

TQS: Product specification report

André Oliveira [107637], Duarte Cruz [107359], Tomás Matos [108624], Diogo Almeida [108902]
v2024-06-03

Índice

TQS: Product specification report 1

1 Introduction..... 2

1.1 Overview of the project 2

1.2 Limitations..... 2

2 Product concept and requirements 3

2.1 Vision statement 3

2.2 Personas and scenarios..... 3

2.3 Project epics and priorities 5

3 Domain model 6

4 Architecture notebook..... 7

4.1 Key requirements and constrains 7

4.2 Architecture view 8

4.3 Deployment architecture..... 8

5 API for developers..... 9

1 Introduction

1.1 Overview of the project

Este projeto tem como objetivo principal aplicar os conhecimentos teóricos e práticos adquiridos para garantir a qualidade de um sistema de reserva e gerenciamento de voos para aeroportos. A cadeira de TQS foca em técnicas para identificar e corrigir defeitos em *software*, além de promover a qualidade em todas as fases do desenvolvimento.

O nosso produto, denominado "*GateMate*", é uma solução abrangente projetada para simplificar e aprimorar a experiência de viagem dos utilizadores que utilizam serviços aéreos. A plataforma oferece uma variedade de funcionalidades, incluindo reserva de bilhetes de avião, *check-in online* e acesso a informações de voos em tempo real.

O "*GateMate*" é uma ferramenta indispensável tanto para os passageiros quanto para os aeroportos. Para os viajantes, proporciona conveniência e praticidade ao permitir que realizem as suas reservas e *check-ins* de forma rápida e fácil, além de fornecer informações atualizadas sobre os horários de partida e embarque dos voos. Já para os aeroportos, contribui para a eficiência operacional ao agilizar o processo de *check-in* e reduzir a necessidade de assistência presencial.

Este oferece também uma oportunidade única para aplicar metodologias de teste, como teste de unidade, teste de integração e teste de sistema, para garantir que o "*GateMate*" atenda aos mais altos padrões de qualidade. Além disso, será dada ênfase à automação de testes e à análise de métricas de qualidade para monitorar e melhorar continuamente a eficácia do software.

Dessa forma, o projeto "*GateMate*" não apenas demonstra a aplicação prática dos conceitos aprendidos na cadeira de TQS, mas também oferece uma solução valiosa para melhorar a experiência de viagem dos utilizadores e a eficiência dos serviços aeroportuários.

1.2 Limitations

Durante o planeamento inicial, identificamos uma potencial limitação que merece destaque: a integração com as companhias aéreas.

Essa integração é essencial para oferecer aos utilizadores uma ampla gama de opções de reserva de voos, garantindo a abrangência e a utilidade do nosso serviço. No entanto, reconhecemos que a integração com os sistemas de reservas das companhias aéreas pode apresentar desafios significativos. Questões como a complexidade dos sistemas existentes, requisitos técnicos específicos e até mesmo políticas internas das companhias podem impactar o processo de integração.

Portanto, é importante ressaltar que estamos cientes desses possíveis obstáculos e estamos preparados para enfrentá-los de maneira proativa. O nosso objetivo é garantir uma integração eficiente e robusta, proporcionando uma experiência de reserva de voos tranquila e satisfatória para os utilizadores.

2 Product concept and requirements

2.1 Vision statement

Este sistema é uma plataforma de reserva e gerenciamento de voos projetada para simplificar e aprimorar a experiência de viagem dos utilizadores que utilizam serviços aéreos. Ele será utilizado pelos viajantes para realizar diversas operações relacionadas a voos, como reserva de bilhetes, *check-in online* e acesso a informações de voos em tempo real.

O problema de alto nível que o nosso sistema busca resolver é a complexidade e a falta de conveniência frequentemente associadas às viagens aéreas. Atualmente, os viajantes muitas vezes precisam lidar com múltiplos *sites* e aplicativos de companhias aéreas diferentes para fazer as suas reservas, realizar *check-in* e obter informações sobre os seus voos. Isso pode ser confuso e demorado, resultando numa experiência de viagem frustrante.

"GateMate" foi concebido para abordar esse problema, oferecendo uma solução abrangente e centralizada para todas as necessidades de reserva e gerenciamento de voos dos utilizadores. Com uma interface intuitiva e fácil de usar, os viajantes podem realizar todas as etapas necessárias antes de sua viagem, desde a reserva do bilhete até o *check-in* e o acesso a informações atualizadas sobre seus voos num único lugar.

Inicialmente, planeávamos incluir uma ampla integração com diversas companhias aéreas, oferecendo aos utilizadores uma seleção abrangente de opções de voos. No entanto, devido à complexidade e aos desafios associados a essa abordagem, decidimos focar apenas nos voos provenientes de Lisboa para o lançamento inicial do sistema, com planos de expansão para incluir mais no futuro.

Este sistema diferencia-se de outros produtos similares pela sua abordagem integrada e centrada no utilizador. Ao consolidar todas as funcionalidades relacionadas a voos numa única plataforma, "GateMate" oferece aos utilizadores uma experiência mais conveniente e eficiente, tornando a viagem aérea mais acessível e agradável.

2.2 Personas and scenarios



Nome: Ana

Idade: 32 anos

Profissão: Executiva de vendas

Descrição: Ana é uma executiva de vendas que viaja com frequência a negócios. Ela trabalha para uma empresa de tecnologia e é responsável por visitar clientes em várias cidades do país. Ana tem uma agenda apertada e precisa coordenar as suas viagens de forma eficiente para maximizar o tempo disponível para as suas reuniões e compromissos profissionais.

Cenário 1: Reserva de Viagem

Ana tem uma reunião importante agendada noutra cidade na próxima semana. Ela precisa reservar um voo que se encaixe na sua agenda lotada. Ana acessa o "GateMate" no seu computador e pesquisa por voos disponíveis para o dia e horário desejados. Ela encontra uma opção conveniente e realiza a reserva em poucos cliques, garantindo o seu bilhete para a viagem.

Cenário 2: Check-in Online

No dia de sua viagem, Ana está ocupada finalizando algumas tarefas antes de sair para o aeroporto. Ela recebe um lembrete do "GateMate" no seu telemóvel para fazer o *check-in online*. Ana abre o *website*, completa o processo de *check-in* de forma rápida e simples, e recebe o seu cartão de embarque eletrônico instantaneamente.

Cenário 3: Atualização de Estado de Voo

Durante a sua viagem de volta para casa, Ana enfrenta um atraso no aeroporto devido a condições meteorológicas adversas. Ela fica preocupada em perder a sua conexão para casa. Usando o "GateMate", Ana verifica o estado de seu voo em tempo real, permitindo reagendar a sua conexão. Com essas informações, ela consegue planejar adequadamente a sua viagem de volta para casa.



Nome: Pedro

Idade: 45 anos

Profissão: Empresário

Descrição: Pedro é um empresário que viaja com frequência a trabalho e lazer. Ele costuma utilizar o aeroporto de sua cidade como ponto de partida para as suas viagens. Como um viajante experiente, Pedro valoriza a pontualidade e a praticidade ao realizar as suas viagens.

Cenário 4: Verificação de Horários de Voos no Aeroporto

Pedro está no aeroporto aguardando o embarque para sua próxima viagem de negócios. Enquanto aguarda, ele precisa verificar o horário de seu voo para garantir que não perca a partida. Pedro dirige-se aos quadros de informações no terminal do aeroporto, onde pode visualizar os horários de partida e de embarque dos voos. Ele localiza o seu voo na lista e confirma os horários. Satisfeito com a confirmação, Pedro aproveita o tempo restante para revisar os seus planos de viagem e preparar-se para embarcar.



Nome: Pedro

Idade: 50 anos

Profissão: Operário do Aeroporto

Descrição: Carlos trabalha no aeroporto há 15 anos, desempenhando várias funções operacionais. Atualmente, ele é responsável pelo atendimento ao cliente e pelo gerenciamento de voos no terminal. Carlos conhece bem as rotinas do aeroporto e é hábil em lidar com situações imprevistas, garantindo que as operações diárias corram sem problemas. Ele dedica-se a fornecer um serviço excelente aos passageiros, ajudando-os a ter uma experiência de viagem tranquila e agradável.

Cenário 5: Verificação de Detalhes de Voos

Carlos está no seu posto de trabalho no balcão de informações do aeroporto. Um passageiro aproxima-se, preocupado com o horário do seu voo e pede informações detalhadas. Carlos acessa ao sistema de gestão de voos do aeroporto no seu computador, onde pode visualizar todos os detalhes dos voos em tempo real, incluindo horários de partida, portões de embarque, atrasos, e alterações de última hora. Ele rapidamente localiza as informações do voo do passageiro e fornece todos os detalhes necessários, acalmando o viajante e ajudando-o a planejar o seu próximo passo no aeroporto.

Cenário 6: Check-in Manual de Passageiros

Durante um período de alta movimentação no aeroporto, o sistema de *check-in* automático encontra-se temporariamente fora de serviço. Carlos, como parte da equipa de atendimento ao cliente, é designado para auxiliar os passageiros com o *check-in* manual. Ele organiza uma fila ordenada e começa a processar cada passageiro individualmente. Utilizando o sistema de reservas do aeroporto, Carlos verifica a identidade dos passageiros, confere as reservas e imprime os cartões de embarque. Ele garante que todos os passageiros sejam atendidos de forma rápida e eficiente, minimizando atrasos e frustrações. Ao final do seu turno, Carlos sente-se satisfeito por ter contribuído para o bom funcionamento das operações do dia.

2.3 Project epics and priorities

- ***Epic 1: Prototype***

Objetivo: O objetivo deste *Epic* foi criar o protótipo da *User Interface* (UI), fundamental para suportar as principais funcionalidades do sistema.

Descrição: Para garantir uma experiência de utilizador intuitiva e eficiente, desenvolvemos um protótipo de alta-fidelidade que demonstra o fluxo de navegação, a disposição visual dos elementos e a interação com as principais funcionalidades. O protótipo incluiu telas como a página inicial, pesquisa de voos, reserva de bilhetes, *check-in online* e atualizações de estado de voo. Este protótipo serviu como uma base sólida para o desenvolvimento posterior, permitindo *feedbacks* precoces e ajustes necessários antes da implementação final.

- ***Epic 2: Fetch and Display Flights***

Objetivo: O objetivo deste *Epic* foi integrar a aplicação com uma *API* externa para recolher dados de voos e exibi-los tanto no *site* quanto nos *displays* do aeroporto.

Descrição: Implementamos a funcionalidade de conexão com uma *API* externa confiável que fornece informações atualizadas sobre voos, incluindo horários de partida e chegada, estado de voos, portões de embarque e possíveis atrasos. Os dados obtidos foram apresentados de maneira organizada e fácil de entender tanto na interface do utilizador do *site* quanto nos *displays* digitais do aeroporto. Esta integração garantiu que os passageiros tivessem acesso a informações precisas e em tempo real, melhorando a eficiência e a satisfação do cliente.

- ***Epic 3: Authentication***

Objetivo: O objetivo deste *Epic* foi desenvolver um mecanismo de autenticação robusto para permitir a compra e o *check-in* de bilhetes de avião *online*.

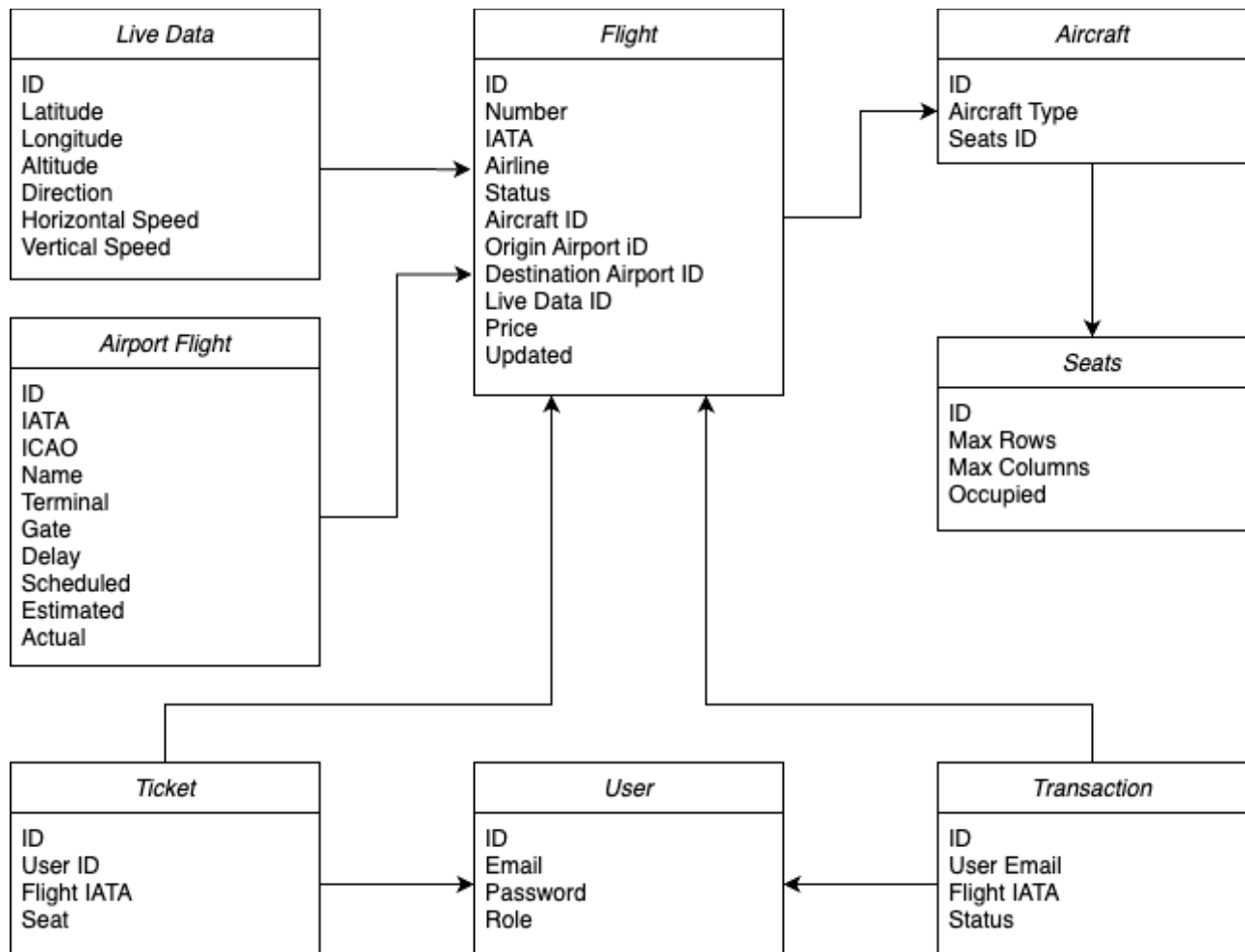
Descrição: Os utilizadores puderam criar contas e fazer *login*. Este sistema de autenticação foi essencial para proporcionar uma experiência personalizada, onde os utilizadores podiam acessar as suas reservas, fazer *check-in online* e adquirir bilhetes de maneira conveniente e segura.

- ***Epic 4: Buy and Check-in a Ticket***

Objetivo: O objetivo deste *Epic* foi implementar a funcionalidade de compra e *check-in* de bilhetes, tanto *online* quanto manualmente.

Descrição: Desenvolvemos uma funcionalidade completa para a aquisição de bilhetes de avião, incluindo seleção de voos e processamento de pagamentos. Além disso, implementamos o processo de *check-in*, permitindo que os utilizadores pudessem realizar o *check-in online* e obter os seus cartões de embarque. Para o *check-in manual*, criamos uma interface que permitisse aos funcionários do aeroporto realizar o *check-in* dos passageiros de forma eficiente. Este *Epic* garantiu uma experiência de utilizador integrada e conveniente, desde a compra do bilhete até o embarque.

3 Domain model



4 Architecture notebook

4.1 Key requirements and constrains

Durante o processo de planeamento e desenvolvimento do nosso sistema de gestão de bilhetes de avião, enfrentamos desafios significativos relacionados à gestão de dados sensíveis dos utilizadores, como informações de pagamento e dados de conta. Além disso, a necessidade de lidar com um grande volume de operações de forma eficiente e sem falhas foi uma prioridade crítica para garantir uma experiência de utilizador contínua e sem interrupções.

Após uma análise cuidadosa das nossas necessidades e requisitos, optamos por adotar uma arquitetura de microserviços para o nosso sistema. Esta decisão foi motivada pela necessidade de independência e escalabilidade dos diferentes componentes do sistema. Ao dividir a aplicação em microserviços individuais, podemos desenvolver, implantar e escalar cada componente de forma independente, sem impactar o funcionamento dos outros serviços. Esta abordagem permite uma maior flexibilidade e agilidade no desenvolvimento, além de facilitar a manutenção e evolução do sistema ao longo do tempo.

Um dos principais benefícios da arquitetura de microserviços é a capacidade de manter os dados sensíveis dos utilizadores, como informações de pagamento e dados de conta, separados e isolados entre os diferentes serviços. Esta separação garante que não haja ligação direta entre as informações, reduzindo significativamente o risco de violações de segurança e protegendo a privacidade dos utilizadores. Cada microserviço tem acesso apenas aos dados necessários para executar as suas funções específicas, garantindo uma maior segurança e conformidade com as regulamentações de proteção de dados.

Para lidar com o grande volume de operações que o nosso sistema processa diariamente, especialmente aquelas relacionadas a transações de pagamento e processos críticos, implementamos uma *message queue*. A *message queue* é uma ferramenta essencial para garantir a confiabilidade e a escalabilidade do sistema, permitindo que as operações sejam processadas de forma assíncrona e distribuída. Isso significa que as mensagens são enfileiradas e processadas de acordo com a capacidade disponível do sistema, evitando sobrecargas e garantindo que nada seja perdido ou ignorado. Além disso, a natureza assíncrona da *message queue* permite que as operações continuem sendo processadas mesmo em caso de falhas temporárias num dos serviços, garantindo uma maior resiliência e disponibilidade do sistema.

A escolha da arquitetura de microserviços e a implementação da *message queue* foram passos fundamentais para garantir o desempenho, a segurança e a escalabilidade do nosso sistema de gestão de bilhetes de avião. Esta abordagem permite-nos atender às necessidades complexas dos nossos utilizadores de forma eficiente e confiável, enquanto nos mantemos ágeis e adaptáveis às mudanças no mercado e nas exigências regulatórias. Estamos confiantes de que esta arquitetura sólida e resiliente nos permitirá oferecer uma experiência de utilizador excecional e garantir o sucesso contínuo do nosso negócio.

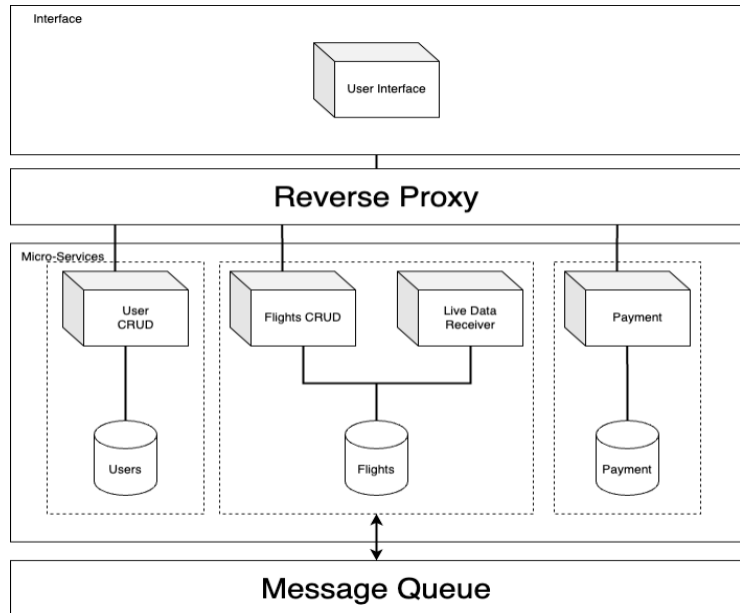
4.2 Architecture view

A arquitetura de sistema proposta é uma estrutura altamente modular e escalável, baseada em microserviços e facilitada por um *reverse proxy*. Essa abordagem permite uma gestão eficiente de várias funcionalidades do sistema, garantindo flexibilidade, desempenho e facilidade de manutenção.

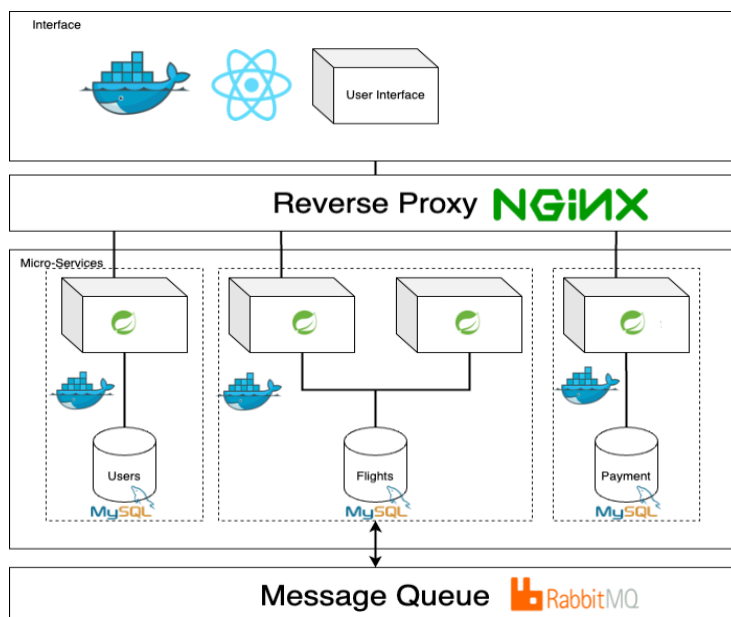
No núcleo dessa arquitetura, encontramos os microserviços, unidades autônomas e especializadas que executam tarefas específicas. O primeiro entre esses é o *User CRUD*, responsável por todas as operações relacionadas aos utilizadores. Desde o registo inicial até à autenticação, esse microserviço interage diretamente com a base de dados de utilizadores, fornecendo uma interface consistente e segura para manipulação de dados de utilizador.

Em paralelo, o *Flights CRUD* assume o controle das operações relacionadas aos voos. Gerenciando uma base de dados dedicada, este microserviço não apenas fornece funcionalidades básicas de CRUD (*Create, Read, Update, Delete*), mas também se integra, via *message queue*, de forma inteligente com o *Live Data Receiver*. Esta integração é vital para manter informações de voo atualizadas em tempo real, permitindo que os utilizadores obtenham dados precisos e atualizados sobre as suas viagens.

Para facilitar a comunicação entre esses componentes e a interface do utilizador, entra em cena o *Reverse Proxy*. Este intermediário inteligente encaminha solicitações dos clientes para os microserviços apropriados, garantindo um roteamento eficiente e um balanceamento de carga adequado. Ao proporcionar uma camada de abstração entre a *User Interface* (UI) e os microserviços, o *reverse proxy* simplifica a arquitetura geral e promove a escalabilidade e a disponibilidade do sistema.



4.3 Deployment architecture



5 API for developers

A *API GateMate* foi organizada numa série de controladores que lidam com diferentes aspetos do sistema de gestão de bilhetes de avião. Cada controlador é responsável por expor *endpoints* que permitem aos desenvolvedores acessar e manipular recursos específicos da aplicação. Existem quatro controladores, sendo estes para autenticação, *checkin*, informações de voos e pagamentos. Abaixo está uma visão geral dos principais controladores e dos serviços/recursos que podem ser obtidos através da nossa *REST-API*:

auth-controller		^
POST	/user Obter detalhes do usuário a partir do token JWT	▼
POST	/userId Obter detalhes do usuário a partir do login	▼
POST	/signup Cadastrar um novo usuário	▼
POST	/login Autenticar um usuário e retornar um token JWT	▼
checkin-controller		^
POST	/checkin/create Efetuar check-in e criar um ticket	▼
GET	/checkin/tickets/{ticketId} Obter informações do ticket pelo ID	▼
flight-controller		^
GET	/scheduledFlights Obter voos agendados	▼
GET	/flights/{flightIata} Obter informações do voo pelo código IATA	▼
GET	/activeFlights Obter voos ativos	▼
transactions-controller		^
PUT	/update_transaction/{id} Atualizar uma transação existente	▼
POST	/create_transaction Criar uma nova transação	▼
GET	/id Obter informações de uma transação pelo ID	▼
GET	/transactions_by_user/{userEmail} Obter transações por e-mail do usuário	▼

Esses controladores oferecem uma interface clara e bem definida para interagir com os diferentes recursos do sistema. Os desenvolvedores podem utilizar esses *endpoints* para integrar funcionalidades relacionadas a bilhetes de avião, transações, autenticação de utilizadores e informações de voos nas suas próprias aplicações.