Uma imagem com texto

Descrição gerada automaticamente

Relatório Final

Serviços e Interoperabilidade de Sistemas

**Aerocontrol**

**Nº do grupo: B**

Elementos do grupo:

Nº: 2211893 Nome: Pedro Henriques

Nº: 2211904 Nome: Pedro Norberto

Nº:2211900 Nome: Rafael Bento

Torres Vedras, *janeiro* de *2023*

**Índice**

Conteúdo

[**1.** **Operações utilizadas na aplicação android** 3](#_Toc124607496)

[**2.** **API REST** 4](#_Toc124607497)

[2.1. API Endpoints 4](#_Toc124607498)

[Login 4](#_Toc124607499)

[Signup 5](#_Toc124607500)

[Repor Password 6](#_Toc124607501)

[Editar dados 7](#_Toc124607502)

[Restaurantes 10](#_Toc124607503)

[Lojas 11](#_Toc124607504)

[Meus Tickets de Suporte 12](#_Toc124607505)

[Criar Ticket de Suporte 13](#_Toc124607506)

[Atualizar Ticket de Suporte 13](#_Toc124607507)

[Criar Mensagem 14](#_Toc124607508)

[Atualizar bilhete 14](#_Toc124607509)

[Meus bilhetes 15](#_Toc124607510)

[Cancelar bilhete 16](#_Toc124607511)

[Comprar bilhete 17](#_Toc124607512)

[**3.** **Messaging** 18](#_Toc124607513)

[3.1. Canais 18](#_Toc124607514)

[3.2. Exemplo 18](#_Toc124607515)

# **Operações utilizadas na aplicação android**

O cliente para poder utilizar os dados na aplicação android, necessita de operações que permitem ao utilizador fazer login e sign up.

Para a utilização da aplicação Aerocontrol foi necessário fazer outras operações para que a aplicação tivesse um bom funcionamento, como por exemplo, pesquisar voos filtrando os mesmos através de aeroportos e datas, comprar bilhetes para os mesmos, criar tickets de suporte tanto para duvidas como para perdidos e achados e também fornecer uma possibilidade ao cliente de editar os seus dados dentro da aplicação.

A aplicação para além da informação relativa ao aeroporto fornece também uma lista com os restaurantes, as lojas e as suas respetivas informações (horário, descrição, menu do restaurante, etc.).

# **API REST**

Para cumprir com os requisitos da aplicação foram criados vários *endpoints*, no próximo ponto é possível visualizar todos os *endpoints* criados. Será ainda apresentado um exemplo de como é possível aceder aos mesmos através do Curl.

## API Endpoints

A seguir são apresentados todos os *endpoints* criados. Em caso de teste deve substituir o que está sublinhado a amarelo, “{pasta}”, pelo caminho desde o servidor até à pasta do projeto.

|  |  |
| --- | --- |
| Login |  |
| **Descrição** | Permite ao cliente fazer login na aplicação android. Requer um header Authorization, sendo esse para a autenticar o user através de “Basic Auth” e devolve um objeto JSON com todos os dados do user para a aplicação. |
| **Método** | POST |
| **Comando Curl** | curl --header "Authorization: Basic cmFmYWVsOjEyMzQ1Njc4" -X POST http://localhost/{pasta}/aerocontrol/backend/web/api/auth/login/ |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Signup |  |
| **Descrição** | Permite ao cliente fazer registo na aplicação android. |
| **Método** | POST |
| **Comando Curl** | curl -X POST -d "username=novoutilizador" -d " password\_hash=12345678" -d "first\_name=Manuel" -d "last\_name=Joaquim" -d "gender=Masculino" -d "birthdate=2000-10-8" -d "country=Portugal" -d "city=Lisboa" -d "email=novouser@gmail.pt" --data-urlencode "phone\_country\_code=+351" -d "phone=919193123" http://localhost/{pasta}/aerocontrol/backend/web/api/auth/signup/ |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Repor Password |  |
| **Descrição** | Permite ao cliente repor a palavra-passe através de um parâmetro de email que envia para que posteriormente receba um email com a informação necessária para repor a password. |
| **Método** | POST |
| **Comando Curl** | curl -X POST -d "email=rafaelcarreirabento7@gmail.com" http://localhost/{pasta}/aerocontrol/backend/web/api/user/reset-password |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Editar dados |  |
| **Descrição** | Permite ao cliente alterar os seus dados pessoais. |
| **Método** | PUT |
| **Comando Curl** | curl -X PUT -d "username=utilizadorupdated" -d "password\_hash=12345678" -d "first\_name=Manuel" -d "last\_name=Joaquim" -d "gender=Masculino" -d "birthdate=2000-10-8" -d "country=Portugal" -d "city=Lisboa" -d "email=novouser@gmail.pt" --data-urlencode "phone\_country\_code=+351" -d "phone=919193123" -d "confirm\_password=12345678" http://localhost/{pasta}/aerocontrol/backend/web/api/user/38?access-token=-PyQHpi-6zWtEvDKVfye4is-iMTvIaKL |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Pesquisar Voos |  |
| **Descrição** | Permite ao cliente pesquisar os voos através de datas e aeroportos. Caso não sejam encontrados voos, permite também através de um “tryAgain” enviar todos os voos do aeroporto de origem e destino sem datas. |
| **Método** | POST |
| **Comando Curl** | curl -X POST -d "origin=ANA Aeroporto de Lisboa" -d "destiny=ANA Aeroporto de Faro" -d "two\_way\_trip=1" -d "origin\_departure\_date=2023-01-28" -d "destiny\_departure\_date=2023-01-30" -d "passengers=1" -d "tryAgain=0" http://localhost/{pasta}/aerocontrol/backend/web/api/flight/search |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Aeroportos |  |
| **Descrição** | Devolve uma lista com todas os aeroportos para futuramente o cliente escolher o destino e a origem do seu voo. |
| **Método** | GET |
| **Comando Curl** | curl -X GET http://localhost/{pasta} /aerocontrol/backend/web/api/airports |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Restaurantes |  |
| **Descrição** | Devolve uma lista com todas os restaurantes e os seus menus ao cliente |
| **Método** | GET |
| **Comando Curl** | curl -X GET http://localhost/{pasta} /aerocontrol/backend/web/api/restaurants |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Lojas |  |
| **Descrição** | Devolve uma lista com todas as lojas. |
| **Método** | GET |
| **Comando Curl** | curl -X GET http://localhost/{pasta} /aerocontrol/backend/web/api/stores |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Meus Tickets de Suporte |  |
| **Descrição** | Devolve uma lista com todos os tickets de suporte do utilizador que efetuar login. O login é efetuado por “QueryParam”, ou seja, enviado no url o “access-token”. |
| **Método** | GET |
| **Comando Curl** | curl -X GET http://localhost/{pasta}/aerocontrol/backend/web/api/support-tickets/my-support-tickets?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo** | Uma imagem com texto  Descrição gerada automaticamente |

|  |  |
| --- | --- |
| Criar Ticket de Suporte |  |
| **Descrição** | Cria um ticket de suporte com uma mensagem. |
| **Método** | POST |
| **Comando Curl** | curl -X POST -d "title=Casaco perdido" -d “message=Perdi um caso dia 3 de janeiro" http://localhost/{pasta}/aerocontrol/backend/web/api/support-tickets?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Atualizar Ticket de Suporte |  |
| **Descrição** | Altera o estado do ticket de suporte para “Concluído” e posteriormente se estiver associado a algum item dos perdidos e achados, alterará também o item para “Entregue” |
| **Método** | PUT |
| **Comando Curl** | curl -X put http://localhost/{pasta}/aerocontrol/backend/web/api/support-tickets/1?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Criar Mensagem |  |
| **Descrição** | Cria uma mensagem no ticket de suporte através dos parâmetros especificados por POST. |
| **Método** | POST |
| **Comando Curl** | curl -X POST -d "message=Nova mensagem" -d "sender\_id=4" -d "support\_ticket\_id=2" http://localhost/{pasta}/aerocontrol/backend/web/api/ticket-messages?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Atualizar bilhete |  |
| **Descrição** | Atualiza o bilhete de voo, ou seja, sempre que o utilizador efetua um checkin na aplicação android. |
| **Método** | PUT |
| **Comando Curl** | curl -X PUT -d "checkin=1" http://localhost/{pasta} /aerocontrol/backend/web/api/flight-tickets/1?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Meus bilhetes |  |
| **Descrição** | Atualiza o bilhete de voo, ou seja, sempre que o utilizador efetua um checkin na aplicação android. |
| **Método** | GET |
| **Comando Curl** | curl -X GET http://localhost/{pasta}/aerocontrol/backend/web/api/flight-tickets/my-tickets?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo** |  |

|  |  |
| --- | --- |
| Cancelar bilhete |  |
| **Descrição** | Cancela o bilhete de voo se possível, caso contrário como por exemplo se tentar cancelar o bilhete 7 dias antes do voo é lançada uma exceção que proíbe o utilizador de efetuar a ação. |
| **Método** | DELETE |
| **Comando Curl** | curl -X DELETE http://localhost/{pasta}/aerocontrol/backend/web/api/flight-tickets/5?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo Sucesso** |  |
| **Exemplo Erro** |  |

|  |  |
| --- | --- |
| Comprar bilhete |  |
| **Descrição** | Compra o bilhete de voo |
| **Método** | POST |
| **Comando Curl** | curl -X POST -d "flightGo\_id=1" -d "flightBack\_id=2" -d "numPassengers=2" -d "payment\_method=Multibanco" -d "name[0]=Rafael" -d "name[1]=Joaquim" -d "gender[0]=Masculino" -d "gender[1]=Masculino" -d "extra\_baggage[0]=0" -d "extra\_baggage[1]=1" http://localhost/{pasta}/aerocontrol/backend/web/api/flight-tickets?access-token=sOAZ\_ou8A8ZImJjZ8C5R9mYrnSn3MjdR |
| **Exemplo** |  |

# **Messaging**

Relativamente à parte de messaging no projeto foi feita uma aplicação java, esta é utilizada para melhorar o serviço de atendimento ao cliente, ou seja, para que os funcionários possam ajudar mais facilmente os clientes e atender mais rapidamente os seus tickets de suporte. Tendo em conta que o site não é em tempo real fizemos uma aplicação java que recebe uma notificação sempre que um cliente cria um ticket, e sempre que um cliente cria uma nova mensagem em um dos seus tickets.

## Canais

tickets – Canal que recebe notificação sempre que um ticket é criado e informa o id desse mesmo ticket para que o funcionário possa subscrever o ticket e receber notificações relativas ao ticket.

ticket-{id} – Canal que recebe notificação sempre que um cliente envia uma mensagem, ou seja, para melhorar o funcionamento desde sistema cria-se um novo canal sempre que um cliente inicia um novo ticket de suporte, sendo o “{id}”, o id desse mesmo ticket criado.

## Exemplo

Como é possível visualizar na imagem abaixo, consta a aplicação e à esquerda aparece a mensagem após ter sido criado um ticket através do website.

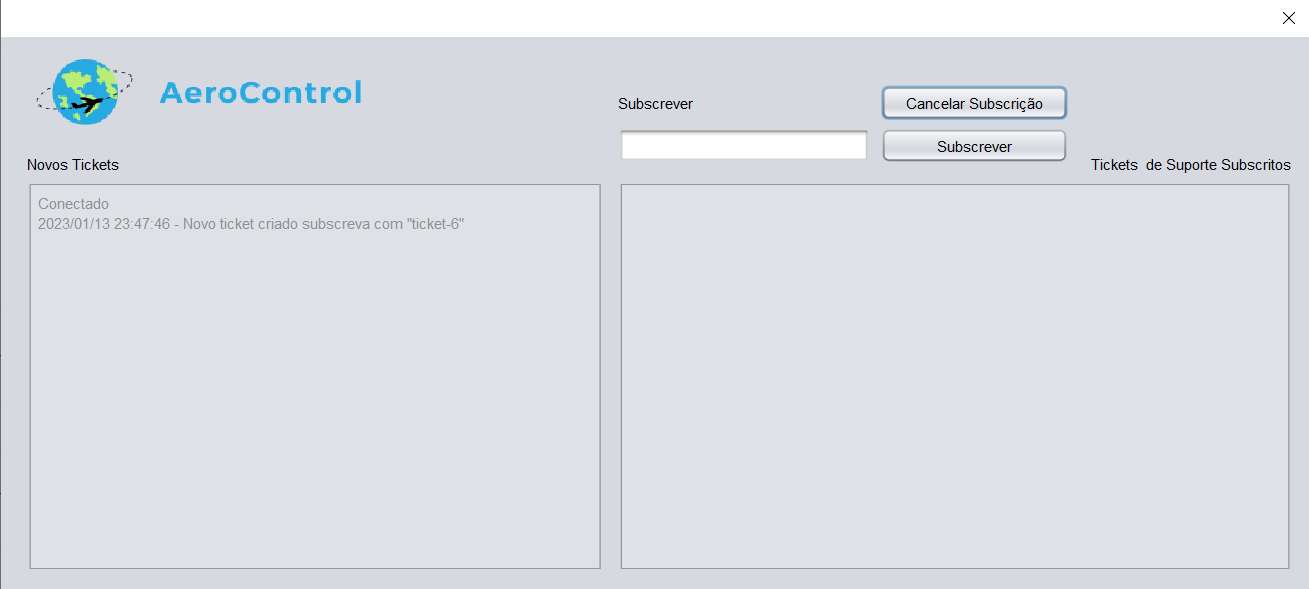


Figura 1 - Notificação após a criação de tickets

Posteriormente, após o funcionário ver a mensagem tudo o que tem a fazer é subscrever o “ticket-6” como é possível observar na seguinte figura.

Uma imagem com texto

Descrição gerada automaticamente

Figura 2 - Subscrição do ticket-6

Após a subscrição do ticket como é possível ver na figura 2, aparece no painel da esquerda uma mensagem a informar ao funcionário que o canal foi subscrito. Para além da subscrição no canal é ao mesmo tempo feita uma publicação no canal de tickets a informar a todos os funcionários que houve um funcionário a subscrever o canal “ticket-6”, para evitar que vários funcionários estejam atentos ao mesmo ticket.

Por fim, após a receção de mensagens do cliente é também enviada uma notificação, tal como tinha sido mencionado anteriormente.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3 - Receção de notificação após mensagem no ticket de suporte

Tal como é mostrado em cima é possível observar que no painel da direita, ou seja, apenas para o funcionário que tem o canal “ticket-6” subscrito recebe uma notificação a informar que o ticket já recebeu uma nova notificação, após isto o mesmo poderá dirigir-se à backend para falar com o cliente, apenas quando o mesmo responder e não estar constantemente à espera de mensagens a atualizar a página. Devido a ser possível a subscrição de múltiplos tickets em simultâneo colocou-se o canal de onde veio a mensagem para que o mesmo possa saber qual o ticket a responder.