possa ter uma melhor experiência**

Backlog: BestRide **AOT16**
Tema: Aplicação Turista

Como **Turista**
Eu quero **poder entrar em contato com o guia turistico**
Para que **possa tirar dúvidas sobre o Tour**

Backlog: BestRide **AOT17**
Tema: Aplicação Turista

Como **Turista**
Eu quero **visualizar os detalhes de um Tour**
Para que **possa saber mais sobre o Tour**

Backlog: BestRide **AOT18**
Tema: Aplicação Turista

Como **Turista**
Eu quero **definir quantas pessoas vão no Tour**
Para que **o guia se adapte ao tamanho do grupo**

Backlog: BestRide **AOD1**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **criar uma conta**
Para que **possa aceder á plataforma**

Backlog: BestRide **AOD2**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **fazer login**
Para que **possa aceder á plataforma**

Backlog: BestRide **AOD3**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **criar conta com o Google**
Para que **a criação da conta seja mais rápida**

Backlog: BestRide **AOD4**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **criar conta com o Facebook**
Para que **a criação da conta seja mais rápida**

Backlog: BestRide **AOD5**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **ser capaz de alterar a password**
Para que **possa atualizar os dados da minha conta**

Backlog: BestRide **AOD6**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **poder recuperar a minha password**
Para que **possa fazer login na conta**

Backlog: BestRide **AOD7**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **poder alterar o idioma**
Para que **possa ter uma melhor experiência**

Backlog: BestRide **AOD8**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **visualizar os meus dados pessoais**
Para que **mante-los atualizados**

Backlog: BestRide **AOD9**
Tema: Aplicação Driver

Como **Empresa Turística**
Eu quero **criar um Tour**
Para que **os turistas possam realizar o meu Tour**

Backlog: BestRide **AOD10**
Tema: Aplicação Driver

Como **Empresa Turística**
Eu quero **adicionar fotos ao Tour**
Para que **os turistas tenham mais informações sobre o Tour**

Backlog: BestRide **AOD11**
Tema: Aplicação Driver

Como **Empresa Turística**
Eu quero **definir a duração do Tour**
Para que **os turistas tenham a duração do Tour**

Backlog: BestRide **AOD12**
Tema: Aplicação Driver

Como **Empresa Turística**
Eu quero **definir quais os veiculos disponiveis no Tour**
Para que **os turistas possam escolher o veiculo**

Backlog: BestRide **AOD13**
Tema: Aplicação Driver

Como **Empresa Turística**
Eu quero **saber quantas pessoas vão no Tour**
Para que **possa definir a frota a levar e que Motoristas atribuir**

Backlog: BestRide **AOD14**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **inserir as línguas que falo**
Para que **o turista tenha uma melhor experiência**

Backlog: BestRide **AOD15**
Tema: Aplicação Driver

Como **Empresa Turística**
Eu quero **adicionar detalhes do Tour**
Para que **os turistas tenham mais informações sobre o Tour**

Backlog: BestRide **AOD16**
Tema: Aplicação Driver

Como **Empresa Turística**
Eu quero **definir o preço do Tour**
Para que **os turistas saibam o preço do Tour**

Backlog: BestRide **AOD17**
Tema: Aplicação Driver

Como **Guia Turístico**
Eu quero **ver os meus Tours futuros**
Para que **orientar o meu trabalho**

4.1.2 Requisitos Não-Funcionais:

Backlog: BestRide Tema: Não Funcionais	NOF1
Como Utilizador Eu quero que a aplicação seja rápida e responsiva Para que eu possa navegar facilmente pela aplicação	

Backlog: BestRide Tema: Não Funcionais	NOF2
Como Utilizador Eu quero que a aplicação seja fácil de usar Para que eu possa encontrar as informações e realizar as tarefas de forma intuitiva	

Backlog: BestRide Tema: Não Funcionais	NOF3
Como Utilizador Eu quero que a aplicação seja segura e protegida contra ameaças externas Para que as minhas informações pessoais e dados estejam seguros.	

Backlog: BestRide Tema: Não Funcionais	NOF4
Como Utilizador Eu quero eu quero que a aplicação seja compatível com a maioria dos sistemas operacionais móveis Para que possa usá-la independentemente do dispositivo	

Backlog: BestRide Tema: Não Funcionais	NOF5
Como Utilizador Eu quero que a aplicação tenha suporte para diferentes idiomas Para que possa usá-la independentemente da minha língua nativa	

<p>Backlog: BestRide NOF6 Tema: Não Funcionais</p>	<p>Backlog: BestRide NOF7 Tema: Não Funcionais</p>
<p>Como Utilizador Eu quero que a aplicação tenha uma boa performance de bateria Para que a bateria do meu dispositivo não acabe mais rápido</p>	<p>Como Utilizador Eu quero que a aplicação carregue rápido Para que possa utiliza-la o mais rápido possível</p>
<p>Backlog: BestRide NOF8 Tema: Não Funcionais</p>	<p>Backlog: BestRide NOF9 Tema: Não Funcionais</p>
<p>Como Utilizador Eu quero que a aplicação tenha uma opção de atualização automática Para que esteja sempre atualizada</p>	<p>Como Utilizador Eu quero que a aplicação tenha gráficos atraentes e design moderno Para que ela seja agradável de usar</p>

4.1.2.1 Requisitos de migração:

MIG01	Como desenvolvedor, eu quero ter uma base de dados que suporte toda a aplicação, para guardar todas as informações necessárias
MIG02	Como desenvolvedor, eu quero que minha base de dados seja segura, para que eu não perca as informações
MIG03	Como desenvolvedor, eu quero que o serviço hospedado no Azure seja capaz de se adaptar a mudanças na demanda, para garantir uma escalabilidade eficiente e uma resposta ágil
MIG04	Como desenvolvedor, eu quero que os gastos com os serviços do Azure sejam otimizados, para identificar e corrigir potenciais desperdícios de recursos

MIG05	Como desenvolvedor, eu quero garantir a integridade e a consistência dos dados durante todo o processo de migração, para garantir que não ocorram perdas ou corrupção de dados
MIG06	Como desenvolvedor, eu quero que seja implementado um plano de backup e recuperação de dados no ambiente do Azure, para garantir a disponibilidade e a integridade dos dados migrados
MIG07	Como desenvolvedor, eu quero garantir o gerenciamento seguro de chaves e criptografia dos dados migrados no ambiente do Azure, para garantir a proteção adequada dos recursos e dados migrados
MIG08	Como desenvolvedor, eu quero garantir o controle de acesso e permissões adequados para os recursos e dados no ambiente do Azure, para garantir a integridade e a qualidade dos dados após a migração
MIG09	Como desenvolvedor, eu quero que seja realizado um teste completo de validação dos dados migrados no ambiente do Azure, para garantir a integridade dos dados

Tabela 2 - Tabela de Requisitos da Migração

A coluna “Novo Requisito” é baseada apenas nos requisitos realizados pelos colegas do ano passado, segundo o easyBackLog fornecido pelo orientador.

Requisitos	Nível de Importância (1-10)	Esforço esperado (1-10)	Novo Requisito	Implementação
AOT1	10	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT2	10	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT3	6	6	<input type="checkbox"/>	<input type="checkbox"/>
AOT4	6	6	<input type="checkbox"/>	<input type="checkbox"/>
AOT5	10	7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

AOT6	7	7	<input checked="" type="checkbox"/>	Parcialmente
AOT7	10	8	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT8	10	7	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT9	8	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOT10	8	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT11	10	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT12	8	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOT13	10	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT14	10	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT15	10	7	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOT16	6	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOT17	9	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOT18	8	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD1	10	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOD2	10	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD3	6	6	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AOD4	6	6	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AOD5	10	4	<input type="checkbox"/>	<input checked="" type="checkbox"/>
AOD6	10	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD7	10	7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD8	10	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD9	10	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD10	8	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD11	10	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

AOD12	10	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD13	10	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD14	8	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD15	10	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD16	10	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
AOD17	8	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF1	10	8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF2	8	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF3	10	8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF4	10	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF5	10	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF6	8	9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF7	7	8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF8	5	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
NOF9	10	5	<input checked="" type="checkbox"/>	Parcialmente
MIG01	10	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG02	10	7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG03	10	3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG04	10	7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG05	10	9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG06	10	9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG07	10	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG08	10	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MIG09	10	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Tabela 3 - Tabela Requisitos

Requisitos	Nível de Importância (1-10)	Não implementado / Parcialmente	Explicação
AOT3	6	Não implementado	Não foi possível a implementação da autenticação com a Google pois não utilizamos o oauth2 que seria necessário para a implementação
AOT4	6	Não implementado	Não foi possível a implementação da autenticação com o Facebook pois não utilizamos o oauth2 que seria necessário para a implementação
AOT6	7	Parcialmente implementado	O filtro “Perto de mim” não funciona, tentamos implementar, mas não obtivemos o resultado que gostaríamos
AOT7	10	Implementado apenas no back-end	Não implementamos no front-end mas está implementado no back-end
AOT10	8	Implementado apenas no back-end	Back-end <input checked="" type="checkbox"/> Front-end <input type="checkbox"/>

AOT16	6	Implementado apenas no back-end	Back-end <input checked="" type="checkbox"/> Front-end <input checked="" type="checkbox"/>
AOT17	9	Implementado apenas no back-end	Back-end <input checked="" type="checkbox"/> Front-end <input checked="" type="checkbox"/>
AOD3	6	Não implementado	Não foi possível a implementação da autenticação com a Google pois não utilizamos o oauth2 que seria necessário para a implementação
AOD4	6	Não implementado	Não foi possível a implementação da autenticação com o Facebook pois não utilizamos o oauth2 que seria necessário para a implementação
AOD8, AOD9, AOD10, AOD11, AOD12, AOD13, AOD15, AOD16	78	Implementados apenas no back-end	Back-end <input checked="" type="checkbox"/> Front-end <input checked="" type="checkbox"/>
AOD17	9	Implementado apenas no back-end	Back-end <input checked="" type="checkbox"/> Front-end <input checked="" type="checkbox"/>
NOF4	10	Não implementado	Pelo facto do front-end não estar implementado não é possível a realização deste requisito
NOF6	8	Não implementado	Não conseguimos

			testar esse requisito
NOF8	10	Não implementado	Não desenvolvemos esta funcionalidade
NOF9	10	Parcialmente implementado	Alteramos alguns ecrãs da aplicação para terem um design mais simples e moderno, mas não conseguimos realizar isso para toda a aplicação

Tabela 4 - Requisitos não implementados / parcialmente implementados

Os requisitos AOD8 – AOD16 estão juntos pois fazem parte da mesma categoria de requisitos (criação de tours) .

Para a realização do cálculo do valor obtido tendo por base a classificação de importância com a realização ou não dos requisitos, vamos admitir os seguintes pontos:

- Importância do requisito = v
- Requisito implementado = v * 1
- Requisito parcialmente implementado = v / 2
- Requisito não implementado = 0
- Resultado = média dos valores

Como o nosso objetivo principal era o back-end, vamos assumir que os requisitos implementados apenas no back-end são considerados como requisitos implementados e, por isso, tem o seu valor completo v. Portanto:

Apreciação de valor obtido = 8.0

4.2 Diagramas de Casos de Uso

Os diagramas de caso de uso são ferramentas fundamentais no desenvolvimento de um projeto, pois fornecem uma visão geral dos requisitos funcionais do sistema e como os utilizadores interagem com ele. Eles são utilizados para modelar os casos de uso, ou seja, as tarefas que os utilizadores precisam realizar com o sistema.

Os diagramas de caso de uso nos permitem identificar as funcionalidades do sistema, bem como as interações entre os atores (usuários e sistemas). Isso nos ajuda a compreender as necessidades dos utilizadores e a garantir que o sistema esteja alinhado com essas necessidades. Além disso, os diagramas de caso de uso também nos ajudam a identificar quaisquer problemas ou ambiguidades nos requisitos, antes mesmo do início do desenvolvimento. Com isso, desenvolvemos dois diagramas de casos de uso, um para a aplicação do Turista e outro para a aplicação do *Driver*:

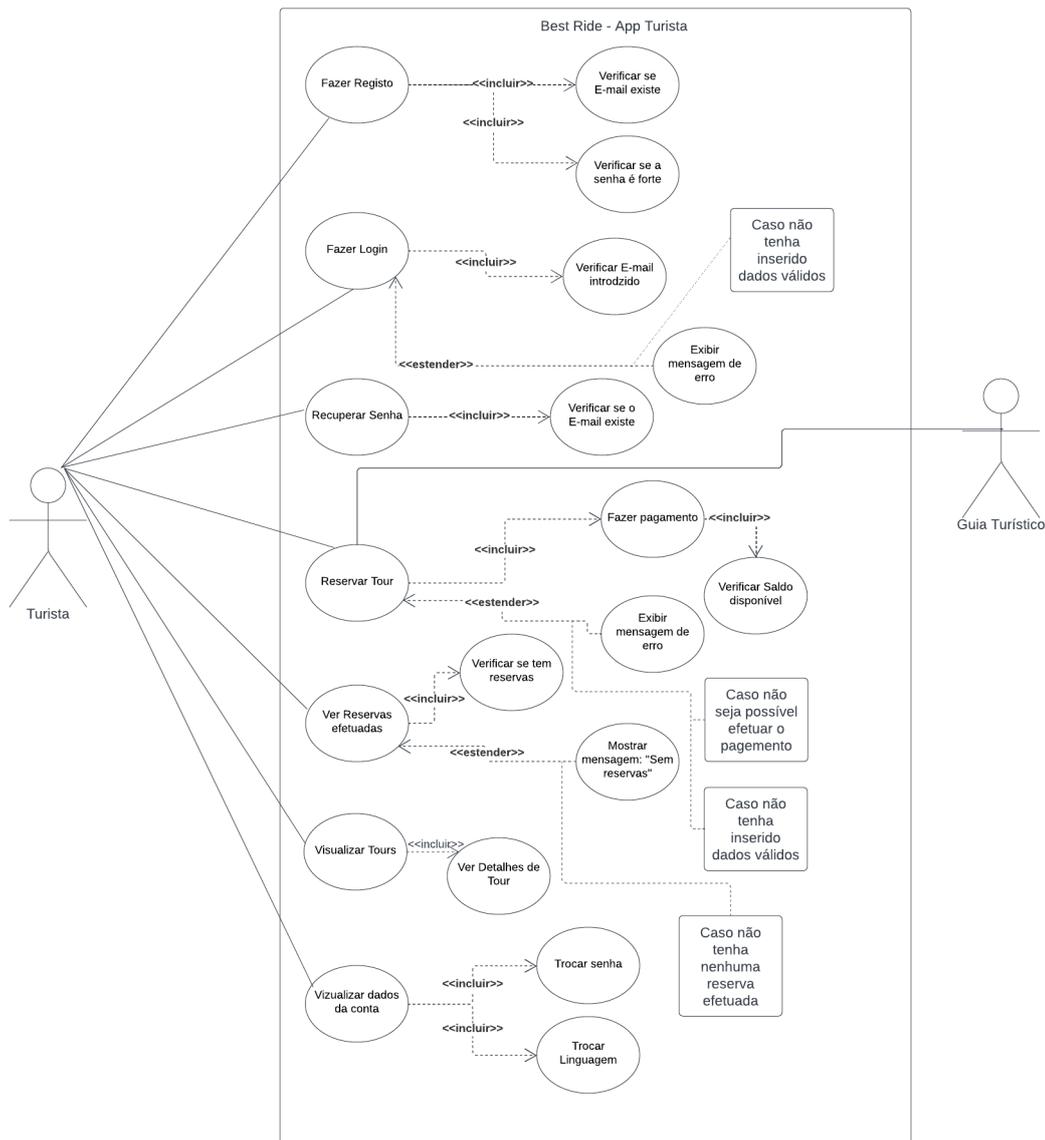


Figura 3 - Diagrama de Caso de Uso App Turista

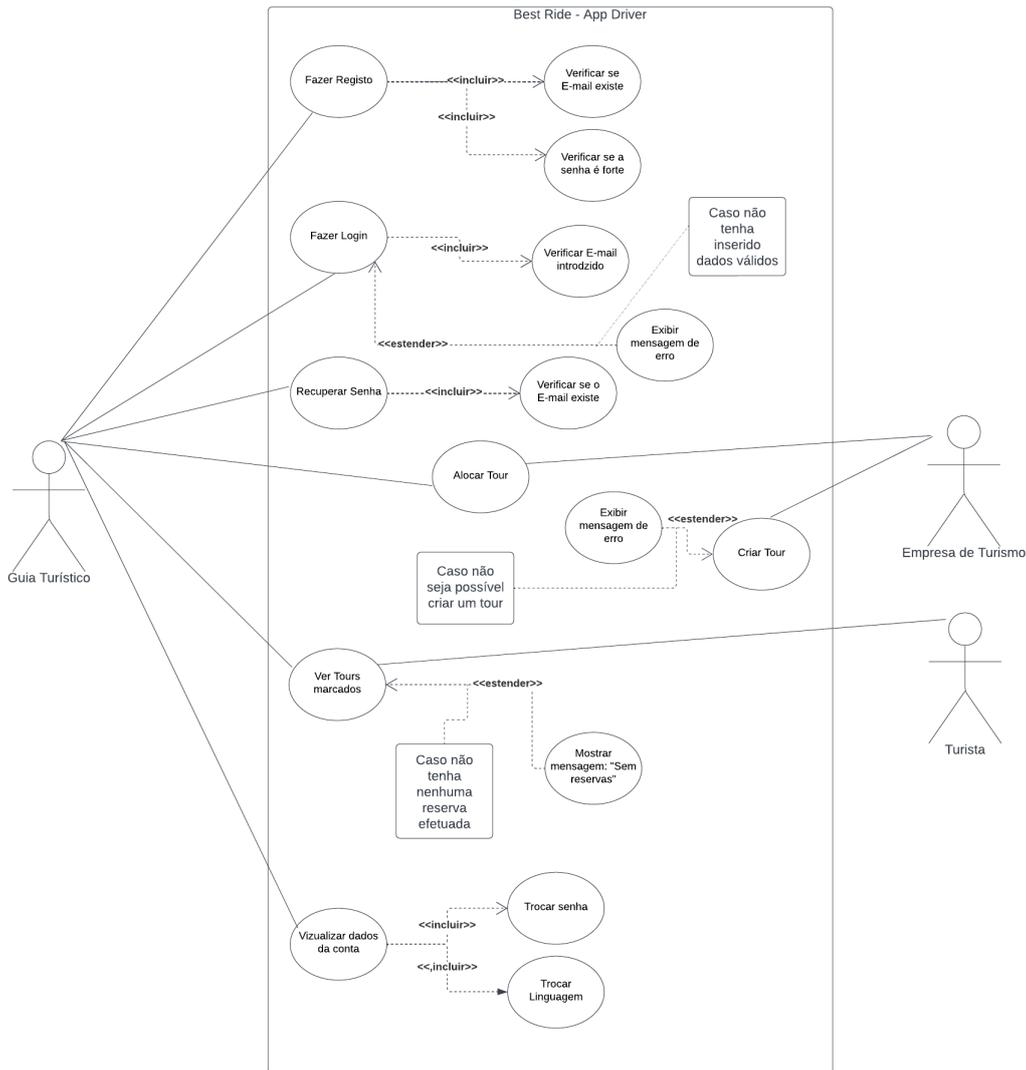


Figura 4 - Diagrama de Caso de Uso App Driver

4.3 Diagramas de Atividades

Os diagramas de atividade são ferramentas importantes na modelagem e documentação de processos de negócios e sistemas. Eles permitem representar visualmente a sequência de atividades e decisões que ocorrem em um processo, bem como os participantes envolvidos e os objetos de dados utilizados.

Além disso, os diagramas de atividade também podem ser utilizados para identificar pontos de melhoria no processo. Isso pode ajudar a evitar erros e atrasos no projeto, e aumentar a eficiência deste processo.

O desenvolvimento destes diagramas é muito pertinente para a nossa aplicação e por isso desenvolvemos dois diagramas de atividade com os processos principais da nossa aplicação: a Reserva de um Tour e a Criação de um Tour:

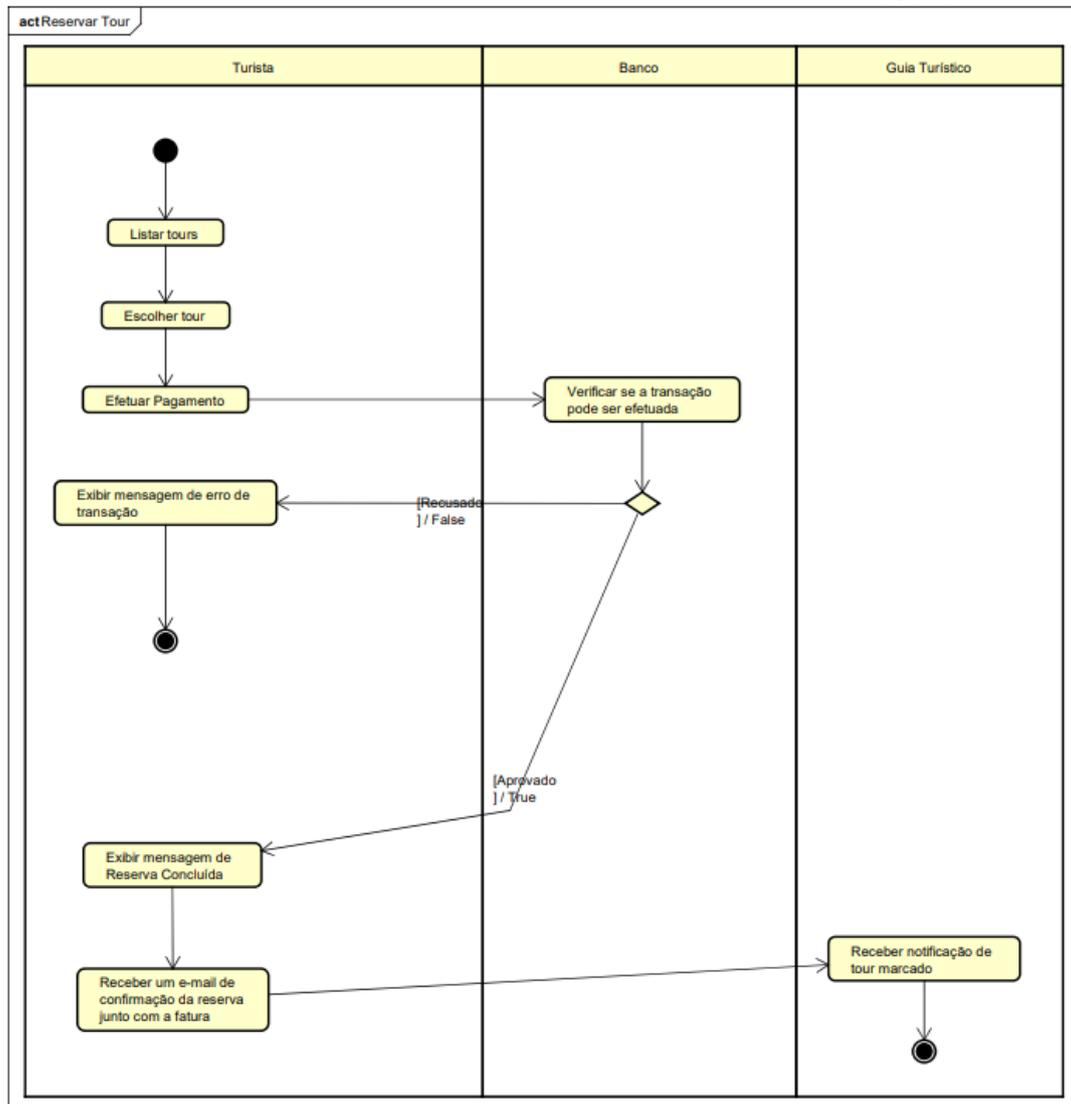


Figura 5 - Diagrama de Atividade Reservar Tour

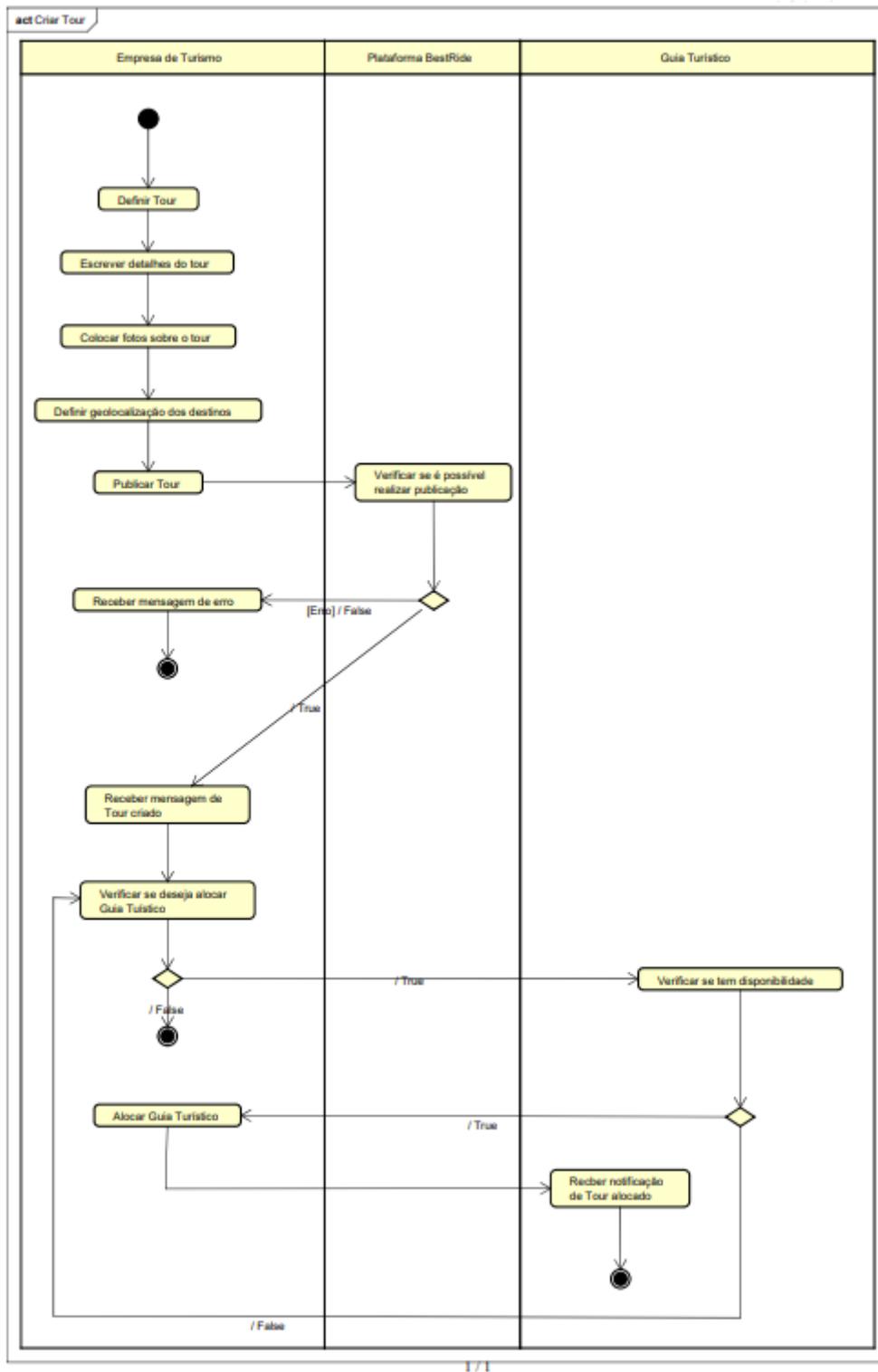


Figura 6 - Diagrama de Atividade Criar Tour

4.4 Modelos Relevantes

Neste tópico iremos colocar alguns diagramas e modelos relevantes que fizemos para o desenvolvimento do nosso projeto. Estes modelos são:

- Diagrama Entidade – Relação
- Modelo de Classes
- Diagrama de Sequência

A utilização destas técnicas de modelagem permitem uma melhor compreensão da estrutura lógica do sistema da nossa aplicação, facilitando a identificação de problemas e garantindo a qualidade e manutenibilidade do projeto.

4.4.1 Diagrama Entidade - Relação

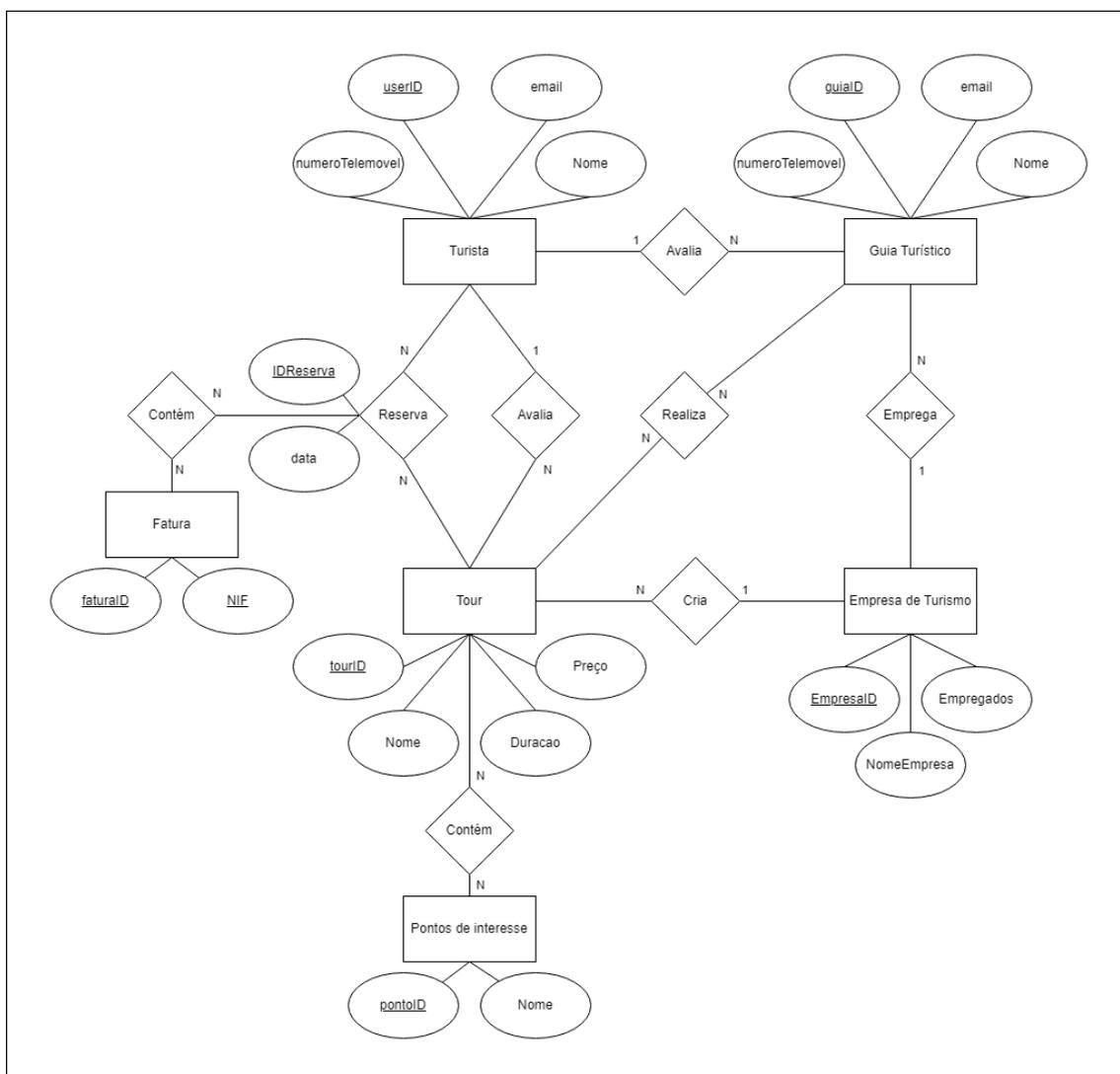


Figura 7 - Diagrama Entidade – Relação

4.4.2 Diagrama de Classes

Alterámos o nosso diagrama de classe face à nossa base de dados atual. Com isso o novo diagrama criado é:

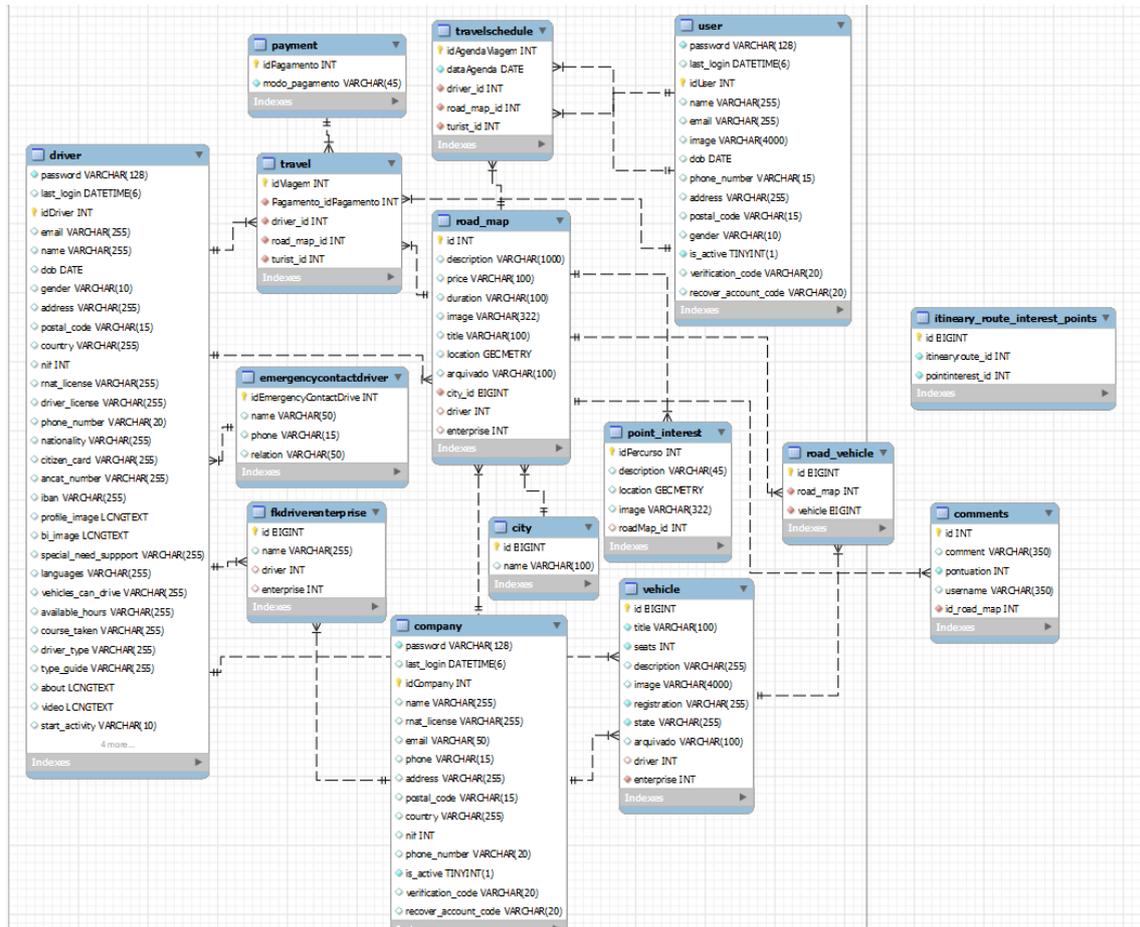


Figura 8 - Diagrama de Classes

4.4.3 Diagramas de Sequência

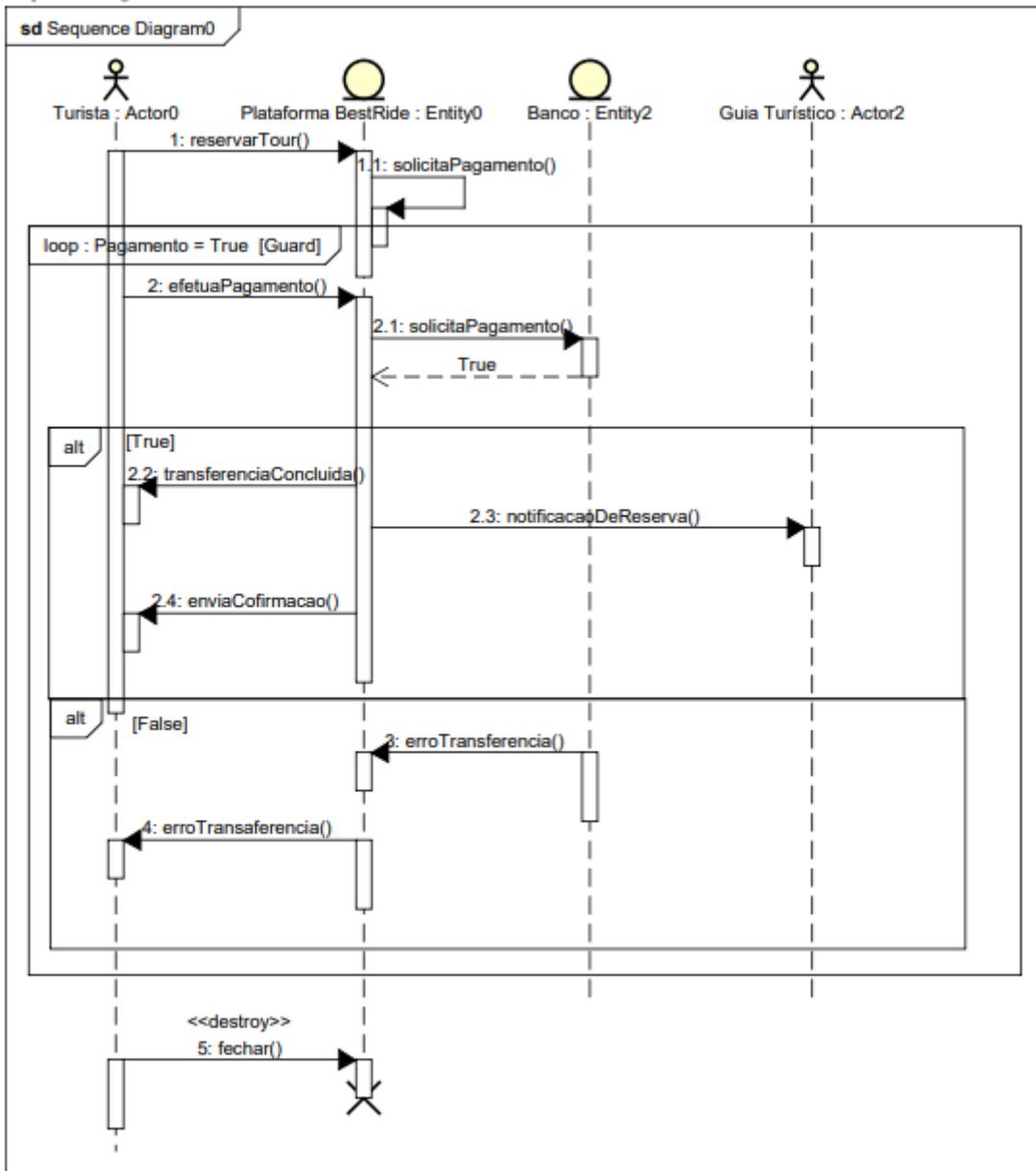


Figura 9 - Diagrama de Sequência Reservar Tour

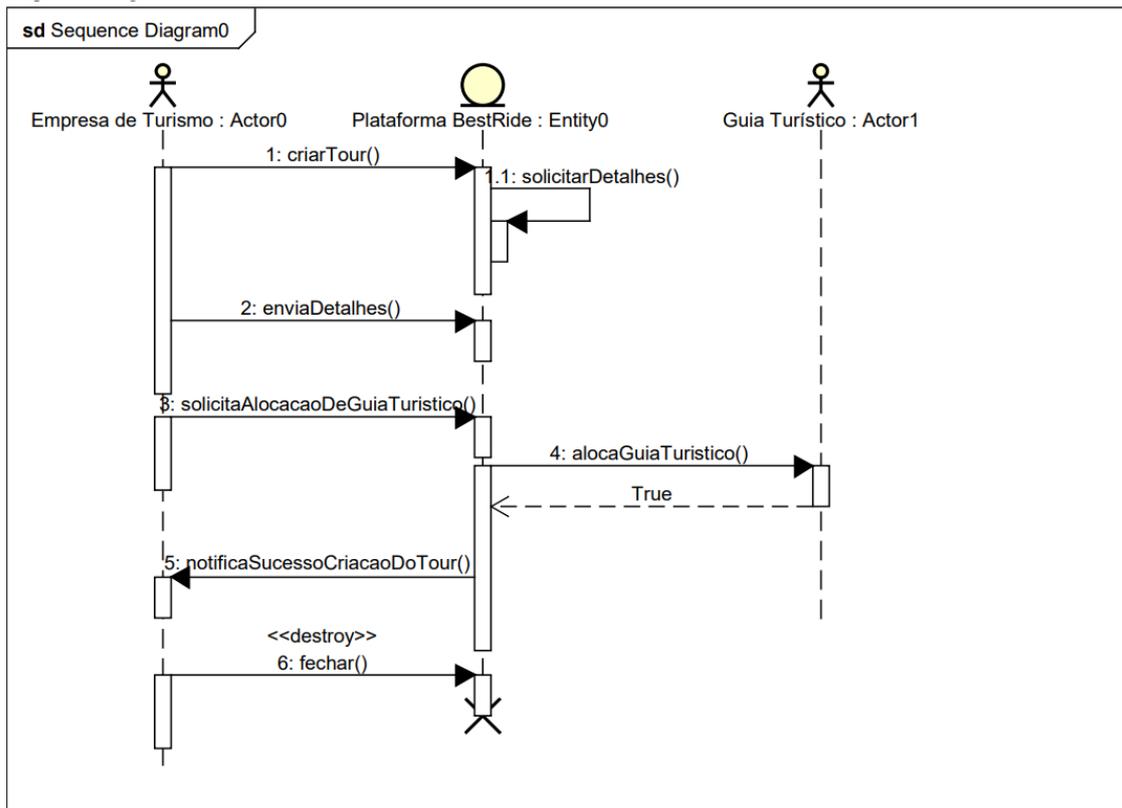


Figura 10 - Diagrama de Sequência Criar Tour

4.5 Estrutura (*Mockups*, *StoryBoards* e Mapa Aplicacional)

Os mockups, o storyboard e o mapa aplicativo são ferramentas importantes para a concepção e planejamento de interfaces de usuário (UI) e experiência do usuário (UX) em um projeto. Além disso, também são importantes para garantir que a interface e a experiência do usuário sejam projetadas de forma intuitiva e fácil de usar, identificando problemas de projeto e garantindo a qualidade do mesmo antes da sua implementação.

4.5.1 Mockups

Login



BestRide

E-mail

Password

Login

Não tem conta? Crie uma agora!

Recupere sua senha

 Sign in with Facebook

 Sign in with Google

Figura 11 - Mockup Página Login

Create

Nome
Insira o seu nome

Data de nascimento
DD/MM/YYYY

Número telefone
+351 91xxxxxxx

Género
Género

E-mail
Insira o seu E-mail

Password
Password

Repita a password
Password

Criar conta

Figura 12 - Mockup Página de Registo

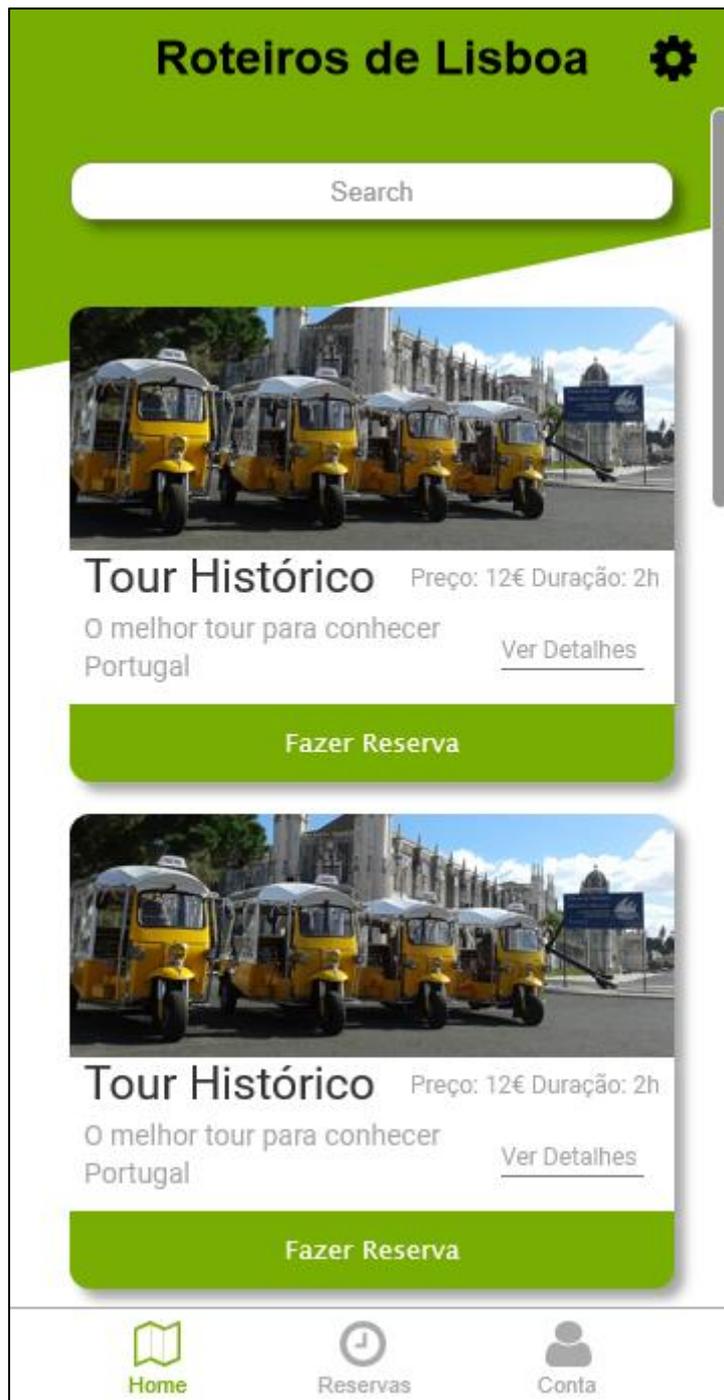


Figura 13 - Mockup Página Inicial



Tour Histórico

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Convalis tellus id interdum velit laoreet id donec ultrices. Dignissim diam quis enim lobortis scelerisque fermentum dui. Tempor orci dapibus ultrices in. Non nisi est sit amet facilisis magna etiam tempor. Enim nulla aliquet porttitor lacus. Enim sed faucibus turpis in eu.

Preço: 12€ Duração: 2h

Guia Turístico:

 João Almeida 4.8/5 ★
English, French, Portuguese, Italian

Reservar

Figura 14 - Mockup Detalhes do Tour

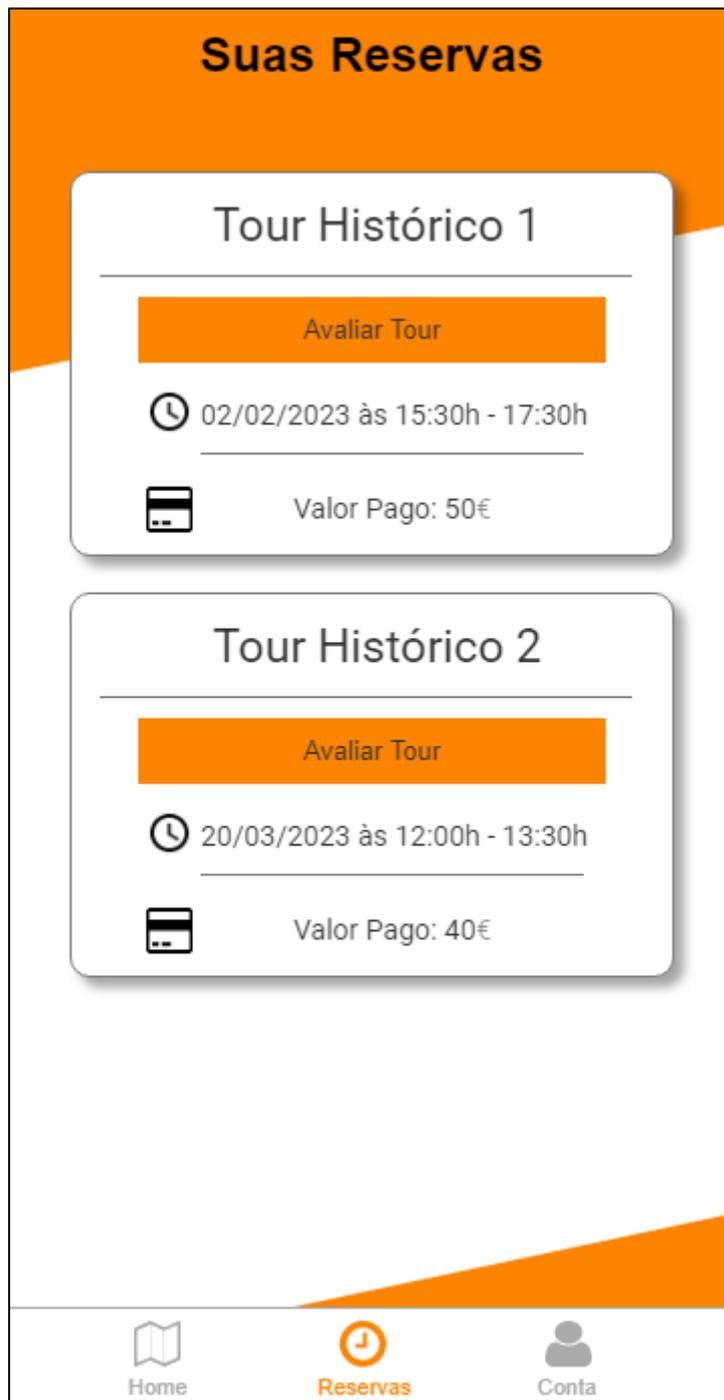


Figura 15 - Mockup Página de Reservas

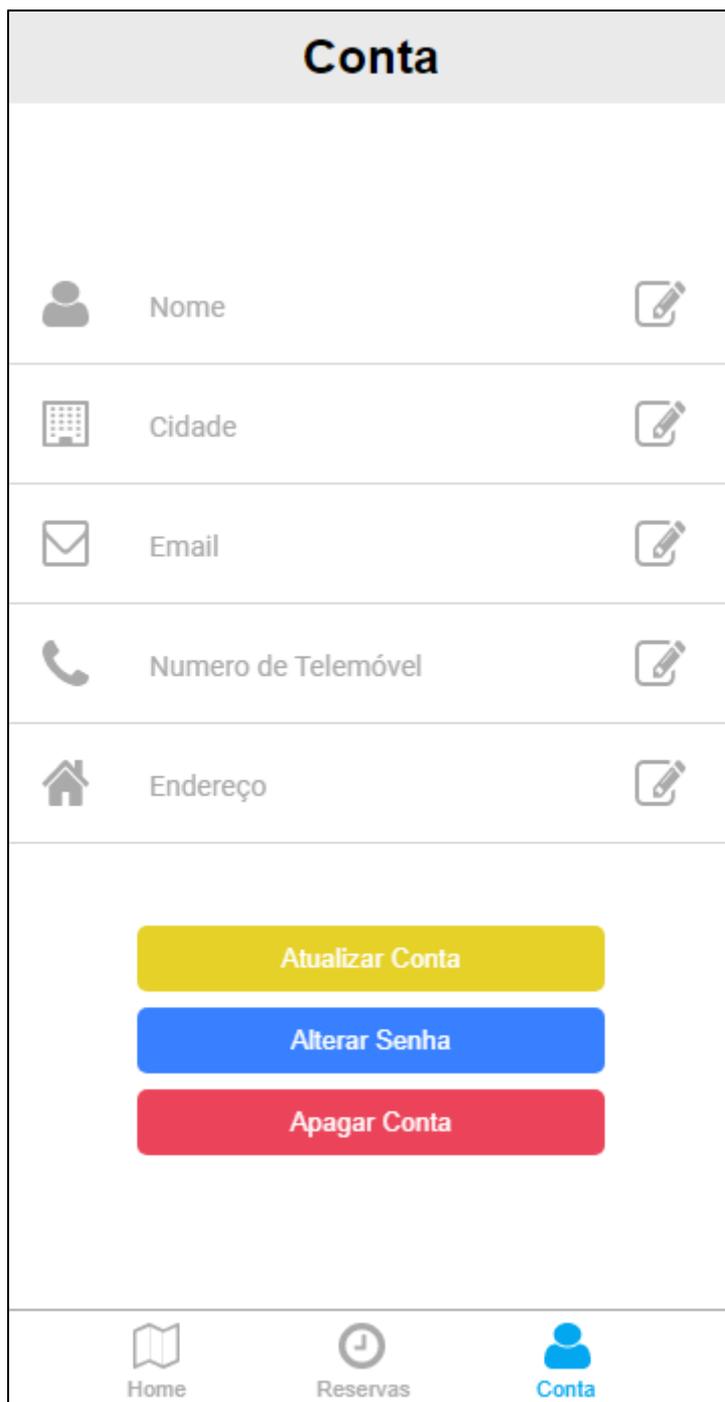


Figura 16 - Mockup Página Conta do Usuário

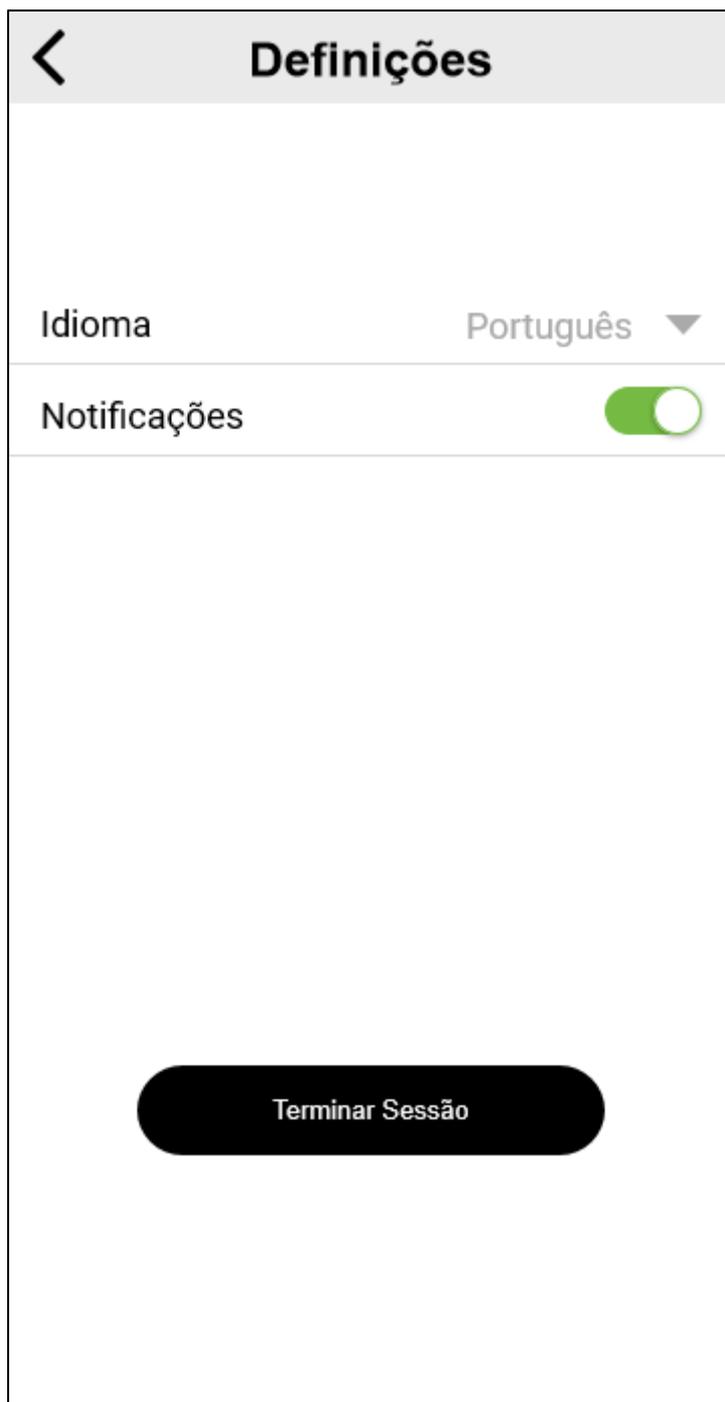


Figura 17 - Mockup Página de Definições

×

Reservar Tour

Data do Tour	DD/MM/YYYY
Quantas pessoas?	1 ▼
Agendar horário	10:30
Veículo	▼

Preço : 10€
Duração : 2h

Reservar

Figura 18 - Mockup Página Reservar Tour

4.5.2 StoryBoard



Figura 19 – StoryBoard

4.5.3 Mapa Aplicacional

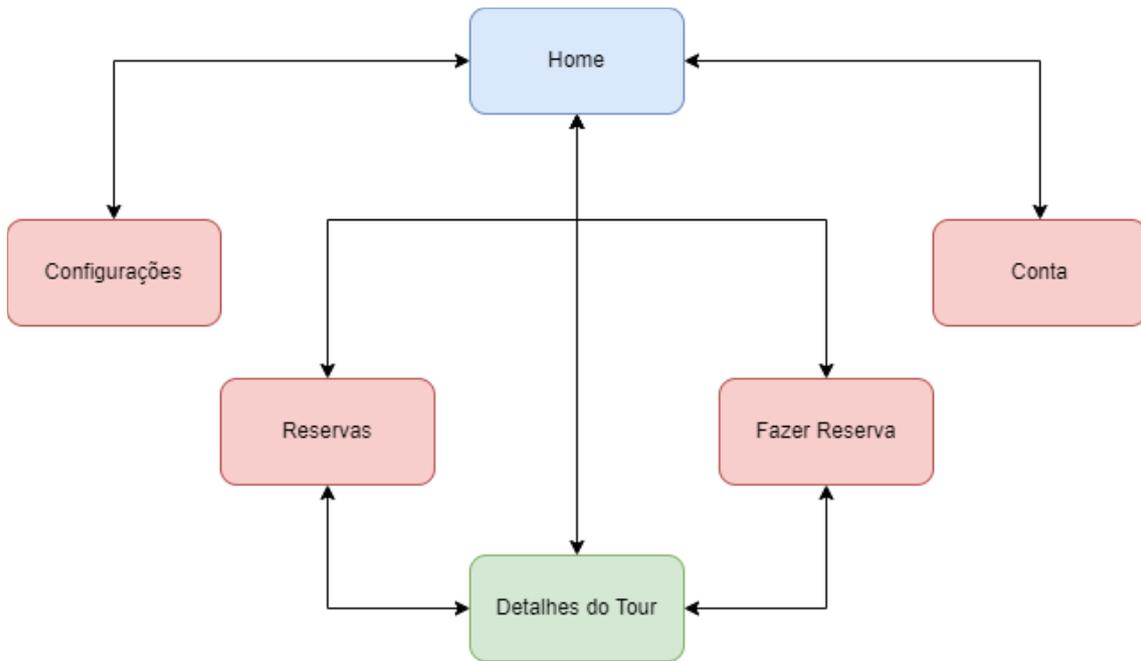


Figura 20 - Mapa Aplicacional BestRide

5 Solução Proposta

5.1 Introdução

Na presente seção iremos apresentar a solução proposta para o desenvolvimento das nossas aplicações (Aplicação Turista e Aplicação Driver). Iremos apresentar a arquitetura da nossa aplicação, assim como, as tecnologias que já foram utilizadas no desenvolvimento inicial da aplicação pela equipa anterior e que tivemos que alterá-las para dar continuidade ao projeto além de também apresentar a forma de implementação dessas tecnologias.

Aqui iremos colocar o link do vídeo demonstrativo do ambiente de desenvolvimento da nossa aplicação. Como já referido anteriormente, o âmbito do nosso projeto se tornou a migração e o desenvolvimento completo do back-end da aplicação. Com isso o nosso vídeo demonstrativo está voltado a demonstrar o funcionamento das funções implementadas do back-end, para isso utilizamos o postman, uma plataforma de colaboração e teste de APIs que permite-nos, no nosso caso, testar todas as funções que precisamos.

Posteriormente explicaremos como realizamos a migração do projeto para o Azure e como realizamos desenvolvimento completo do back-end e parcial do front-end.

- Ip de acesso à aplicação turista
51.124.8.14
- Ip de acesso à aplicação driver
20.76.98.7

Nota: Estes dois Ip's irão direcionar o utilizador para cada uma das aplicações. Contudo, o deployment do back-end não foi 100% concluído no azure, o que impossibilita a utilização da aplicação de forma correta por conta da falta dos serviços do back-end. Iremos explicar o problema do deployment ainda neste tópico porém podemos adiantar que o deployment do front-end foi bem-sucedido.

- Link para o vídeo demonstrativo:
<https://www.youtube.com/watch?v=02Mwst2zu-k>

Link dos repositórios Git:

- Frontend-Turista:
<https://github.com/diogocerqueira-22002160/BestRideTurista>
<https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI300-BestRide-Turista>
- Frontend-Driver:

<https://github.com/diogocerqueira-22002160/BestRideDriver>

<https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI300-BestRide-Driver>

- Backend:

<https://github.com/diogocerqueira-22002160/BestRideBackend>

<https://github.com/DEISI-ULHT-TFC-2022-23/TFC-DEISI300-BestRide-Backend>

5.2 Tecnologias e ferramentas utilizadas

5.2.1 Microsoft Azure

Microsoft Azure é uma plataforma de computação em nuvem do Microsoft que permite aos desenvolvedores criar, gerenciar e escalar aplicativos e soluções em nuvem. Ele oferece uma ampla variedade de serviços, incluindo computação, armazenamento, base de dados, inteligência artificial e aprendizado de máquina, gerenciamento de identidade e segurança, entre outros.

Essa plataforma permite que os desenvolvedores criem aplicações e soluções escaláveis, altamente disponíveis e seguras, usando sua escolha de linguagens de programação, ferramentas e frameworks. Ele também oferece recursos de automação e implantação contínua, tornando fácil implementar e gerenciar aplicativos e soluções em produção. Além disso, ele oferece integração com outros serviços e ferramentas da Microsoft, como o Office 365, Dynamics 365 e Power Platform, permitindo aos desenvolvedores aproveitar ao máximo a plataforma.

Além disso, como parte do nosso projeto, utilizamos os serviços do Azure Blob Storage para realizar o upload de imagens para a nuvem. Armazenamos essas imagens em containers específicos, garantindo a organização e o acesso seguro às imagens. Também utilizamos o Azure Translator para implementar a alteração de linguagens na aplicação. Para demonstração, aqui temos uma função na qual o seu objetivo é dar upload à imagem para um container no Azure Blob-Storage, Caso esse container não exista, a função cria-o no ambiente:

```

1 usage  + diogocerqueira-22002160
@api_view(['PUT'])
def updateImageUser(request, email):
    driver = Driver.objects.get(email=email)
    storage_connection_string = env.str('AZURE_STORAGE_CONNECTION_STRING')
    blob_service_client = BlobServiceClient.from_connection_string(storage_connection_string)

    container_name = 'imagesdriver'

    try:
        blob_service_client.create_container(container_name)
    except ResourceExistsError:
        pass

    file = request.FILES['image']
    file_extension = '.' + file.name.split('.')[-1]
    file_name = f"{driver.idDriver}/{request.data.get('name')}{file_extension}"
    file_data = file.read()

    blob_client = blob_service_client.get_blob_client(container_name, file_name)
    try:
        blob_client.upload_blob(file_data, overwrite=True)
    except Exception as e:
        print(str(e))

    if(request.data.get('name') == 'profile_image'):

        driver.profile_image = blob_client.url
        driver.save()
        return Response(driver.profile_image)
    elif(request.data.get('name') == 'bi_image'):
        driver.bi_image = blob_client.url
        driver.save()
        return Response(driver.bi_image)

```

Figura 21 - Função de upload de imagens Azure Blob-Storage

Para fins de comparação, aqui está a função utilizada pelos alunos do ano anterior que não estava devidamente implementada, e, conseqüentemente, não funcionava:

```

@api_view(['PUT'])
def updateImageUser(request, email):
    tutorial = User.objects.get(email=email)
    tutorial_data = JSONParser().parse(request)
    tutorial_serializer = UserSerializer(tutorial, data=tutorial_data)
    if tutorial_serializer.is_valid():
        tutorial_serializer.save()
        return JsonResponse(tutorial_serializer.data)
    return JsonResponse(tutorial_serializer.errors, status=status.HTTP_400_BAD_REQUEST)

```

Figura 22 - Função de upload de imagens para a AWS do TFC anterior

Além disso, temos também a função de tradução automática implementada com o serviço do Azure Translator:

```
1 usage  ➤ erick +1
@api_view(['POST'])
def translate(request):
    subscription_key = 'd282af8ef2784b829c28173a468cc593'
    endpoint = 'https://api.cognitive.microsofttranslator.com/'

    location = 'westeurope'

    constructed_url = endpoint + '/translate'

    headers = {
        'Ocp-Apim-Subscription-Key': subscription_key,
        'Ocp-Apim-Subscription-Region': location,
        'Content-Type': 'application/json',
        'X-ClientTraceId': str(uuid.uuid4())
    }

    text = request.data['text']
    source_lang = request.data['sourceLang']
    target_lang = request.data['outputLang']

    params = {
        'api-version': '3.0',
        'to': target_lang,
        'from': source_lang
    }

    data = [{
        'text': text
    }]

    request = requests.post(constructed_url, params=params, headers=headers, json=data)
    response = request.json()

    response_data = {
        'translated_text': response[0]['translations'][0]['text']
    }

    return Response(response_data)
```

Figura 23 - Função de translate da aplicação

Novamente, comparando a implementação das funções, aqui temos a função de translate que aproveitava dos serviços da AWS:

```

class TranslateAWS():

    @api_view(['POST'])
    def translate(request):
        client = boto3.client('translate')
        response = client.translate_text(
            Text=request.data['text'], SourceLanguageCode=request.data['sourceLang'],
            TargetLanguageCode=request.data['outputLang'])

        return JsonResponse({
            "translated_text": response['TranslatedText']
        })

```

Figura 24 - Função de translate com AWS TFC anterior

5.2.2 Docker

O Docker é uma plataforma *open source* que facilita a criação e administração de ambientes isolados. Ele possibilita o empacotamento de uma aplicação ou ambiente dentro de um *container*, se tornando portátil para qualquer outro *host* que contenha o Docker instalado. Então, você consegue criar, implantar, copiar e migrar de um ambiente para outro com maior flexibilidade. A ideia do Docker é utilizar apenas uma máquina, ao invés de várias. E, nessa única máquina, podemos colocar várias aplicações a funcionar sem que haja conflitos entre elas.

Utilizamos o Docker para virtualizar a aplicação num container em uma máquina com o sistema operativo (Amazon Linux 2), num EC2 com um servidor, para correr a aplicação num link de forma a poder entregar as soluções aos clientes.

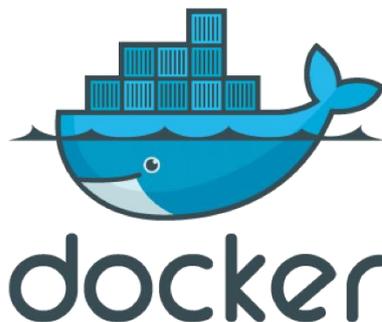


Figura 25 - Docker

5.2.3 Ionic

O Ionic é uma Framework para criação de aplicações móveis e desktop de alto desempenho e qualidade, que utiliza as tecnologias web HTML, CSS e JavaScript. Além disso, apesar de ser um framework híbrido, ele trabalha bem com os hardwares nativos dos dispositivos (Gps, Camera e mais).

Porque Ionic?

O Ionic é multiplataforma, isto é, podemos desenvolver um único código que pode ser executado em diferentes sistemas operacionais, diminuindo o tempo de desenvolvimento da aplicação, assim diminuindo também os custos da aplicação.

O Ionic foi utilizado na nossa aplicação para o desenvolvimento do front-end. Com esta framework foram construídas as páginas da aplicação que serão utilizadas pelos profissionais e clientes (User Interface).

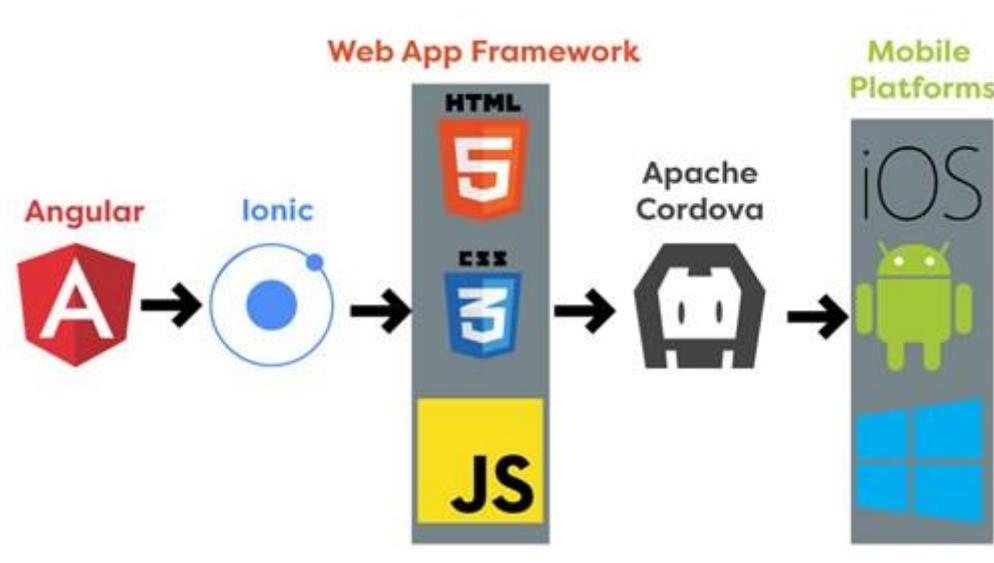


Figura 26 - Arquitetura de uma aplicação Ionic

5.2.4 Stripe

A Stripe é uma das principais instituições de pagamento que gerenciam empresas que atuam na internet. Sua operação está voltada para o desenvolvimento de ferramentas poderosas e flexíveis para o comércio online.

Por meio de APIs cuidadosamente projetadas e funções exclusivas, a Stripe desenvolve uma atividade para que possamos obter o melhor produto possível ao criar, por exemplo:

- Serviços de assinaturas;
- Lojas de e-commerce;
- Plataformas de crowdfunding.

Em termos de segurança e como no caso da BestRide estamos a lidar com transações bancárias, a Stripe dispõe de ferramentas de A.I que analisam comportamentos suspeitos, na questão de crimes e fraudes, de forma a proteger os clientes e consumidores.

Utilizamos a ferramenta Stripe no nosso projeto na seção de pagamento da aplicação. Isto é, quando um cliente pretende reservar um tour e precisa realizar uma transação.



Figura 27 - Stripe

5.2.5 ArcGIS WebService

ArcGIS é um webservice público que disponibiliza serviços avançados de geolocalização e também agiliza processos que trabalhem com geolocalização. Este Webservice foi utilizado na parte da aplicação do *driver* para criar um ponto de interesse no roteiro escolhido.

5.2.6 Django

Django é uma framework para desenvolvimento rápido para web escrito em Python, que é uma linguagem de programação de alto nível.

Esta framwork foi utilizada no back-end da nossa aplicação para realização de registos, criação das APIs e recepção de dados do front-end da aplicação além da inserção dessas informações na base de dados da aplicação.



Figura 28 - Logo django / python

5.2.7 MySQL

O MySQL é um sistema open-source de gerenciamento de base de dados relacional SQL, desenvolvido e suportado pela Oracle.

Utilizamos o MySQL no nosso projeto para realizar o gerenciamento da nossa base de dados.



Figura 29 – MySQL

5.3 Arquitetura e Implementação

O nosso projeto foi inicialmente desenvolvido baseado em Cloud Computing do serviço da Amazon AWS. Isto foi positivo para a colaboração independente das várias tecnologias / componentes do projeto. O projeto tem uma arquitetura de microservices, nesta arquitetura, a aplicação é dividida em pequenos serviços que se comunicam através de APIs. Isso permite que os serviços sejam escalados e geridos de forma independente, e facilita a implementação de novas funcionalidades ou correções de bugs.

Abaixo temos a Figura 30, que demonstra a arquitetura da aplicação no serviço AWS.

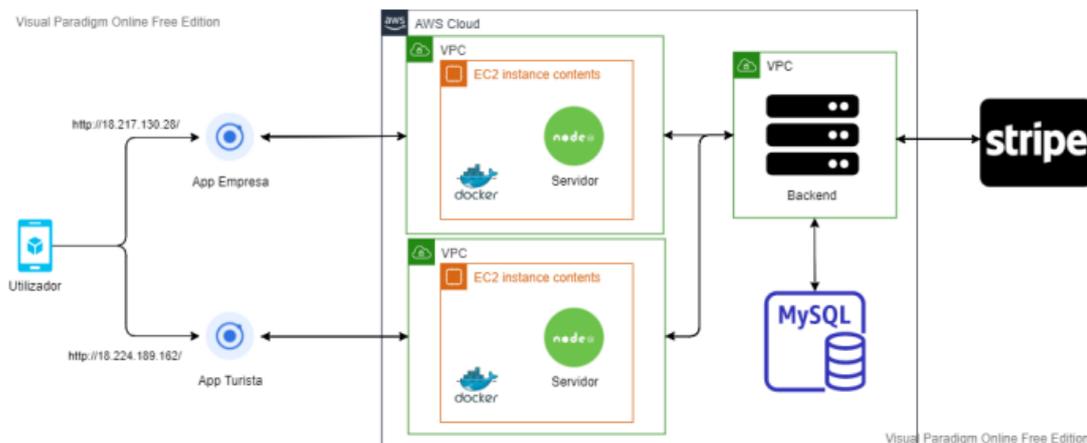


Figura 30 - Arquitetura da aplicação

Como referido anteriormente, a nossa aplicação sofreu alterações na sua estrutura, assim sendo tivemos de migrar para os serviços da Microsoft Azure Cloud. Em relação à conclusão do projeto, conseguimos desenvolver todo o back-end da aplicação utilizando os serviços do Azure como planeado, com exceção do Active Directory. Isto ocorreu por conta de dificuldades de implementação e como nosso tempo era curto, decidimos fazer a autenticação dos utilizadores através da nossa base de dados. Com isso, aqui temos uma comparação dos serviços anteriormente utilizados na AWS e que, após a nossa migração, foram alterados para os serviços do Azure.

Amazon AWS	Microsoft Azure
AWS CodeCommit, Deploy e Pipeline	Azure DevOps
EC2 (Elastic Compute Cloud)	Azure Virtual Machines (VMs)
Amazon S3 (Simple Storage Service)	Azure Blob Storage
Amazon Cognito	Azure Active Directory B2C
Amazon Translate	Microsoft Translator
Amazon RDS	Azure SQL Database

Tabela 5 - Amazon Aws vs Microsoft Azure

Com isso, vamos caracterizar cada serviço tanto da Amazon AWS quanto do Microsoft Azure já conseguindo também comparar as funcionalidades de cada um:

Amazon AWS		Microsoft Azure	
AWS CodeCommit, Deploy e Pipeline	É um serviço de gerenciamento de código-fonte que permite que os usuários armazenem, gerenciem e controlem o acesso a seus repositórios de código-fonte. Ele é integrado com outros serviços da AWS, como o AWS CodeDeploy e o AWS CodePipeline	Azure DevOps	É uma plataforma de gerenciamento de projetos e implantação contínua do Microsoft Azure. Ele permite que os usuários planejem, rastreiem e gerenciem projetos de software, bem como automatizem fluxos de trabalho de implantação, incluindo testes
EC2 (Elastic Compute Cloud)	É o principal serviço de computação da AWS. Ele permite que os usuários criem e gerenciem instâncias de máquinas virtuais na nuvem, escalando-as de acordo com as necessidades de suas aplicações.	Azure Virtual Machines (VMs)	É um serviço de computação em nuvem. Ele permite que os usuários criem e gerem instâncias de máquinas virtuais, escalando-as de acordo com as necessidades de suas aplicações.
Amazon S3 (Simple Storage Service)	É um serviço de armazenamento de objetos na nuvem. Ele permite que os usuários armazenem e recuperem dados de forma escalável e segura.	Azure Blob Storage	É um serviço de armazenamento de objetos. Ele permite que os usuários armazenem e recuperem grandes volumes de dados não estruturados, como arquivos de imagem, vídeo e áudio.
Amazon Cognito	É um serviço de gerenciamento de identidade e acesso da AWS. Ele permite que os usuários gerenciem e protejam as identidades de seus usuários em aplicativos móveis e web, incluindo autenticação e autorização.	Azure Active Directory B2C	É um serviço de gerenciamento de identidade e acesso do Microsoft Azure. Ele permite que os desenvolvedores criem e gerenciem contas de usuário para seus clientes finais (B2C) em aplicativos e sites, sem a necessidade de gerenciar e armazenar senhas.

Amazon Translate	É um serviço de tradução automática da AWS. Ele permite que os usuários traduzam texto de uma linguagem para outra de forma rápida e precisa.	Microsoft Translator	É um serviço de tradução automática do Microsoft Azure. Ele permite que os usuários traduzam texto de uma linguagem para outra de forma rápida e precisa. Ele é capaz de traduzir texto em mais de 60 idiomas diferentes e pode ser integrado em aplicativos, sites e soluções de inteligência artificial.
Amazon RDS	É um serviço de banco de dados relacional na nuvem. Ele permite que os usuários criem, escalem e gerenciem bancos de dados relacionais, incluindo MySQL, por exemplo.	Azure SQL Database	É um serviço de banco de dados relacional gerenciado. Ele permite que os usuários criem, escalem e gerenciem bancos de dados relacionais usando o SQL Server.

No momento atual do projeto, concluímos a arquitetura do Azure. Nessa arquitetura, conseguimos realizar o deploy bem-sucedido do front-end da aplicação, utilizando os dockerFiles e as pipelines. Ao realizar um commit das aplicações no GitHub, as pipelines do Azure são ativadas, permitindo o deployment da solução e gerando a nossa Docker Image. Essa imagem é então inserida em um registro de containers.

Nesse ponto, conduzimos testes de automação em cada parte da aplicação, por meio das pipelines. No entanto, apenas o front-end foi capaz de executar esses testes corretamente. O back-end enfrenta limitações devido à biblioteca gdal, que foi instalada pelos alunos anteriores deste trabalho. No final deste tópico, explicaremos esse problema em detalhes.

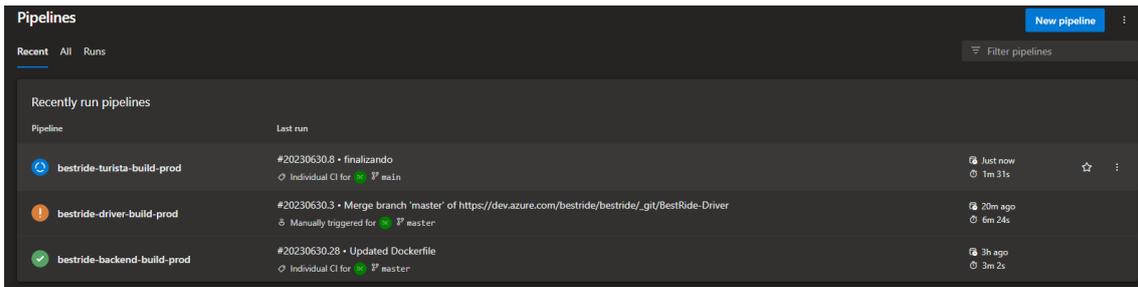


Figura 31 - Pipelines no azure

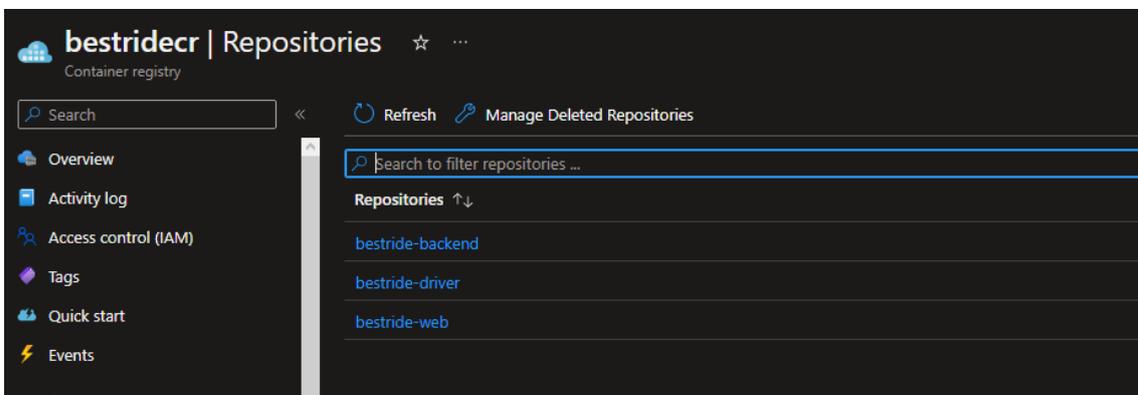


Figura 32 - Container registry Azure

Com as imagens inseridas no Container Registry, o próximo passo é implantá-las em instâncias de contêiner (container instances) para finalizar o processo de deployment da aplicação no Azure.. Para tal, temos 3 containers instances criados, um para cada parte da aplicação (front-end turista, front-end driver e back-end).

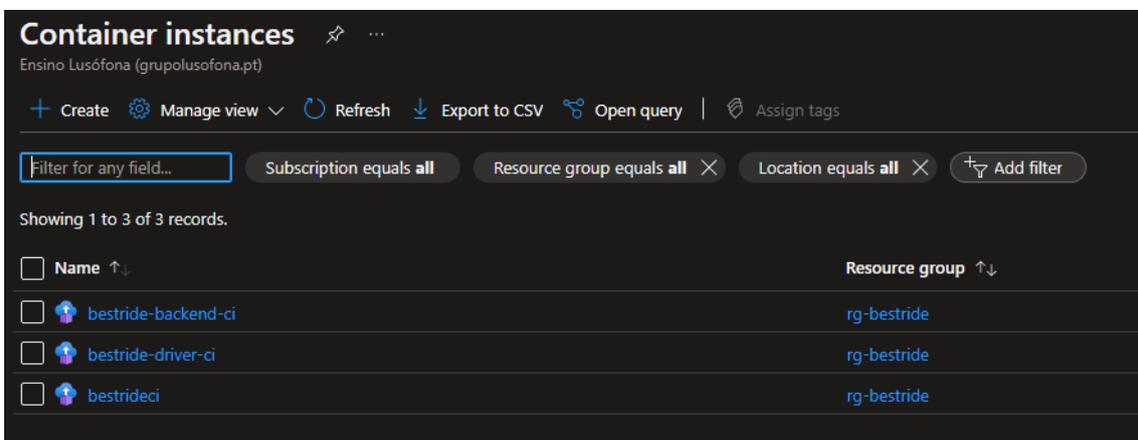


Figura 33 - Container instance Azure

Com isso, o deployment da aplicação foi concluído. Contudo, como referido, tivemos um problema com o deployment do back-end. Devido à falta de documentação entregue pelos antigos alunos deste trabalho do back-end, não sabíamos em qual lugar a biblioteca GDAL estava inserida. A biblioteca GDAL (Geospatial Data Abstraction Library) é uma biblioteca de código aberto que serve para ler, escrever e manipular dados geoespaciais. Ela fornece uma interface consistente para acessar uma variedade de formatos de dados geoespaciais, como imagens de satélite, dados de sensoriamento remoto e informações geográficas vetoriais.

Equanto desenvolvíamos o back-end em localhost, conseguíamos utilizar esta biblioteca ao instalar a mesma em nosso computador. Contudo, ao inserir o docker image do back-end no container instance do azure, já não temos acesso aos nossos ficheiros locais do computador e por isso, deixamos de ter acesso à biblioteca instalada fora do ambiente de desenvolvimento. Tentamos contactar os alunos do ano anterior, para entender qual foi o método que usaram para utilizar a biblioteca gdal dentro das instâncias criadas na AWS anteriormente. Porém, não tivemos sucesso, o que consequentemente, nos impossibilitou de concluir o a inserção da docker image do back-end no na container instance criada. Aqui está a demonstração da docker image sendo enviada para a instância, contudo, nunca é concluída.



The screenshot shows a table with the following data:

Name	Image	State	Previous state	Start time	Restart count
bestride-backend-ci	bestridecr.azurecr.io/bestride-backend:latest	Waiting	Terminated		107

Figura 34 - Inserção docker image back-end na container instance

Como tivemos este problema, por mais que o deployment do front-end das duas aplicações no Azure tenha sido bem-sucedido, infelizmente tivemos que correr a aplicação apenas no localhost visto que o processo de deployment do back-end não foi 100% concluído, faltando apenas o a inserção da docker image corretamente.

Fora este problema, como já referido anteriormente, a implementação do back-end foi concluída e conseguimos utilizar todos os serviços do azure que desejávamos. Com isso, aqui estão os serviços criados dentro da plataforma para sustentar o nosso projeto:

Name ↑↓	Type ↑↓	Resource group ↑↓
bestride-backend-ci	Container instances	rg-bestride
bestride-db	Azure Database for MySQL flexible server	rg-bestride
bestride-db-fs	Azure Database for MySQL flexible server	rg-bestride
bestride-db.private.mysql.database.azure.com	Private DNS zone	rg-bestride
bestride-driver-ci	Container instances	rg-bestride
bestride-translator	Translator	rg-bestride
bestrideci	Container instances	rg-bestride
bestridecr	Container registry	rg-bestride
bestridestorage	Storage account	rg-bestride
bestrideTenant.onmicrosoft.com	B2C Tenant	rg-bestride
NetworkWatcher_eastus	Network Watcher	NetworkWatcherRG
nsg-frontend	Network security group	rg-bestride
nsg-mysql	Network security group	rg-bestride
vm-backend	Virtual machine	rg-bestride
vm-backend933	Network Interface	rg-bestride
vm-backend_OsDisk_1_1daae807fc8c44d0b0fac3e403e168f9	Disk	RG-BESTRIDE
vm-frontend-driver	Virtual machine	RG-BESTRIDE

Figura 35 - Serviços criado no Azure

vm-frontend-driver-ip	Public IP address	rg-bestride
vm-frontend-driver662	Network Interface	rg-bestride
vm-frontend-driver_OsDisk_1_b14defc8c0c149fc9b4ae52cb9772d97	Disk	RG-BESTRIDE
vNetBestRide	Virtual network	rg-bestride

Figura 36 - Serviços criado no azure - 2

Além disso, aqui temos a arquitetura da aplicação implementada. Nesta demonstração, as aplicações estão armazenadas dentro das container instances como explicado anteriormente ainda nesse tópico:

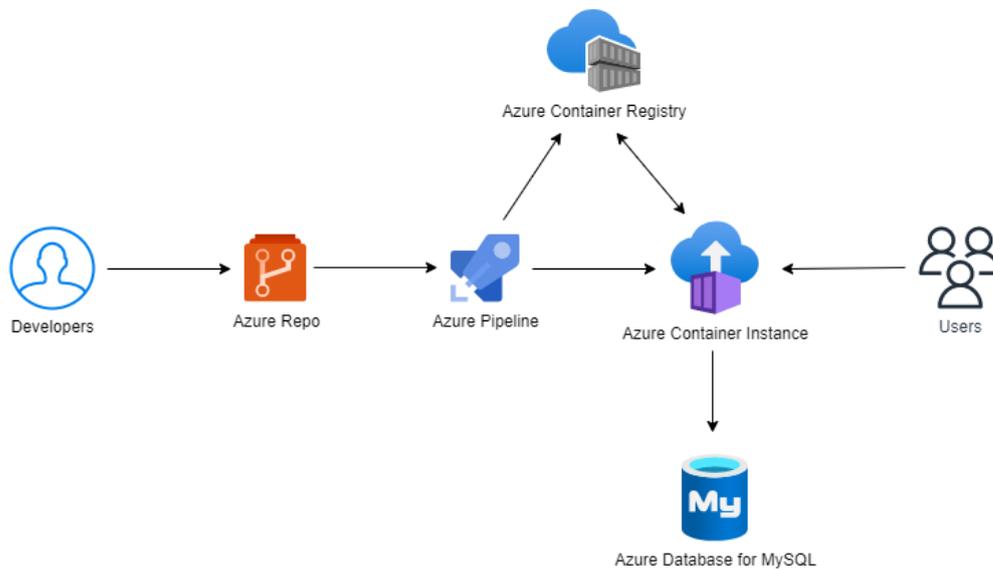


Figura 37 - Nova arquitetura no Azure

5.4 Abrangência

Concluindo, vamos listar as disciplinas e matérias que foram e ainda serão aplicadas no desenvolvimento do projeto:

1. **Engenharia de Requisitos e Testes:** Definição de requisitos e realização de testes para os mesmos
2. **Programação Web:** Desenvolvimento do front-end e back-end (HTML, Django, CSS, JavaScript).
3. **Interação Humano-Máquina / Computação Móvel:** Desenvolvimento de boas práticas de UI/UX.
4. **Computação Distribuída:** Comunicação entre front-end e back-end, além de utilização de Webservices.
5. **Bases de Dados:** Gerenciamento e desenvolvimento de uma base de dados

6. **Algoritmia e Estrutura de Dados:** Maximização da performance da aplicação e boas práticas de estruturação dos dados.
7. **Engenharia de Software:** Criação de diagramas e modelos necessários para o planeamento do projeto.
8. **Sistemas de informação na nuvem:** Desenvolvimento da arquitetura e deploy da aplicação no Azure.
9. **Segurança informática:** Encriptação dos dados na base de dados e utilização de Json Web Tokens.

Com isso, podemos afirmar que as matérias mais utilizadas nesse projeto até o momento foram:

- Interação Humano-Máquina, com o aprendizado de boas práticas de design para aplicação, assim como o desenvolvimento de uma aplicação de fácil entendimento para todos os utilizadores.
- Programação Web, com os ensinamentos de programação de front-end em HTML e back-end com a framework Django.
- Bases de Dados, com a experiência em gestão da base de dados do nosso projeto e do manuseamento da mesma.

6 Plano de testes e validação

A realização de testes e validação é uma etapa fundamental na implementação de uma solução, pois vai além de simplesmente demonstrar o funcionamento da solução. É importante que os testes sejam projetados de forma a comprovar que a solução cumpre os objetivos propostos, especialmente em relação à sua aplicabilidade, pertinência e relevância na solução do problema.

Nesta seção, serão apresentadas a abordagem e a justificação para os testes. Através desses testes, será possível obter evidências confiáveis sobre a eficácia da solução proposta, garantindo assim uma validação e contribuindo para a credibilidade do trabalho desenvolvido.

Como o projeto BestRide passou por uma migração devido ao cancelamento dos serviços de Cloud da AWS e o objetivo passou a ser principalmente o desenvolvimento do back-end e realocação para o Azure, é essencial realizar testes abrangentes para garantir o correto funcionamento da nova estrutura do projeto. Os objetivos dos nossos testes estão mais focados em garantir que a migração tenha sido bem sucedida, conseguindo transferir todos os dados anteriores da aplicação. Estes testes incluem a verificação das informações contidas na base de dados, o teste do serviço de tradução, a validação dos serviços de autenticação e outras áreas críticas do projeto. Com isso, após conversas com nosso orientador, chegamos a conclusão que não seria possível realizar testes com terceiros, pois os testes envolvem verificar se todas as funções implementadas tem o funcionamento desejado, e, para isso, teríamos que usar a ferramenta Postman, visto que o front-end não está completamente implementado.

Por conta disso, relativamente à avaliação anterior, tivemos que fazer pequenas alterações aos testes, testando então as mesmas funcionalidades porém apenas no back-end e sem a participação de terceiros.

É importante destacar que os testes são uma parte crucial do processo de validação do projeto BestRide após a migração, visando identificar e corrigir quaisquer problemas antes do lançamento. Com isso, buscamos assegurar a qualidade e confiabilidade do projeto, garantindo que ele atenda aos requisitos anteriormente propostos.

7 Método e Planejamento

Com base no calendário desenvolvido e entregue nas avaliações anteriores, iremos abordar como decoreu o nosso projeto durante o ano. Além de realizar uma avaliação geral do planejamento do projeto, verificando se conseguimos ou não cumprir com os objetivos dentro dos prazos estabelecidos.

7.1 Método de trabalho

No início do projeto, estabelecemos uma rotina de reuniões semanais com nosso orientador, o professor Rui Ribeiro, para acompanhar o andamento do projeto. Nessas reuniões, discutíamos o progresso feito durante a semana e planeávamos as atividades para a semana seguinte. No entanto, devido à mudança de horários entre os semestres, precisámos cancelar essas reuniões. Em vez disso, começamos a realizar breves discussões logo após as aulas do professor, dedicando alguns minutos para conversar sobre as mudanças que gostaríamos de implementar e decidir o caminho a seguir, especialmente após a mudança de direção e objetivo do trabalho.

Esses diálogos semanais, embora breves, foram fundamentais para nos orientar e tomar decisões no projeto como um todo.

7.2 Cumprimento dos calendários propostos

Com relação ao primeiro calendário desenvolvido no primeiro semestre, podemos afirmar que ainda não tínhamos uma base sólida sobre o tempo de desenvolvimento das novidades da aplicação, com isso, o planejamento proposto ficou incompleto e sem muitas conclusões.

Na segunda avaliação o nosso projeto havia sofrido alterações. Com isso perdemos os serviços da Amazon AWS e o novo objetivo era conseguir migrar o projeto para o Microsoft Azure. Com isso, tivemos que alterar o planejamento do nosso projeto, porém fomos muito otimistas com o tempo que demos para realizar a migração, pois acreditávamos que seria possível realizá-la no espaço de um pouco menos de um mês, o que na verdade nos tomou todo o restante do trabalho.

Face à última avaliação entregue, o nosso planejamento desenvolvido para a parte final da aplicação estava mais de acordo com a realidade e conseguimos seguir grande parte dele. Com um atraso no começo da criação da estrutura no Azure, infelizmente tivemos que diminuir o tempo de realização dos testes da aplicação. Contudo, conseguimos cumprir com todos os objetivos propostos e ainda melhorar o front-end da aplicação.

7.3 Avaliação geral do planeamento

Na avaliação atual podemos afirmar que conseguimos realizar todos os objetivos propostos dentro do planeamento elaborado. Contudo, durante o trabalho tivemos um imprevisto que levou mais tempo do que gostaríamos para ser consertado. Com isso, conseguimos perceber a grande importância de gerir os riscos de um projeto por mais estranhos que eles pareçam. Caso isso tivesse sido feito desde o começo do projeto, poderíamos ter um plano de recuperação dos serviços perdidos ou um plano de migração mais rápido, sem ter que recomeçar todo o desenvolvimento do back-end da aplicação. Porém, mesmo com todos os imprevistos e atrasos que ocorreram conseguimos cumprir o prazo e entrar o que nos foi proposto.

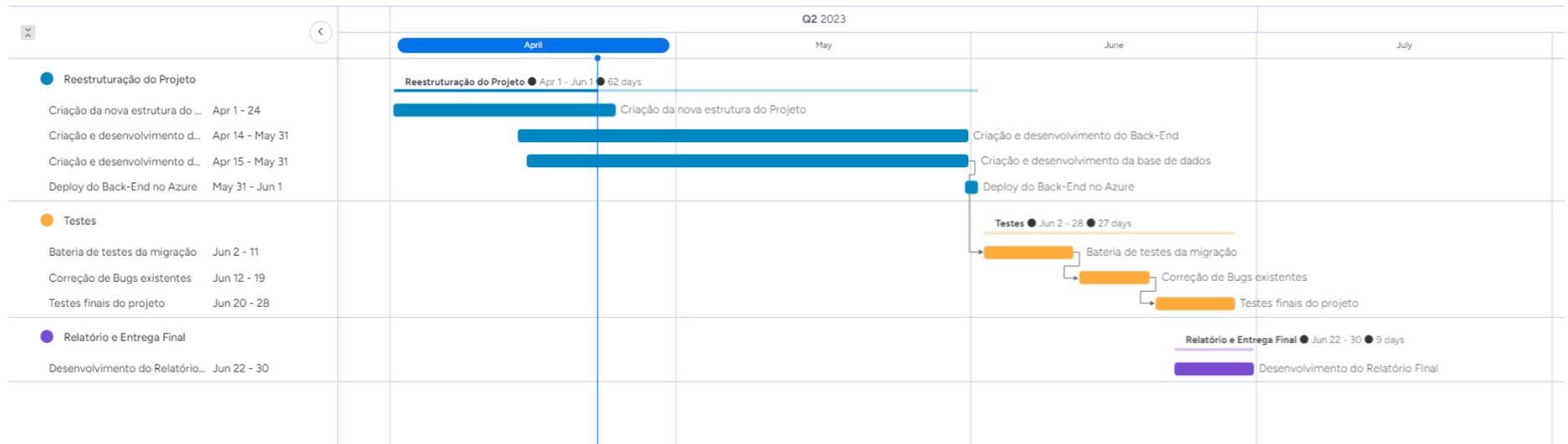


Figura 38 - Calendário de Planejamento

8 Resultados

Como já mencionado no relatório, o objetivo principal do nosso projeto foi alterado. Durante o desenvolvimento do projeto a cliente decidiu não fazer mais parte dele e cancelou os serviços da AWS que estávamos a utilizar para sustentar a aplicação sem aviso prévio. Com isso, tivemos que iniciar um plano de migração da aplicação.

Os serviços da AWS eram utilizados diretamente pelo back-end da aplicação (Django) e, com isso, o nosso foco passou a ser principalmente o desenvolvimento do back-end e criação da arquitetura da aplicação no Azure. No entanto, ainda queríamos fazer algumas alterações no front-end (Ionic), queríamos poder desenvolver novas funcionalidades propostas mas pela falta de tempo e com todo o atraso, decidimos apenas alterar o design da aplicação além de corrigir erros. Após a perda dos serviços, não conseguíamos executar o front-end e, conseqüentemente, não conseguíamos realizar as alterações gráficas corretamente na aplicação. Diante disso, após várias reuniões com o nosso orientador, Professor Rui Ribeiro, começamos a criar a estrutura da aplicação no Azure, a fim de iniciar o desenvolvimento e migrar os serviços da AWS para os serviços do Azure no Django. Para isso, criámos os serviços do Azure que iríamos utilizar na plataforma e desenvolvemos as funções do back-end com eles.

Após a criação da estrutura completa da aplicação no Azure, iniciamos o desenvolvimento do Back-End no localhost. Inicialmente, deixamos de utilizar o Amazon Translate (AWS) e passamos a utilizar o Microsoft Translator (Azure) [Figura 21], proporcionando à aplicação um serviço de tradução inteligente e automático. Em seguida, criamos uma base de dados local (MySQL) e iniciamos a implementação da autenticação de utilizadores. Para isso, tínhamos que alterar a utilização do serviço Amazon Cognito e começar a utilizar o serviço Azure Active Directory B2C (AD B2C). Enfrentamos muitas dificuldades ao utilizar este serviço para realizar a autenticação da aplicação, principalmente devido à nossa falta de experiência e à dificuldade de encontrar documentação suficiente para nos auxiliar na utilização e implementação deste serviço. Como o tempo estava a se esgotar e não estávamos a obter os resultados desejados, decidimos alterar a forma de autenticação da aplicação. Assim, a autenticação da nossa aplicação passou a ser feita por meio da nossa base de dados (MySQL) e utilizando métodos de autenticação próprios do Django, verificando diretamente na base de dados todas as informações necessárias para o funcionamento correto e seguro da autenticação da aplicação. Após a implementação dessas funcionalidades, realizamos os testes apartir da ferramenta Postman, que nos auxiliou bastante na verificação do funcionamento de cada função implementada. Aqui está uma demonstração de uma função de registo de utilizador implementada:

```
diogocerqueira-22002160
def post(self, request):
    if request.method == 'POST':
        try:
            user_data = JSONParser().parse(request)
            user = User(email=user_data['email'], name=user_data['name'],
                       address=user_data['address'], image=user_data['image'], dob=user_data['dob'],
                       phone_number=user_data['phone_number'], gender=user_data['gender'],
                       postal_code=user_data['postal_code'])
            user.verification_code = generate_confirmation_code()
            user.set_password(user_data['password'])
            user.save()
            send_confirmation_email(user_data['email'], user.verification_code)
            return JsonResponse(user_data, status=status.HTTP_201_CREATED, safe=False)
        except KeyError:
            return JsonResponse("Invalid data", status=status.HTTP_400_BAD_REQUEST, safe=False)
        except IntegrityError:
            return JsonResponse("Email already exists", status=status.HTTP_400_BAD_REQUEST, safe=False)
        except Exception as e:
            return JsonResponse(str(e), status=status.HTTP_400_BAD_REQUEST, safe=False)
```

Figura 39 - Função de registo de utilizador

Nesta função, o utilizador envia suas informações e as mesmas são guardadas na base de dados. Contudo a password passa por um processo de encriptação e é enviada para a base de dados cifrada com a cifra pbkdf2_sha256.

Esta é apenas uma função de autenticação que desenvolvemos na aplicação, ainda temos o resto das funções do utilizador, as funções autenticação do Driver e da Company que também foram implementadas

Com a autenticação concluída, passamos para o desenvolvimento das funcionalidades da aplicação propriamente dita. Nesse ponto, deixamos de utilizar o serviço Amazon S3 (Simple Storage Service) da AWS e passamos a utilizar o serviço Azure Blob Storage para permitir o upload de imagens na nossa aplicação. Esta funcionalidade de upload das imagens para a Cloud não foi implementada corretamente pelos alunos anteriores. Ou seja, as funções estavam desenvolvidas mas não funcionavam. Com a reestruturação do back-end, conseguimos realizar o desenvolvimento das funções com a utilização do Azure Blob Storage. Ao darmos upload, as imagens enviadas são armazenadas em um container no Azure e guardamos na nossa base de dados um link para esta imagem que é utilizado posteriormente no front-end [Figura 21]. Além disso, durante as implementações das alterações, reparamos que os tours utilizados na aplicação tinham apenas um ponto como rota fazendo com que a lógica fosse incoerente, pois um tour deveria ter um caminho com mais de um ponto. Posto isso, alterámos a lógica utilizada anteriormente e agora, ao criar um tour, podemos escolher mais de um ponto, criando assim um percurso de turismo.

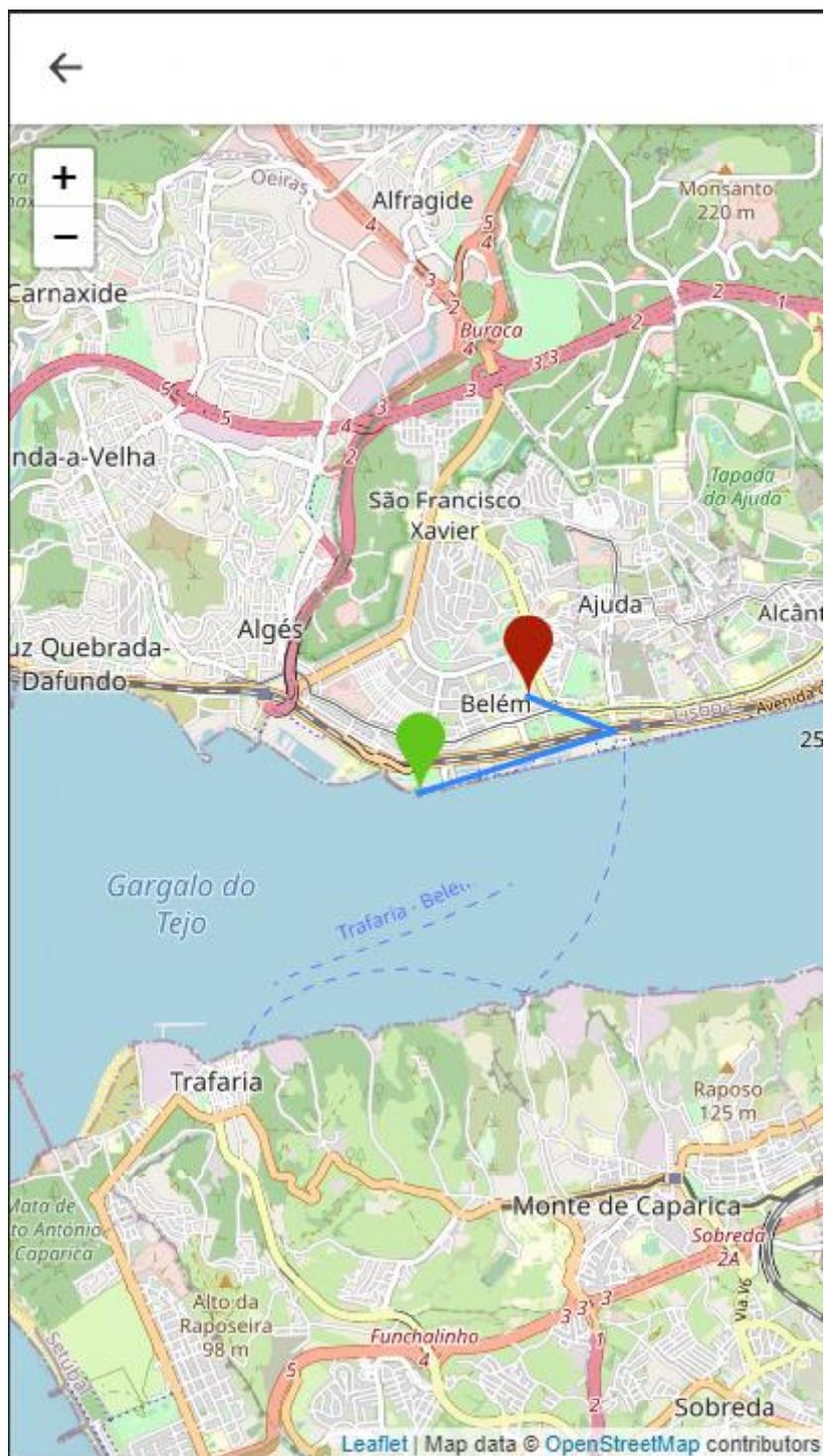


Figura 40 - Nova implementação do mapa do tour

Com isso, retomamos o desenvolvimento do front-end. Nessa etapa, corrigimos bugs e erros de código e fizemos algumas alterações no design da aplicação para que estivesse de acordo com o planeamento inicial (mockups). Aqui estão as comparações dos ecrãs que conseguimos alterar:

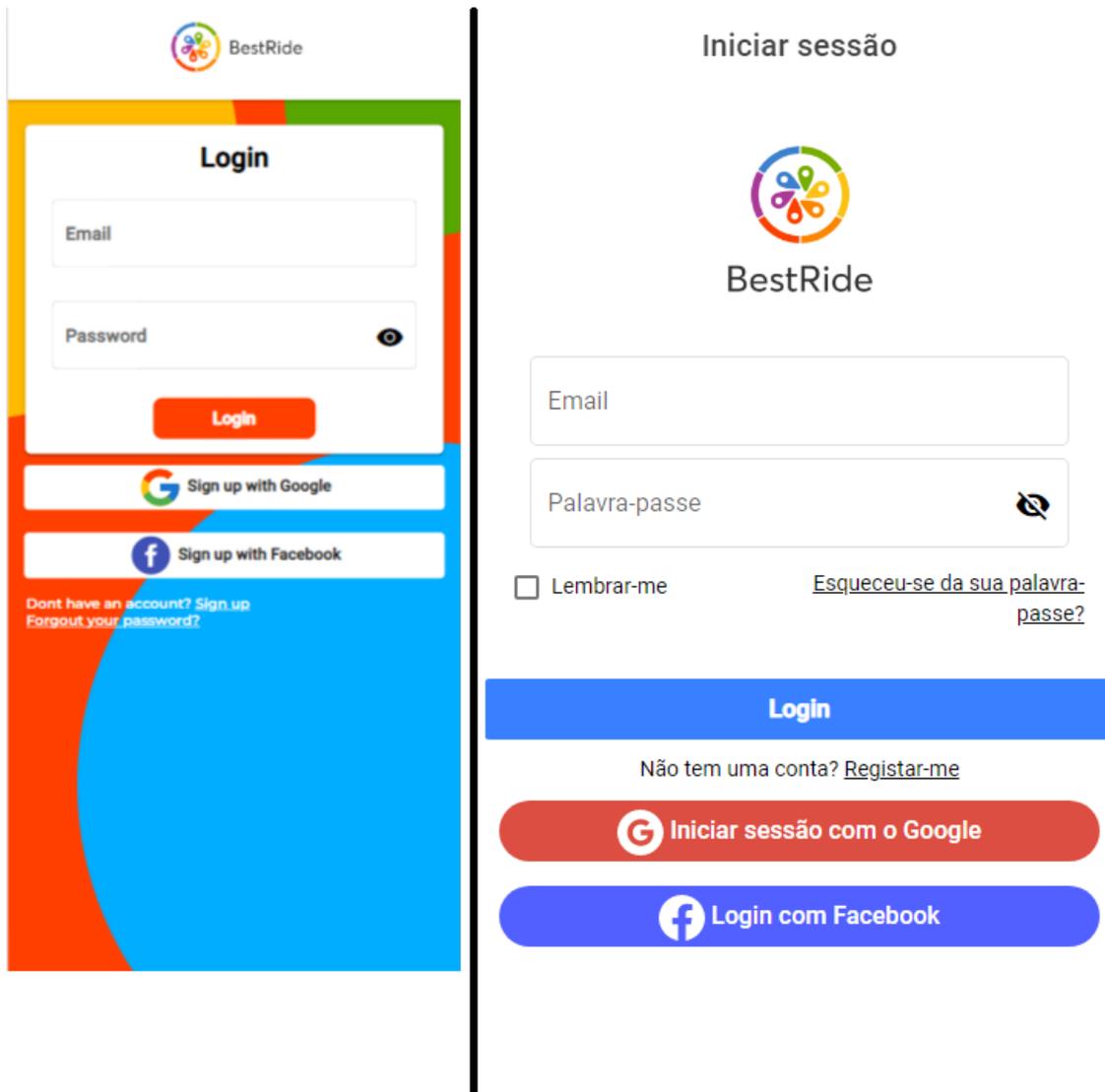


Figura 41 - Comparação login anterior com novo login

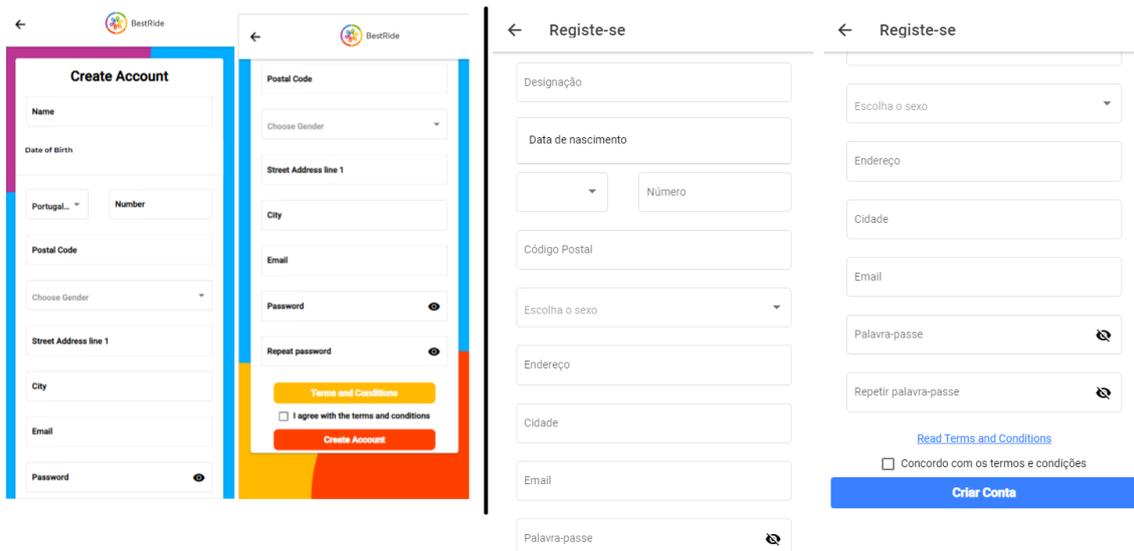


Figura 42 - Comparação ecrã de registo anterior com o novo ecrã de registo

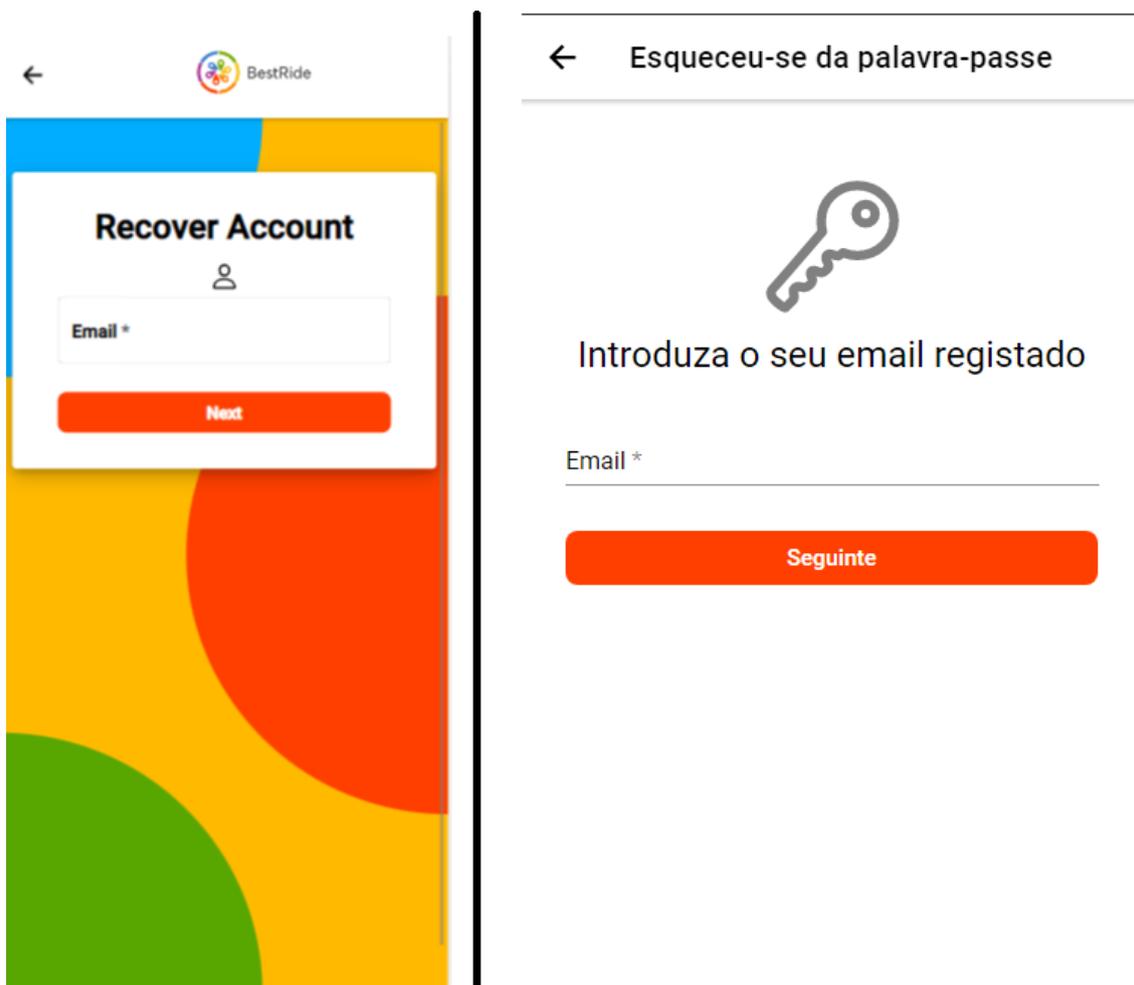


Figura 43 - Comparação ecrã de recuperação de conta

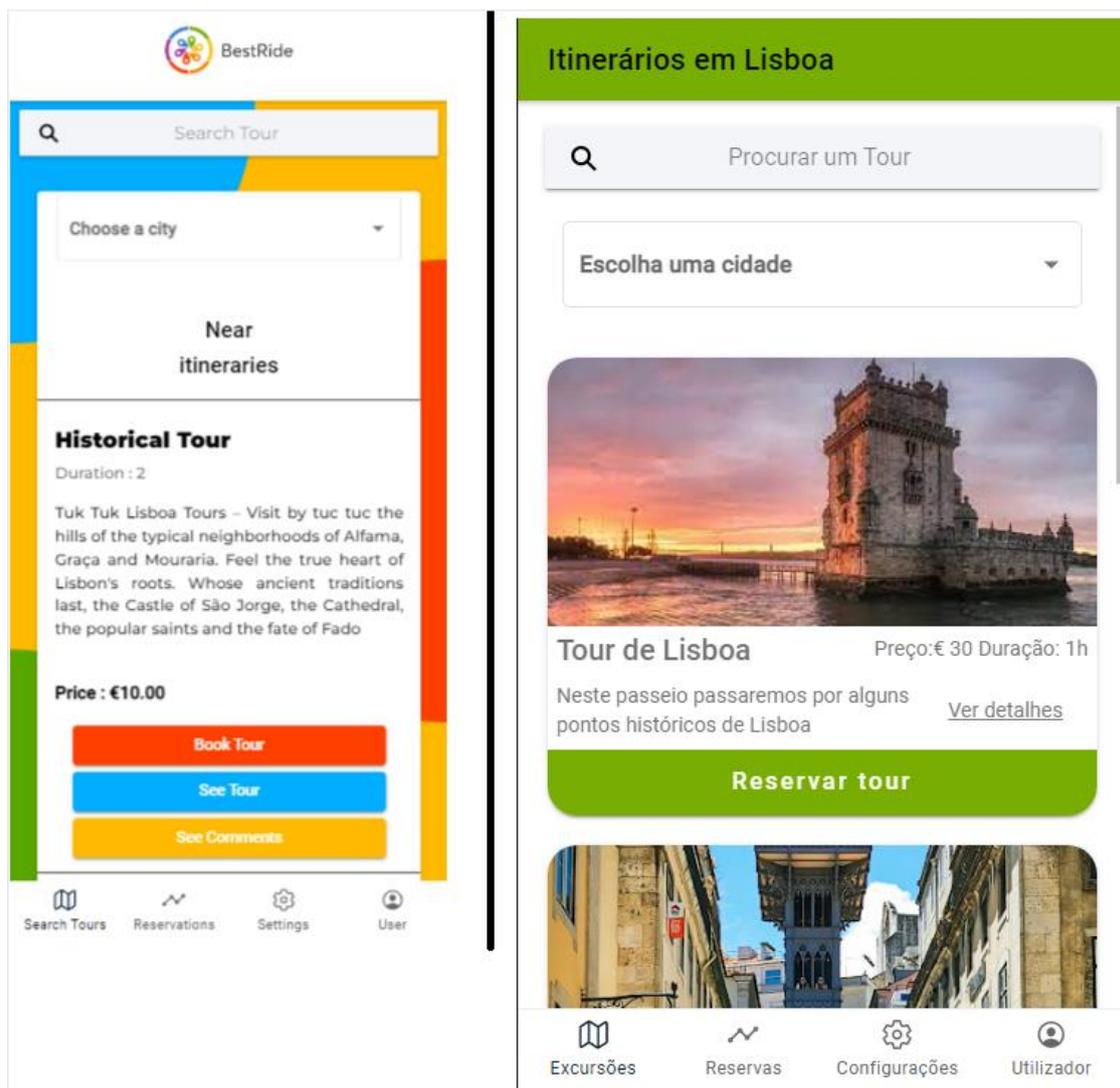


Figura 44 - Comparação ecrã de tours



Figura 45 - Ecrã de detalhes

Vale ressaltar que este último ecrã apresentado (Detalhes de um tour) não existia na aplicação antiga, portanto criámos essa nova funcionalidade na aplicação seguindo o padrão de design que estávamos a implementar nos outros ecrãs. Estas foram as principais alterações realizadas no front-end. Além disso, alterámos alguma lógica do front-end para se adequar às novas implementações do back-end, como por exemplo o facto dos antigos

alunos não guardarem todas as informações que são pedidas para o utilizador na base de dados, e também alguns erros de design.

Realizamos testes de funcionamento da aplicação a partir do postman principalmente, para testar todas as funções implementadas no front-end. Todas correram como gostaríamos com exceção da função que busca os tours mais próximos do utilizador, pois essa não conseguimos implementar de forma correta:

```
## POR FAZER
1 usage 2 diogocerqueira-22002160 +1
@api_view(['POST'])
def getRoadbyDistance(request):
    distanciaMax = float(request.data['distanciaMax']) # Distância máxima em metros
    Road = RoadMap.objects.all()

    distance_dict = {}
    point = Point(request.data['lng'], request.data['lat'],
                  srid=4326) # Inverter as coordenadas para (longitude, latitude)
    print(point)
    for rd in Road:
        linestring = rd.route

        points = list(linestring.coords)
        for p in points:
            distance = point.distance(p)
            print(distance)
            distance_dict[str(rd.title)] = distance

    filtered_road = [rd for rd in Road if distance_dict.get(str(rd.title), float('inf')) <= distanciaMax]

    Road_Serializer = RoadMapSerializer(filtered_road, many=True)
    return Response(Road_Serializer.data)
```

Figura 46 - Função que retorna tours de acordo com a distância

Em relação às mudanças feitas no front-end, testamos a aplicação para verificar se todas as funcionalidades e serviços implementados estavam corretos, e assim finalizamos nosso projeto.

Em resumo, o objetivo do nosso Trabalho Final de Curso foi alterado na metade do desenvolvimento do mesmo, tornando-se principalmente a migração da aplicação da Amazon AWS para a Microsoft Azure Cloud. No entanto, mesmo com esse novo objetivo até a entrega final do projeto, ainda fizemos alterações no front-end da aplicação, resolvendo bugs e erros de código e modificando o design da aplicação de acordo com o planeamento inicial, com o desenvolvimento das mockups e do protótipo. Infelizmente, devido à falta de tempo e à mudança de objetivo e direção do projeto repentina, não conseguimos implementar as novas funcionalidades que gostaríamos. No entanto, alcançamos nosso objetivo principal, que era a migração da aplicação, e conseguimos utilizar com sucesso os serviços de outra cloud, criando toda a nova estrutura apresentada anteriormente, criando uma nova base de dados e desenvolvendo todas as funções do back-end para que funcionassem corretamente. Além disso conseguimos adquirir

competências fundamentais sobre Cloud Computing e o aprendizado de duas frameworks que forma essenciais para o desenvolvimento da nossa aplicação, Angular e Ionic. Infelizmente, não conseguimos finalizar o deployment do back-end no azure por conta da falta de documentação entregue pelos alunos anteriores sobre a instalação do gdal no ambiente de Cloud. Contudo, conseguimos seguir todos os passos necessários para realizar esse deployment e até o conseguimos fazer para as duas aplicações do front-end.

9 Manual Técnico da aplicação

Download das Ferramentas Necessárias para Correr front-end Localmente

- Para correr este trabalho iremos necessitar de 5 softwares diferentes, 4 para correr o projeto e o outro para visualizar o código do mesmo.
- Node.js para interagir com o ecossistema do Ionic.
- Um editor de código (se pretender ver o código).
- Nós utilizamos o Visual Studio Code.
- Linha de Comandos, usar o terminal embutido no Visual Studio Code, CMD ou Powershell CLI.
- Git, para clonar os repositórios.
- Browser, pode utilizar qualquer browser que tenha instalado no seu sistema operativo.

Correr o Projeto

- Ir para a pasta onde pretende fazer o download do código, e abrir um terminal.
- Clonar o repositório da aplicação pretendida.
 - <https://github.com/diogocerqueira-22002160/BestRideTurista.git>
 - <https://github.com/diogocerqueira-22002160/BestRideDriver.git>
- Por fim basta correr o programa em si. Se tudo correu como esperado, o browser irá ter um link com a aplicação a correr localmente por exemplo no localhost:8000.

Download das Ferramentas Necessárias para Correr back-end Localmente

Ir para a pasta onde pretende fazer o download do código, e abrir um terminal.

Clonar o repositório

- git clone <https://github.com/diogocerqueira-22002160/BestRideBackend.git>
- Instalar o Python 3.11

Instalar dependências

- pip install pipenv
- pipenv install

Entrar no ambiente virtual

- pipenv shell

Efetuar as migrações

- python manage.py makemigrations
- python manage.py migrate

Correr o programa.

- `python manage.py runserver`

Se tudo correu como esperado, o browser ira ter um link com a aplicação a correr localmente por exemplo no localhost:8000. Recomendamos a utilização da ferramenta Postman para testar as rotas da API. Juntamente com este projeto está um ficheiro JSON com exemplos de requests para cada rota, que pode ser importado para o Postman.

10 Agradecimentos

Por fim, gostaríamos de expressar a nossa sincera gratidão a todos os professores que contribuíram para a realização deste projeto. Os seus esforços e apoio foram fundamentais para o sucesso do mesmo.

Primeiramente, gostaríamos de agradecer ao nosso orientador, Professor Rui Ribeiro, pela sua orientação valiosa e por compartilhar seu conhecimento e experiência ao longo deste trabalho. Sua orientação e feedback foram essenciais para o desenvolvimento do projeto.

Também gostaríamos de estender nossos agradecimentos ao Professor Pedro Perdigão e Professor Rui Santos pelo suporte e colaboração durante o desenvolvimento do trabalho.

Atenciosamente,

Erick Pina e Diogo Cerqueira

Bibliografia

- [1] DEISI, Regulamento de Trabalho Final de Curso, Set. 2021.
- [2] Universidade Lusófona de Humanidades e Tecnologia, www.ulusofona.pt,
acedido em Out. 2021.
- [3] Empresa BestRide

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso

Anexos

Testes

Guião de Tarefas

TESTES DE USABILIDADE

Instruções

O objetivo deste teste é verificar o funcionamento da aplicação BestRide, após a migração do projeto estar concluída.

Será informado(a) caso seja ultrapassada a duração total prevista e daremos o teste por concluído.

Relembramos que não está a ser testado(a), mas sim a ajudar-nos a melhorar o protótipo da aplicação.

Quando der por concluído o teste a equipa da BestRide pedir-lhe-á para preencher um breve questionário.

O teste é anónimo e confidencial.

Confidencial

ID n° _____

O objetivo do presente teste é que utilize a aplicação disponibilizada.

Tente realizar as seguintes tarefas na aplicação:

1. Abrir seção de tours e escolher a localização “Lisboa” e verificar se existem tours em Lisboa.
2. Selecionar um tour e ver os detalhes do mesmo (preço, duração, guia turístico, etc...).
3. Tentar efetuar uma reserva de um tour.
4. Efetuar login na aplicação de seguida troque o idioma da aplicação para um idioma que esteja familiarizado e por fim efetue a avaliação de um tour.
5. Vá as definições da conta e altere um dos dados da sua conta.
6. Tente efetuar a alteração da senha.
7. Apague a sua conta da aplicação.

Caso necessite de mais algum dado contacte o Administrador de teste.

Obrigada pela sua Colaboração!

Confidencial

ID n° _____

Questionário de Satisfação

QUESTIONÁRIO DE SATISFAÇÃO – BestRide

Por favor, leia cada uma das seguintes questões e responda com sinceridade e espontaneidade.
Assinale com uma cruz (X) a opção que considerar mais correcta.



		1	2	3	4	5	
1	Neste protótipo posso facilmente ver as tarefas prioritárias a realizar	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
2	O nome dos menus está adequado com as páginas	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
3	O desenho gráfico dos ecrãs é	Desagradável	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agradável
4	Vocabulário utilizado	Difícil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Simple
5	As cores utilizadas permitem uma fácil leitura	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
6	O contraste das cores da aplicação é agradável	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
7	Facilidade de voltar ao ecrã anterior	Difícil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fácil
8	Facilidade em trocar de páginas na aplicação	Difícil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fácil
9	Os ícones utilizados são de fácil compreensão	Discordo completamente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Concordo completamente
10	Tamanho da letra	Desadequado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Adequado
11	Aspecto geral dos ecrãs	Desagradável	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Agradável
12	Acessibilidade geral da aplicação	Difícil	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fácil
13	As informações que a aplicação disponibiliza são	Inúteis	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Úteis
14	Com base na sua experiência, qual a probabilidade de utilizar a aplicação BestRide no futuro	Difícilmente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Fácilmente



Aspectos mais **negativos**:

Aspectos mais **positivos**:

Obrigado pela colaboração

Compromisso de Confidencialidade

Compromisso de Confidencialidade

A equipa BestRide conta com a sua participação num estudo de usabilidade à aplicação BestRide. Este estudo tem como objetivo melhorar a aplicação e testar o funcionamento da mesma.

Iremos recolher informação sobre a forma como utiliza a aplicação e como esta responde às suas necessidades informativas. Após o teste vamos solicitar-lhe o preenchimento de um questionário. Utilizaremos esta informação, bem como a recolhida junto dos outros participantes, para recomendarmos formas de melhoria do serviço atualmente disponibilizado pela aplicação BestRide.

Ao assinar este acordo, nós, juntamente com a ULHT, garantimos que esta informação será utilizada apenas para efeitos de avaliação do serviço e análise dos resultados e que em caso algum iremos ceder esta informação a terceiros. A sua assinatura do acordo garante o seu consentimento para utilizarmos os comentários verbais ou escritos.

Sempre que desejar interromper o estudo e fazer uma pausa, pode fazê-lo, bastando para isso informar a equipa de teste que o pretende fazer.

Pode igualmente dar por terminado o estudo, a qualquer momento.

Se tiver alguma dúvida pode colocá-la a equipa do teste, agora ou durante o mesmo.

Se está de acordo com os termos do presente acordo, por favor assine em baixo, indicando assim a sua aceitação.

	Administradores	Participante
Nome:	Equipa BestRide	_____
Data:	_____	_____
Assinatura:	_____	_____

Este estudo destina-se a testar a aplicação BestRide, não a testá-lo(a) a si.

Obrigado pela sua participação!

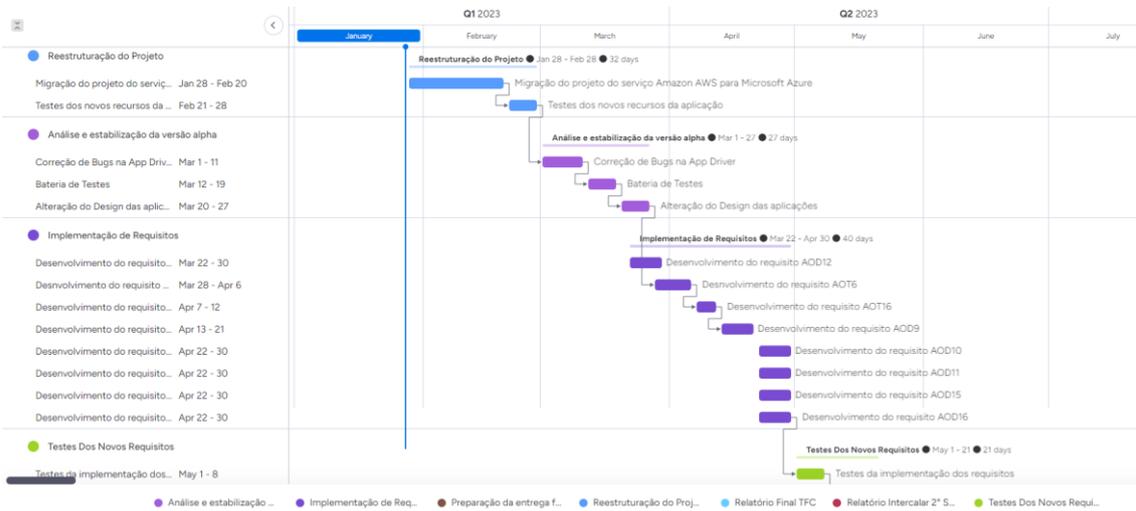


Figura 47 - Calendário de Planeamento Antigo

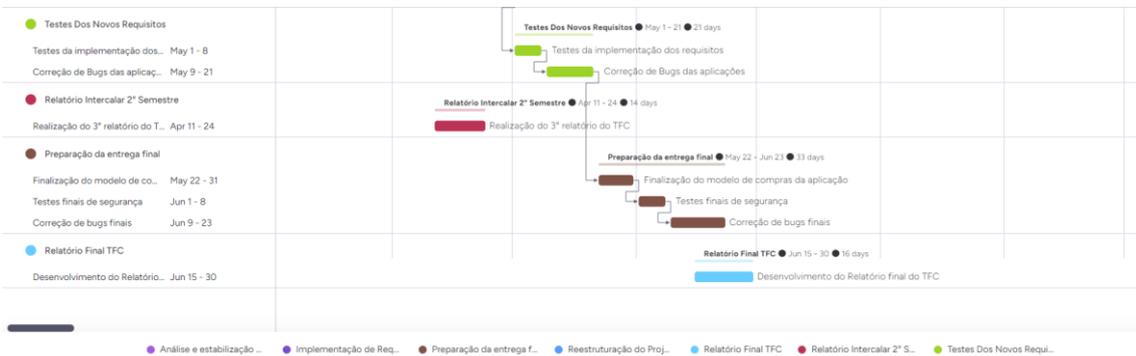


Figura 48 - Calendário de Planeamento Antigo