

Francisco Fernandes (98178), Miguel Reis (108545), Nuno Pinho (108648), Paulo Macedo (102620), Pedro Ferreira (98620)

Versão deste relatório: **2022-05-23**, v1.0

RELATÓRIO – *ELABORATION & CONSTRUCTION*

Construção

Conteúdos

1 Introdução.....	1
1.1 Sumário executivo.....	1
1.2 Controlo de versões.....	2
1.3 Referências e recursos suplementares.....	2
2 Arquitetura do sistema.....	2
2.1 Objetivos gerais.....	2
2.2 Requisitos com impacto na arquitetura.....	3
2.3 Decisões e justificação.....	3
2.4 Arquitetura do software.....	4
2.5 Arquitetura física de instalação.....	5
3 Incremento 1.....	5
3.1 Casos de utilização no Incremento 1.....	5
3.2 Histórias de utilização selecionadas.....	6
3.3 Estratégia e estado da implementação.....	7
4 Incremento 2.....	7
4.1 Casos de utilização no incremento 2.....	8
4.2 Histórias de utilização selecionadas.....	8
4.3 Aceitação e garantia de qualidade.....	8
4.4 Estado de implementação.....	8
Apêndice.....	9
5 Especificação dos casos de utilização.....	9
5.1 Pacote: Conta do Utilizador.....	9
5.1.1 Registo do Utilizador.....	9
5.1.2 Alteração de dados da Conta.....	10

1 Introdução

1.1 Sumário executivo

Este relatório apresenta os resultados da construção do incremento 1 e 2, apresentando a arquitetura do sistema e casos de utilização da aplicação.

Os cenários são baseados nos casos de utilização apresentados em apêndice (secção 5)

O primeiro incremento, desenvolvido na Iteração 3, foca a validação da arquitetura proposta. Foram consideradas sobretudo as funcionalidades relacionadas com o utilizador.

No segundo incremento foram focados os casos de utilização principais para ir de encontro ao objetivo inicial do novo sistema de informação.

1.2 Controlo de versões

Quando?	Responsável	Alterações significativas
2023-05-13	Paulo Macedo	Sistema de autenticação e registo da aplicação
2023-05-15	Paulo Macedo	Interface do perfil do utilizador
2023-05-15	Pedro Ferreira	Arquitetura física de instalação
2023-05-16	Pedro Ferreira	Arquitetura do software
2023-05-20	Paulo Macedo	Interface do histórico do utilizador
2023-05-20	Paulo Macedo	Decisões e justificação
2023-05-20	Paulo Macedo	Requisitos com impactos na arquitetura
2023-05-22	Nuno Pinho	Estratégia e estado da implementação
2023-05-23	Nuno Pinho	Especificação dos casos de utilização
2023-05-23	Francisco Fernandes	Objetivos Gerais
2023-05-23	Miguel Reis	Incremento 1
2023-05-28	Paulo Macedo	Implementação do caso de uso “reserva de trotinetes”
2023-05-28	Paulo Macedo	Implementação do caso de uso “aluguer de trotinetes”
2023-05-29	Paulo Macedo	Implementação do caso de uso “suporte ao cliente”
2023-05-30	Francisco Fernandes	Casos de utilização no Incremento 2
2023-06-05	Paulo Macedo	Aceitação e garantia de qualidade
2023-06-05	Nuno Pinho	Estado da Implementação
2023-06-05	Francisco Fernandes	Histórias de utilização selecionadas

1.3 Referências e recursos suplementares

Para estudar a tecnologia que mais se adequa ao sistema de informação, sendo este um sistema que suporta a geolocalização, foi utilizada a seguinte fonte:

- <https://kafka.apache.org/>

Também para verificar todas as funcionalidades que a biblioteca **LeafLet** disponibiliza, de forma a verificar a viabilidade da mesma para substituir o Google Maps, foi utilizada a seguinte fonte:

- <https://leafletjs.com/>

2 Arquitetura do sistema

2.1 Objetivos gerais

Os objetivos gerais para a arquitetura da aplicação de reserva e aluguer de trotinetes são:

1. Acesso fácil e universal: Os utilizadores devem poder aceder à aplicação de reserva e aluguer de trotinetes a partir de qualquer lugar, através de um browser ou de aplicações móveis, sem a necessidade de instalar software específico.
2. Funcionalidades completas: A aplicação deve oferecer um conjunto abrangente de recursos, incluindo reserva de trotinetes e aluguer em grupo ou individual, pagamento, localização e monitorização de trotinetes disponíveis, histórico de aluguer, leitor de QR code.
3. Integração com sistemas externos: A arquitetura deve permitir a integração com sistemas externos, como serviços de pagamento, serviços de mapeamento e localização e serviços de terceiros para verificar a disponibilidade de trotinetes.
4. Desempenho e escalabilidade: A aplicação deve ser projetada para lidar com um grande número de solicitações simultâneas, garantindo um desempenho adequado mesmo em momentos de alta demanda. Além disso, a arquitetura deve ser escalável para suportar o crescimento futuro do número de utilizadores e trotinetes.
5. Confiabilidade e desempenho: A aplicação deve ser confiável e ter um bom desempenho, com tempos de resposta rápidos e mínima ocorrência de erros ou falhas.
6. Segurança: A aplicação deve incorporar medidas de segurança robustas para proteger as informações pessoais dos utilizadores, transações financeiras e garantir a integridade dos dados.
7. Suporte a pagamentos eletrónicos: A aplicação deve integrar-se com plataformas de pagamentos eletrónicos para permitir transações seguras e desmaterializadas. A arquitetura deve ser flexível o suficiente para permitir a substituição do fornecedor de serviços de pagamento sem interrupção das operações.

Estes objetivos gerais orientarão as escolhas de arquitetura para a construção da aplicação de reserva e aluguer de trotinetes.

2.2 Requisitos com impacto na arquitetura

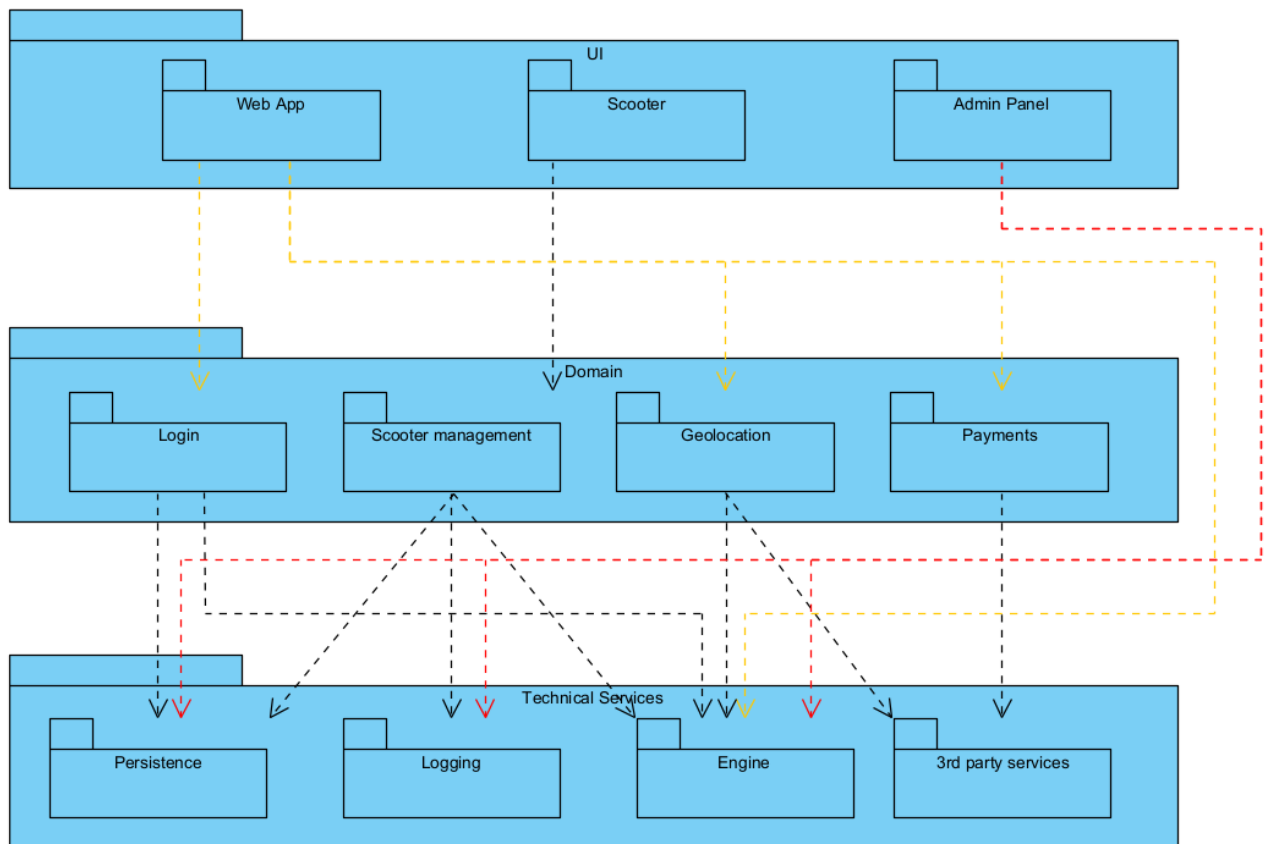
Requisitos	Descrição
Rint.2	O portal da loja deve-se ajustar para ter uma apresentação adequada ao ecrã, designadamente para <i>smartphones</i> e <i>tablets</i> .
RDes.2	A plataforma deverá fornecer os dados mais recentes sobre a sua frota de trotinetes aos seus utilizadores.
RDes.4	Atualização e manutenção contínua da aplicação para garantir o funcionamento da mesma.
RSeg.2	Toda a informação relativa a um utilizador(histórico de reservas e alugueres de trotinetes, também como o registo do trajeto efetuado) deve ser cifrada.
RExt.1	É necessário visualizar trotinetes e trajetos num mapa

2.3 Decisões e justificação

Tendo em conta os objetivos para a arquitetura, e os requisitos levantados na Análise, foram tomadas as seguintes decisões:

- Frontend implementado com o recurso a **HTML** e **CSS**, devido à sua compatibilidade com diversos dispositivos, e à fácil implementação, recorrendo a bibliotecas, tal como a biblioteca **Bootstrap 5**, que disponibiliza vários componentes para serem implementados, com a possibilidade de alterar o seu aspeto com as classes disponibilizadas. Os membros do grupo também têm bastante conhecimento com esta biblioteca e linguagens, facilitando o desenvolvimento da plataforma.
- Para tornar a plataforma interativa e dinâmica, foi utilizado **JS** e **JQuery** (biblioteca em **JS**). O **JavaScript** é bastante versátil facilitando, por exemplo, a interação com botões. Permite criar uma experiência interativa com o utilizador, auxiliando as tarefas de alugar trotinetes e visualização da disponibilidade de trotinetes.
- Transmitir informação dos equipamentos para a **API** é necessário uma plataforma rápida, com a capacidade de processar muita informação, sempre com a aptidão de escalabilidade. Portanto, para esse efeito, recorreu-se ao **Kafka**, capaz de enviar e receber conjuntos de dados para um posterior processamento. Para o serviço em questão, isto reflete-se em recolher dados sobre a disponibilidade das trotinetes, reservas e alugueres efetuados. Kafka permite lidar com vários eventos de forma eficiente, assegurando a transmissão de dados fulcrais para o bom funcionamento do serviço.
- A análise de dados e a sua respetiva monitorização é feita com o recurso à ferramenta, **Prometheus**. Ao integrar o **Prometheus** no serviço de aluguer e reserva de trotinetes, é possível recolher dados relacionados com a performance do sistema Estratégia e estado da implementação (latência, ocorrência de erros, utilização de recursos), que ajuda a identificar possíveis otimizações de performance, e a detetar anomalias. O **Prometheus**, tem a capacidade de gerar alertas, quando certas condições são verificadas. Por exemplo, se o número de alugueres efetuados sem sucesso exceder um certo limite, esta ferramenta pode ser configurada para alertar este problema em específico, levando à identificação e resolução de problemas antes que possam impactar a todo o serviço.
- Para construir uma **REST API**, utilizou-se **Flask**, devido ao facto desta *framework* ser simples e minimalista, fazendo com que seja fácil de usar e compreender. Como os membros do grupo estão familiarizados com **Python**, usar **Flask** para o desenvolvimento de uma **REST API** pode resultar numa melhor eficiência e produtividade no que toca à fase de desenvolvimento.
- A biblioteca de mapas digitais será a **LeafLet**. Este serviço possui um plano gratuito e ao contrário de alternativas, como o Google Maps, não mostra watermarks ao longo de todo o mapa, melhorando a experiência. Esta biblioteca fornece também uma grande quantidade de funcionalidades e opções de customização, sendo ideal para mostrar a localização de trotinetes e destinos, facilitando a navegabilidade e visualização do percurso do utilizador.

2.4 Arquitetura do software



A articulação entre os módulos decorre da seguinte forma:

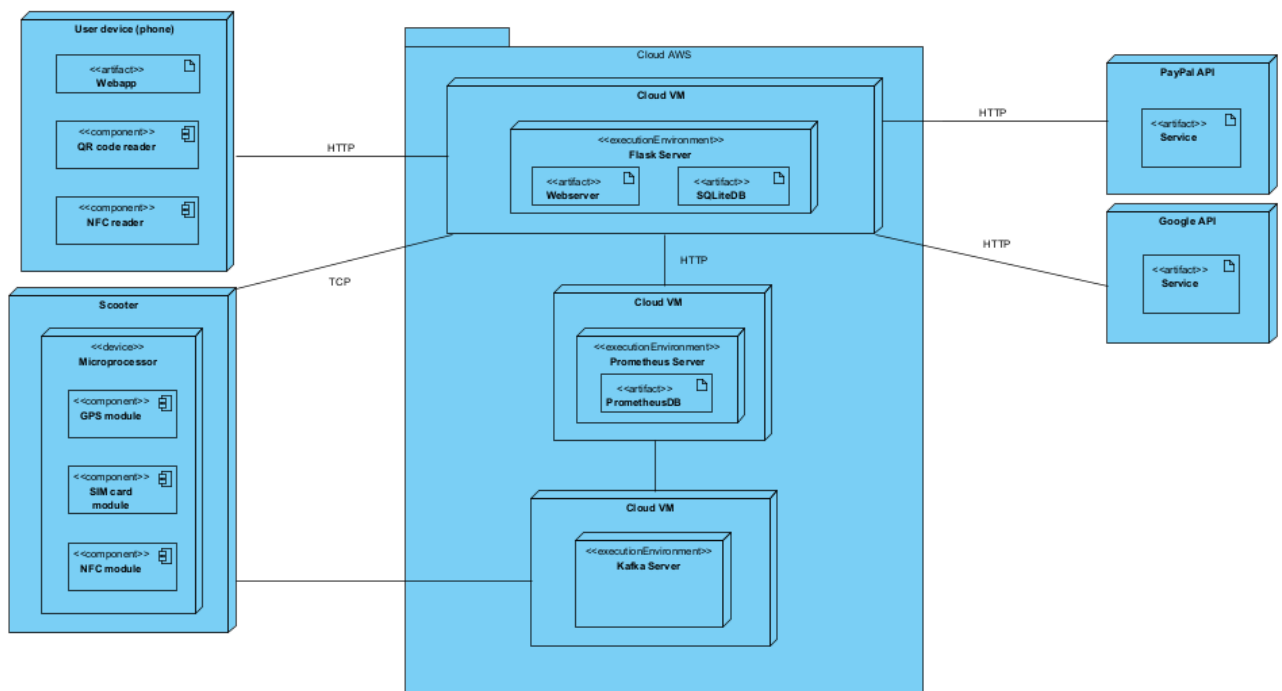
A webapp tem as componentes de login, pagamentos e geolocalização.

A gestão de trotinetes passa por enviar dados para logging, guardando-os numa base de dados.

Para bloquear/desbloquear uma trotinete, é necessário receber um sinal do servidor (engine).

O admin tem acesso a todas as informações guardadas sobre as trotinetes e a webapp, pelo que tem acesso às bases de dados. Também pode bloquear ou desbloquear as trotinetes, por isso também tem acesso aos respetivos endpoints.

2.5 Arquitetura física de instalação



As trotinetes vão estar periodicamente a enviar informações (geolocalização, bateria, estado, etc) para o **Kafka**, que por sua vez as redireciona para o Prometheus (base de dados time-series).

O servidor Flask está a fazer *host* dos *sites* e trata da lógica da aplicação *web*, guardando informações da mesma e dos utilizadores na base de dados (**SQLite**). Quando necessita de informação relativa às trotinetes, envia um pedido **HTTP** para o Prometheus.

Também existe uma ligação websocket entre o servidor **Flask** e as trotinetes para, após ser confirmado o aluguer da trotinete(s), o servidor enviar um sinal para a(s) desbloquear. O mesmo acontecerá no fim da viagem.

Para pagamentos e representação da geolocalização de cada trotinete num mapa, o servidor **Flask** comunica com serviços de terceiros, como a **API** do PayPal e da Google, respetivamente.

3 Incremento 1

3.1 Casos de utilização no Incremento 1

Neste primeiro incremento implementado, o foco esteve na validação da arquitetura proposta, através da implementação da arquitetura do negócio e registo do utilizador. Para isso, seleccionámos os casos de uso "Registo do Utilizador" e "Atualização do Perfil". Sendo uma das partes essenciais da aplicação e necessária para o teste e desenvolvimento da mesma, daí a sua implementação em primeiro lugar.

A especificação detalhada dos casos de utilização encontra-se em anexo (secção 4). A partir dessa análise, definiram-se as histórias de utilização a implementar.

3.2 Histórias de utilização selecionadas

As histórias (*user stories*) incluídas nesta interação fazem parte do *backlog* do projeto, acessíveis em https://github.com/PauloMaced0/AS_Project

Histórias incluídas nesta interação:

- A Diana estuda na Universidade de Coimbra e quer arranjar um método de transporte para as suas viagens entre casa e universidade, por isso decidiu registar-se na aplicação “Trotinet”.
- Após o login na aplicação, a Diana decidiu que devia mudar a sua palavra-passe para uma mais segura e de forma a preencher os seus dados pessoais, acedeu à sua página de perfil.

História/ <i>use case slice</i>	CrITÉrios de aceitação
A Diana decidiu registar-se na aplicação Trotinet. Após saber da aplicação e da possibilidade de alugar trotinetes para as suas viagens diárias decidiu experimentar.	Cenário 1: Registo com sucesso Sendo a primeira vez que usa a aplicação. Estando na página inicial da aplicação. Escolhe a opção de registo e preenche com os seus dados pessoais, escolhendo uma palavra-passe que corresponde aos requisitos. A sua conta é criada com sucesso e pode fazer o <i>login</i> na aplicação. Surge uma notificação a dizer que a conta foi criada com sucesso e redireciona-a para a página de Login. Cenário 2: Registo sem sucesso Estando na página inicial da aplicação. Escolhe a opção de registo e preenche com os seus dados pessoais, escolhendo uma palavra-passe que corresponde aos requisitos. A sua conta não é criada devido a um erro no qual o nome de utilizador ou e-mail já se encontram registados na aplicação.
A Diana decidiu que devia mudar a sua palavra-passe para uma mais segura. Após decidir que a aplicação “Trotinet” é bastante útil e prática para a sua vida, decide atualizar os seus dados pessoais e mudar a sua palavra-passe para uma mais segura.	Cenário 1: Atualização com sucesso Ao aceder à sua página de perfil, preenche os dados pessoais que restam do primeiro registo na aplicação, primeiro e último nome, a data de nascimento e número de telemóvel. Clica no botão para guardar as alterações e consegue atualizar os seus dados com sucesso. Ainda na mesma página, confirma a sua senha atual e cria uma nova palavra-passe, digita novamente a nova palavra-passe de forma a confirmá-la. Clica no botão para guardar as alterações e consegue atualizar a palavra-passe com sucesso. Surge uma notificação a dizer que a mesma foi atualizada. Cenário 2: Atualização sem sucesso A atualização dos dados pessoais não foi possível devido à falta de dados introduzidos ou ao mau preenchimento.

3.3 Estratégia e estado da implementação

Funcionalidades Implementadas:

- Interface para a visualização de alugueres e reservas efetuadas.
- Sistema de log in/sign up com base de dados funcional.
- Perfil do utilizador totalmente funcional.
- Interface para visualizar trotinetes (puramente estática).
- Navegação entre páginas.

Ferramentas Utilizadas:

Frontend:

- **HTML, CSS, JavaScript** - formatação e responsividade do website.

Backend:

- **Kafka** - transmissão de dados.
- **Prometheus** - base de dados (time-series).
- **SQLite** - base de dados.
- **Flask** - rest api.
- **LeafLet** - biblioteca de mapas.

Foi utilizado Github para a gestão de tarefas e backlog do projeto. como também para a gestão do código e controlo de versões.

4 Incremento 2

4.1 Casos de utilização no incremento 2

No incremento 2, o foco passou pelo desenvolvimento e finalização das interfaces pretendidas bem como pela integração dos serviços de infraestrutura, de forma a que o processo de construção do projeto ficasse concluído. De encontro a esse objetivo seleccionámos os casos de uso: "Alugar Trotinetes", "Reservar Trotinetes", "Verificar o histórico de operações efetuadas", e também, "Resolver problemas relativos ao serviço/equipamento", pois são parte do propósito da criação deste projeto. Seria então fulcral que estes estivessem funcionais no segundo e último incremento do processo de construção da aplicação. Desta forma, é possível aos utilizadores alugar, reservar trotinetes, e caso necessite de ajuda, recorrer ao *chat*, entrando diretamente em contacto com um administrador.

A especificação detalhada dos casos de utilização encontra-se em anexo (secção 4).

4.2 Histórias de utilização selecionadas

As histórias (*user stories*) incluídas nesta interação fazem parte do *backlog* do projeto, acessíveis em https://github.com/PauloMaced0/AS_Project

Histórias incluídas nesta interação:

- O Dinis é um estudante universitário que pretende conciliar o estudo com a prática de desporto. Para isso ele necessita de um meio rápido e de baixo custo que lhe permita chegar ao complexo desportivo a tempo de realizar o treino. Perante esta necessidade, o Dinis decidiu alugar uma trotinete na aplicação “Trotinet”.
- A Maria é uma turista que está a visitar Aveiro com amigos. De forma a tirar o maior proveito do pouco tempo que dispõem para visitar a cidade, a Maria e os amigos necessitam de um meio que lhes permita deslocar-se de forma rápida entre os pontos turísticos da cidade. Para isso, o grupo de amigos decidiu utilizar o serviço personalizado de reserva de trotinetes da “Trotinet”.
- Tanto a Maria como o Dinis encontraram algumas dúvidas relativas ao funcionamento do serviço e dos equipamentos, e portanto, decidem recorrer à ajuda fornecida por um administrador através do *chat* dedicado ao suporte de clientes, dessa forma puderam esclarecer todas as suas dúvidas.

História/ <i>use case slice</i>	CrITÉRIOS de aceitação
<p>O Dinis decidiu alugar uma trotinete na “Trotinet”.</p> <p>Após tomar conhecimento da “Trotinet” e da possibilidade de alugar trotinetes para as suas viagens diárias, o Dinis decidiu experimentar.</p>	<p>Cenário 1: Aluguer com sucesso</p> <p>O Dinis entra na aplicação, cria a sua conta e faz login. Ele acede à página de aluguer e, estando junto da trotinete que pretende alugar, faz a leitura do QR code que lhe corresponde. Feita a ligação trotinete – aplicação, o Dinis clica em “checkout” para proceder à finalização do processo. Confirma os dados do aluguer, escolhe o método de pagamento e efetua o mesmo. O aluguer é concluído com sucesso e o Dinis pode agora utilizar a trotinete.</p> <p>Cenário 2: Aluguer sem sucesso</p> <p>Dinis faz o registo e login na aplicação trotinete. Entra na página de aluguer e faz scan do QR code da trotinete que está perto de si. A associação trotinete – aplicação não é feita devido a um erro na leitura do código QR ou então a trotinete encontra-se reservada, não podendo efetuar o seu aluguer.</p>
<p>A Maria decidiu reservar trotinetes na aplicação “Trotinet” durante a sua viagem.</p> <p>Após tomar conhecimento do serviço de reserva da “Trotinet” a Maria decidiu usar a aplicação para tornar a sua visita mais dinâmica.</p>	<p>Cenário 1: Reserva com sucesso</p> <p>A Maria cria a sua conta e faz login na “Trotinet”. Na página inicial, clica no ícone de reserva dirigindo-se a esta página. Dado que pretende ter trotinetes para todos na hora que planeia terminar a visita de cada ponto turístico, a Maria escolhe um</p>

	<p>serviço mais personalizado clicando nessa opção. Preenche os dados da reserva relativos ao local, hora, duração e quantidade de trotinetes pretendida de forma correta e procede ao Checkout clicando em “Book”. A Maria efetua o pagamento e a reserva é efetuada com sucesso.</p> <p>Cenário 2: Reserva sem sucesso</p> <p>A Maria acede à aplicação, cria a sua conta e procede ao login. Dirige-se à página de reserva e clica em reserva personalizada calculando ser a que mais se adequa à sua visita. Ao entrar na página procede ao preenchimento dos dados não especificando qualquer local de recolha. Clica em “Book” para submeter a reserva e não consegue, pois é necessário indicar um local de recolha.</p>
<p>A Maria e o Dinis decidem recorrer ao chat para resolver algumas questões.</p> <p>No momento de usufruir do serviço, para terem a certeza que estavam a proceder da melhor forma, decidiram pedir ajuda.</p>	<p>Cenário 1: Resolução das questões/problemas</p> <p>Os utilizadores iniciam a interação com o administrador, apresentando todas as suas questões. Do outro lado essas mensagens são recebidas e analisadas, sendo posteriormente enviada a resposta ao cliente.</p>

4.3 Aceitação e garantia de qualidade

Os testes foram realizados com o selenium IDE, onde foram feitos os seguintes testes:

- User book/rent history
- Join group trip
- Live chat
- Login/Logout
- Register
- Rent scooter
- Scooter booking
- Custom scooter booking
- Update user information

Para testar os vários casos de utilização, de forma a garantir os critérios de aceitação, foram feitos testes aos cenários implementados, tal como é possível analisar a partir da seguinte figura.

Project: TrotiNET

Test suites +

Search tests...

Use Cases

- Register and Login
- Rent scooter
- Scooter booking
- Join group trip
- Scooter custom booking
- History
- Update user information
- Live chat
- Client support
- Logout

http://localhost:5001

	Command	Target	Value
2	set window size	459x995	
3	click	id=register_button	
4	click	id=username	
5	type	id=username	André
6	click	id=email	
7	type	id=email	andré@ua.pt
8	click	id=password	
9	type	id=password	12345678
10	click	css=body	
11	click	id=confirm_password	
12	type	id=confirm_password	12345678
13	click	id=submit	
14	click	css=h1	Login
15	click	id=email	
16	type	id=email	andré@ua.pt
17	type	id=password	12345678
18	click	id=submit	
19	click	css=.title	TrotiNET

Command open

Target /

Value

Description

Pode-se testar a aplicação importando “TrotiNet_Webapp/TrotiNET.side” para o Selenium IDE, onde se pode encontrar todos os cenários dos casos de utilização, podendo fazer a sua depuração ao executar o test suite “Use cases”.

4.4 Estado de implementação

Funcionalidades implementadas (final):

- Sistema de *login/sign up*.
- Perfil do utilizador totalmente funcional e customizável.
- Interface para a visualização do histórico de reservas/aluguel (estática).
- Interface de pagamento do serviço.
- Integração dos serviços da infraestrutura.
- Interface para aluguer de trotinetes, inclui simulação de leitura de QR code.
- Interface para a reserva de trotinetes, inclui mapa para escolha de trotinetes.
- Interface para aderir a uma viagem de grupo.
- Interface para visualização da localização e percurso da trotinete.
- Implementação do chat para resolução de problemas (interface para cliente e administrador).

Funcionalidades por implementar:

- Interface para a visualização do histórico de reservas/aluguel funcional com *backend*.

- Pagamento de serviço responsivo a todas as operações.
- Diferentes métodos de pagamento disponíveis.

No geral, foi implementada a maior parte das funcionalidades esperadas e planeadas, no entanto o funcionamento dos casos de utilização centrais não sofrem qualquer impacto, não comprometendo o fluxo e utilização da aplicação.

Apêndice

5 Especificação dos casos de utilização

5.1 Pacote: Conta do Utilizador

5.1.1 Registo do Utilizador

Caso de utilização:	#1: Registo do Utilizador
Versão:	Iteração 1, v2023-05-20
Breve descrição	O Utilizador pretende usufruir do serviço, mas como não possui conta, necessita de primeiro criar uma.
Pré-condições:	O utilizador encontra-se no website do serviço.
Pós-condições	O utilizador pode então utilizar o serviço com a sua conta. Possui também uma página para ver e alterar os detalhes da sua conta.
Fluxo base:	1. Aceder à página principal do website O utilizador acede a esta página para usar o serviço. 2. Aceder á página para criar conta Na caixa de login o utilizador clica em “criar conta” e é dirigido à página. 3. Preencher dados O utilizador preenche os seus dados escolhendo uma palavra-passe que corresponde aos requisitos. 4. Conta criada A sua conta é criada com sucesso e pode fazer o login na aplicação.
Fluxos alternativos:	Passo 2: Já possui conta Se o utilizador já possuir uma conta não necessita de criar outra, podendo assim fazer login direto no serviço.
Exceções:	Ex1: Sistema ou página do perfil indisponível

Requisitos especiais:	[Usabilidade] O preenchimento de dados tem que ter em conta os requisitos pedidos.
-----------------------	--

5.1.2 Alteração de dados da Conta

Caso de utilização:	#2: Alteração de dados da Conta
Versão:	Iteração 1, v2023-05-20
Breve descrição	O Utilizador pretende alterar a sua palavra-passe para uma mais segura, como também pretende preencher os restantes dados para obter uma conta mais completa.
Pré-condições:	O utilizador encontra-se no website do serviço. O utilizador já possui uma conta.
Pós-condições	O utilizador consegue alterar e preencher os dados pretendidos obtendo uma conta mais completa.
Fluxo base:	1. Entrar na conta O utilizador entra no website com a sua conta. 2. Aceder à página do perfil O utilizador clica na aba do perfil dirigindo-se a esta página. 3. Preencher dados O utilizador preenche os dados escolhendo uma palavra-passe que corresponde aos requisitos. 4. Guardar alterações O utilizador clica em guardar as alterações feitas. 5. Alterações guardadas As alterações são guardadas com sucesso.
Fluxos alternativos:	Passo 5 : As alterações não são guardadas O utilizador clica em guardar as alterações mas não acontece atualização visto que os requisitos do preenchimento dos dados não foram satisfeitos. Passo 5.1 : Preencher novamente os dados e guardar alterações Preencher novamente os dados tendo em conta os requisitos dispostos e guardar as alterações.
Exceções:	Ex1: Sistema ou página de registo indisponível
Requisitos especiais:	[Usabilidade] O preenchimento de dados tem que ter em conta os requisitos pedidos.

5.2 Aluguer e Reserva de trotinetes

5.2.1 Aluguer de trotinetes

Caso de utilização:	#3: Aluguer de trotinetes
Versão:	Iteração 2, v2023-05-30
Breve descrição	O Utilizador pretende alugar uma trotinete
Pré-condições:	O utilizador encontra-se no website do serviço. O utilizador já possui uma conta. O utilizador possui um dispositivo com câmara e digitalizador de QR code funcionais.
Pós-condições	O utilizador consegue alugar uma trotinete.
Fluxo base:	1. Entrar na conta O utilizador entra no website com a sua conta. 2. Aceder à página de aluguer de trotinete A página inicial o utilizador clica em "Rent Scooter" dirigindo-se a esta página. 3. Escolher trotinete a alugar O utilizador faz scan do QR code da trotinete que pretende alugar. 4. Aceder à página de Checkout O utilizador clica em "Checkout" para aceder à página de pagamento e finalização do aluguer. 5. Realizar o pagamento Após confirmação dos dados de aluguer o utilizador procede ao pagamento clicando no método respetivo.
Fluxos alternativos:	Passo 3 : Pretende alugar mais que uma trotinete O utilizador clica em "add another scooter" Passo 3.1 : Escolher a trotinete a adicionar O utilizador faz scan do QR code da trotinete que pretende adicionar. O cliente pode realizar esta ação as vezes que pretender de acordo com o número de trotinetes que deseja alugar.
Exceções:	Ex1: Sistema ou página de aluguer de trotinetes indisponível. Ex2: O utilizador não possui uma conta no website do serviço. Ex3: O utilizador não possui um dispositivo com câmara e digitalizador QR code funcionais.

Requisitos especiais:	<p>[Usabilidade] O sistema deve informar claramente o utilizador sobre o estado do aluguer em cada passo do fluxo.</p> <p>[Segurança] O sistema deve garantir que o pagamento seja seguro e confiável.</p> <p>[Disponibilidade] O sistema deve garantir que as trotinetes estejam disponíveis para aluguer no momento em que o utilizador faz o scan do QR code.</p> <p>[Acessibilidade] O sistema deve permitir que o utilizador cancele a operação a qualquer momento.</p>
-----------------------	--

5.2.2 Reserva de trotinetes

Caso de utilização:	#4: Reserva de trotinetes
Versão:	Iteração 2, v2023-05-31
Breve descrição	O Utilizador pretende reservar uma trotinete
Pré-condições:	<p>O utilizador encontra-se no website do serviço.</p> <p>O utilizador já possui uma conta.</p> <p>O utilizador encontra-se num dispositivo com GPS ativo.</p>
Pós-condições	O utilizador consegue reservar uma trotinete.
Fluxo base:	<p>1. Entrar na conta O utilizador entra no website com a sua conta.</p> <p>2. Aceder à página de reserva de trotinetes Na página inicial o utilizador clica no ícone de reserva dirigindo-se a esta página.</p> <p>3. Escolher no mapa a(s) trotinete(s) a reservar O utilizador seleciona através do mapa a(s) trotinete(s) a reservar podendo ter como referência a sua localização e bateria.</p> <p>4. Escolher data e hora da reserva O cliente escolhe ainda a data e hora para que pretende fazer a reserva.</p> <p>5. Aceder à página de Checkout Clicar em "Book" para submeter os dados de reserva e aceder à página de Checkout.</p> <p>6. Efetuar o pagamento</p>

	Após confirmação dos dados de reserva, o utilizador procede ao pagamento clicando no método respetivo.
Fluxos alternativos:	<p>3: O utilizador pretende uma reserva mais personalizada O utilizador seleciona a opção reserva personalizada dirigindo-se à respetiva página.</p> <p>4: Preencher dos dados O cliente preenche dados relativos ao local, hora, duração e quantidade de trotinetes pretendidas.</p> <p>5. Aceder à página de Checkout Clicar em “Book” para submeter os dados de reserva e aceder à página de Checkout.</p> <p>6. Efetuar o pagamento Após confirmação dos dados de reserva, o utilizador procede ao pagamento clicando no método respetivo.</p>
Exceções:	<p>Ex1: Sistema de página de reserva de trotinetes indisponível.</p> <p>Ex2: O utilizador não possui uma conta no website do serviço.</p> <p>Ex3: O GPS do dispositivo do utilizador não está ativo e o mapa não pode ser carregado.</p>
Requisitos especiais:	<p>[Usabilidade] O sistema deve informar claramente o utilizador sobre o estado da reserva em cada passo do fluxo.</p> <p>[Segurança] O sistema deve garantir que o pagamento seja seguro e confiável.</p> <p>[Disponibilidade] O sistema deve ser capaz de verificar a disponibilidade das trotinetes em tempo real e atualizar o mapa de forma dinâmica.</p> <p>[Acessibilidade] O sistema deve permitir que o utilizador cancele a operação a qualquer momento.</p>

5.2.3 Suporte com o cliente

Caso de utilização:	#5: Suporte com o cliente
Versão:	Iteração 2, v2023-06-05
Breve descrição	O Utilizador pretende esclarecer dúvidas relacionadas com o funcionamento do serviço.
Pré-condições:	<p>O utilizador encontra-se no website do serviço.</p> <p>O utilizador já possui uma conta.</p>
Pós-condições	O utilizador consegue comunicar através do chat.

Fluxo base:	<p>1. Entrar na conta O utilizador entra no website com a sua conta.</p> <p>2. Aceder à página de suporte ao cliente Na página inicial o utilizador clica no ícone de suporte ao cliente dirigindo-se a esta página.</p> <p>3. Digitar a questão que pretende esclarecer O utilizador digita qualquer dúvida ou problema acerca do funcionamento do sistema que pretenda ver esclarecido.</p> <p>4. Resposta do administrador O administrador recebe e analisa a dúvida questionada pelo cliente e responde de forma a ajudá-lo.</p> <p>5. Dúvida esclarecida A dúvida é esclarecida e o cliente usufrui do sistema da melhor forma.</p>
Fluxos alternativos:	<p>4.1 Administrador solicita esclarecimento acerca da questão A mensagem enviada pelo utilizador não é suficientemente esclarecedora e o administrador solicita mais informações para que possa ajudar da melhor forma.</p> <p>4.2 Cliente explica melhor a questão O utilizador responde ao administrador clarificando melhor o que pretende questionar.</p>
Exceções:	<p>Ex1: Sistema de página de apoio ao cliente indisponível.</p> <p>Ex2: O utilizador não possui uma conta no website do serviço.</p>
Requisitos especiais:	<p>[Disponibilidade] O administrador deve responder de forma rápida a qualquer dúvida colocada pelo cliente.</p> <p>[Clarividência] O administrador deve ser claro e direto na sua resposta.</p>