

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
2. API REST.....	4
2.1. API Endpoints.....	4
Login	4
Signup	5
Repor Password.....	6
Editar dados.....	7
Restaurantes	10
Lojas.....	11
Meus Tickets de Suporte	12
Criar Ticket de Suporte	13
Atualizar Ticket de Suporte	13
Criar Mensagem.....	14
Atualizar bilhete	14
Meus bilhetes	15
Cancelar bilhete.....	16
Comprar bilhete.....	17
3. Messaging.....	18
3.1. Canais.....	18
3.2. Exemplo.....	18

1. 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.).

2. 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.

2.1.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	<pre>C:\>curl --header "Authorization: Basic cmFmYWVsOjEyMzQ1Njc4" -X POST http://localhost/projetofinal/aerocontrol/backend/web/api/auth/login/ { "token": "3LCUGMKxZPpzn34KJQqnRHerxvb9H-GD", "id": 1, "username": "rafael", "first_name": "Rafael", "last_name": "Bento", "gender": "Masculino", "country": "Portugal", "city": "Lisboa", "birthdate": "2002-08-07", "email": "rafael@email.pt", "phone": "912345678", "phone_country_code": "+351" }</pre>

Signup

Descrição	Permite ao cliente fazer registo na aplicação android.
Método	POST
Comando Curl	<pre>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/</pre>
Exemplo	<pre>C:\Users\rafae>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/projetofinal/aerocontrol/backend/web/api/auth/signup/ { "message": "success" }</pre>

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	<pre>curl -X POST -d "email=rafaelcarreirabento7@gmail.com" http://localhost/[pasta]/aerocontrol/backend/web/api/user/reset-password</pre>
Exemplo	<pre>C:\Users\rafae>curl -X POST -d "email=rafaelcarreirabento7@gmail.com" http://localhost/projetofinal/aerocontrol/backend/web/api/user/reset-password { "message": "Verifique o seu email para mais informações." }</pre>

Editar
dados

Descrição	Permite ao cliente alterar os seus dados pessoais.
Método	PUT
Comando Curl	<pre>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</pre>
Exemplo	<pre>C:\Users\rafae>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/projetofinal/aerocontrol/backend/web/api/user/38?access-token=-PyQHpi-6zWtEvDKVfye4is-iMTvIaKL { "username": "utilizadorupdated", "first_name": "Manuel", "last_name": "Joaquim", "gender": "Masculino", "country": "Portugal", "city": "Lisboa", "birthdate": "2000-10-8", "email": "novouser@gmail.pt", "phone": "919193123", "phone_country_code": "+351" }</pre>

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	<pre>C:\Users\rafae>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/projetofinal/aerocontrol/backend/web/api/flight/search { "flightsGo": [{ "id": 1, "terminal": "T1", "estimated_departure_date": "28-01-2023 13:30", "estimated_arrival_date": "28-01-2023 15:00", "departure_date": "28-01-2023 13:30", "arrival_date": "28-01-2023 15:00", "price": 100, "distance": 300, "state": "Previsto", "discount_percentage": 0, "passengers_left": 107, "origin_airport_id": 1, "arrival_airport_id": 3, "airplane_id": 1 }, { "id": 2, "terminal": "T2", "estimated_departure_date": "28-01-2023 08:00", "estimated_arrival_date": "28-01-2023 09:30", "departure_date": "28-01-2023 08:00", "arrival_date": "28-01-2023 09:30", "price": 130, "distance": 300, "state": "Previsto", "discount_percentage": 5, "passengers_left": 92, "origin_airport_id": 1, "arrival_airport_id": 3, "airplane_id": 2 }], "flightsBack": [{ "id": 3, "terminal": "T3", "estimated_departure_date": "30-01-2023 08:00", "estimated_arrival_date": "30-01-2023 09:30", "departure_date": "30-01-2023 08:00",</pre>

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	<pre>C:\>curl -X GET http://localhost/projetofinal/aerocontrol/backend/web/api/airports [{ "id": 1, "country": "Portugal", "city": "Lisboa", "name": "ANA Aeroporto de Lisboa", "website": "https://www.aeroporto lisboa.pt/pt/lis/home" }, { "id": 2, "country": "Portugal", "city": "Porto", "name": "ANA Aeroporto do Porto", "website": "https://www.aeroporto porto.pt/pt/opo/home" }, { "id": 3, "country": "Portugal", "city": "Faro", "name": "ANA Aeroporto de Faro", "website": "https://www.aeroporto faro.pt/pt/fao/home" }, { "id": 4, "country": "Espanha", "city": "Madrid", "name": "Aena Aeropuerto de Madrid-Barajas", "website": "https://www.aena.es/es/adolfo-suarez-madrid-baraja"</pre>

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	<pre>C:\>curl -X GET http://localhost/projetofinal/aerocontrol/backend/web/api/restaurants [{ "id": 1, "name": "Burger King", "description": "Restaurante de hamburguers", "phone": "911111111", "open_time": "05:00", "close_time": "23:59", "logo": "Logo_15-12-2022_20-49.png", "website": "https://www.burgerking.pt/home", "menu": [{ "id": 5, "item": "Brutal Bacon", "image": "Brutal Bacon_15-12-2022_20-49.png", "state": 1, "restaurant_id": 1 }, { "id": 6, "item": "King Chicken", "image": "King Chicken_15-12-2022_20-49.png", "state": 1, "restaurant_id": 1 }, { "id": 7, "item": "Whopper", "image": "Whopper_15-12-2022_20-49.png", "state": 1, "restaurant_id": 1 }, { "id": 8, "item": "Big King XXL", "image": "Big King XXL_15-12-2022_20-49.png", "state": 1, "restaurant_id": 1 }] },]</pre>

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	<pre>C:\>curl -X GET http://localhost/projetofinal/aerocontrol/backend/web/api/stores [{ "id": 1, "name": "Acium", "description": "A Acium apresenta uma proposta única no segmento de joias, refl etindo a sua energia criativa e moderna, e um estilo de vida em que o minimalismo e sim plicidade são fortemente marcados pelas cores universais preto e branco.", "phone": "912965258", "open_time": "08:00", "close_time": "20:00", "logo": "Logo_16-12-2022_12-27.jpg", "website": "https://acium.pt" }, { "id": 2, "name": "Benfica Official Store", "description": "Para os amantes de vermelho e branco visitar esta loja é obriga tório. Equipe-se com as cores do seu clube do coração.", "phone": "986542687", "open_time": "06:00", "close_time": "23:30", "logo": "Logo_16-12-2022_12-30.jpg", "website": "https://www.slbenfica.pt/pt-pt/loja" }, { "id": 3, "name": "Hugo boss", "description": "Hugo Boss é sinónimo de moda. As suas coleções dinâmicas e cosm opolitas variam entre os visuais desportivos e jovens, e os estilos mais requintados e luxuosos", "phone": "984256325", "open_time": "06:00", "close_time": "22:00", "logo": "Logo_16-12-2022_12-32.jpg", "website": "https://www.hugoboss.com/pt/pt/home" }, { "id": 4, "name": "Lacoste", "description": "Originalmente criada como marca desportiva, Lacoste continua a</pre>

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	<pre>curl -X GET http://localhost/{pasta}/aerocontrol/backend/web/api/support-tickets/my-support-tickets?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR</pre>
Exemplo	<pre>C:\>curl -X GET http://localhost/projetofinal/aerocontrol/backend/web/api/support-tickets/my-support-tickets?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR [{ "id": 1, "title": "Camisola Perdida", "state": "Concluido", "messages": [{ "id": 1, "message": "Bom dia, gostava de saber se foi encontrada uma camisola no voo Lisboa Faro no dia 3 de janeiro.", "sender": "antonio" }, { "id": 2, "message": "Olá, a sua camisola é uma camisola às riscas azul e branca?", "sender": "pedro" }, { "id": 3, "message": "Sim!", "sender": "antonio" }, { "id": 4, "message": "Confira se é a camisola indicada na sua lista de bilhetes, em caso de ser o indicado conclua o ticket, obrigado.", "sender": "pedro" }], "items": [{ "id": 1, "description": "Camisa às riscas azul e branca.", "state": "Entregue", "image": "19-12-2022_20-03-20.jpg" }] },]</pre>

Criar Ticket de Suporte

Descrição	Cria um ticket de suporte com uma mensagem.
Método	POST
Comando Curl	<pre>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</pre>
Exemplo	<pre>C:\Users\rafae>curl -X POST -d "title=Casaco perdido" -d "message=Perdi um caso dia 3 de janeiro" http://localhost/projetofinal/aerocontrol/backend/web/api/support-tickets?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR { "title": "Casaco perdido", "message": "Perdi um caso dia 3 de janeiro", "client_id": 4 }</pre>

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	<pre>curl -X put http://localhost/{pasta}/aerocontrol/backend/web/api/support-tickets/1?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR</pre>
Exemplo	<pre>C:\>curl -X put http://localhost/projetofinal/aerocontrol/backend/web/api/support-tickets/1?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR { "state": "Concluido" }</pre>

Criar

Mensagem

Descrição	Cria uma mensagem no ticket de suporte através dos parâmetros especificados por POST.
Método	POST
Comando Curl	<pre>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</pre>
Exemplo	<pre>C:\>curl -X POST -d "TicketMessageForm[message]=Nova mensagem" -d "sender_id=4" -d "support_ticket_id=2" http://localhost/projetofinal/aerocontrol/backend/web/api/ticket-messages?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR { "message": "Nova mensagem", "sender_id": "4", "support_ticket_id": "2" }</pre>

Atualizar

bilhete

Descrição	Atualiza o bilhete de voo, ou seja, sempre que o utilizador efetua um checkin na aplicação android.
Método	PUT
Comando Curl	<pre>curl -X PUT -d "checkin=1" http://localhost/{pasta}/aerocontrol/backend/web/api/flight-tickets/1?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR</pre>
Exemplo	<pre>C:\>curl -X PUT -d "checkin=1" http://localhost/projetofinal/aerocontrol/backend/web/api/flight-tickets/1?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR { "message": "Check-in efetuado" }</pre>

Meus bilhetes

Descrição	Atualiza o bilhete de voo, ou seja, sempre que o utilizador efetua um checkin na aplicação android.
Método	GET
Comando Curl	<pre>curl -X GET http://localhost/{pasta}/aerocontrol/backend/web/api/flight-tickets/my-tickets?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR</pre>
Exemplo	<pre>C:\>curl -X GET http://localhost/projetofinal/aerocontrol/backend/web/api/flight-tickets/my-tickets?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR [{ "id": 1, "payment_method": "Cartão de crédito", "flightState": "Previsto", "flightOrigin": "Lisboa", "flightArrival": "Faro", "flightOriginTime": "13:30", "flightArrivalTime": "15:00", "terminal": "T1", "originalPrice": 100, "paidPrice": 100, "flightDate": "28-01-2023", "purchaseDate": "2022-12-25 13:30:00", "distance": 300, "checkin": 1, "passengers": [{ "id": 1, "name": "António Alberto", "gender": "Masculino", "seat": "A1", "extra_baggage": 0 }] }, { "id": 3, "payment_method": "Multibanco", "flightState": "Previsto", "flightOrigin": "Faro", "flightArrival": "Lisboa", "flightOriginTime": "08:00", "flightArrivalTime": "09:30",</pre>

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	<pre>curl -X DELETE http://localhost/{pasta}/aerocontrol/backend/web/api/flight- tickets/5?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR</pre>
Exemplo Sucesso	<pre>C:\>curl -X DELETE http://localhost/projetofinal/aerocontrol/backend/web/api/flight- tickets/5?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR { "message": "Bilhete cancelado com sucesso" }</pre>
Exemplo Erro	<pre>C:\>curl -X DELETE http://localhost/projetofinal/aerocontrol/backend/web/api/flight- tickets/6?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR { "name": "Forbidden", "message": "Impossível cancelar o bilhete 7 dias antes do voo, por favor contac te o suporte!", "code": 0, "status": 403, "type": "yii\web\\ForbiddenHttpException" }</pre>

Comprar bilhete

Descrição	Compra o bilhete de voo
Método	POST
Comando Curl	<pre>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</pre>
Exemplo	<pre>C:\Users\rafae>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/projetofinal/aerocontrol/backend/web/api/flight-tickets?access-token=sOAZ_ou8A8ZImJjZ8C5R9mYrnSn3MjdR { "message": "success" }</pre>

3. 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.

3.1.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.

3.2.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.

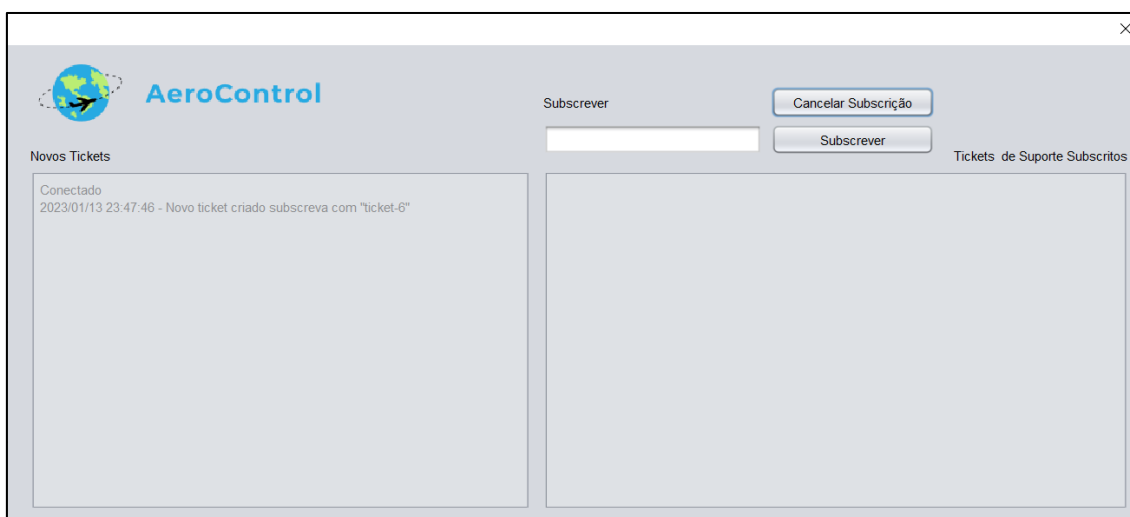


Figura 1 - Notificação após a criação de tickets

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.

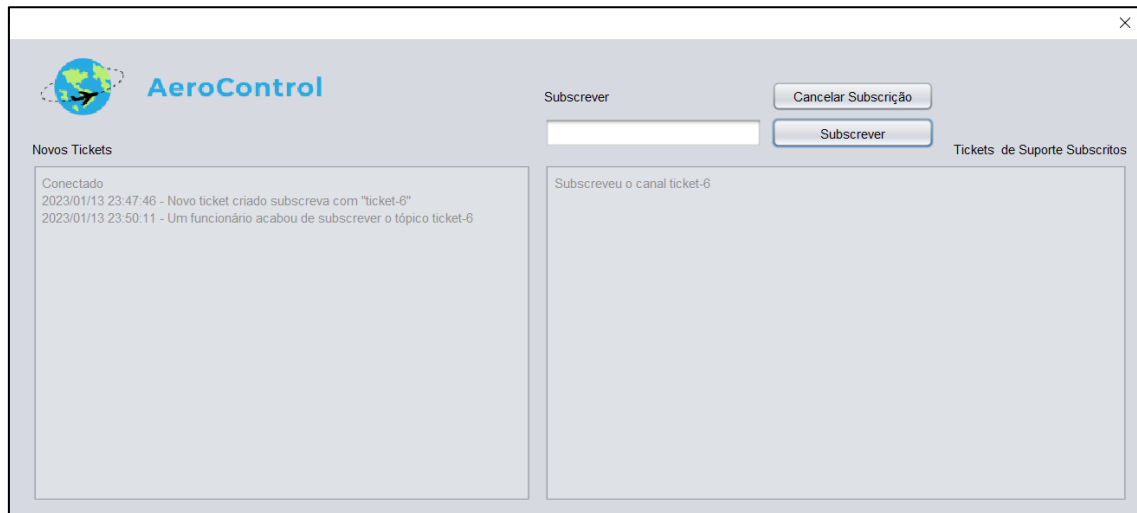


Figura 2 - Subscrição do ticket-6

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.

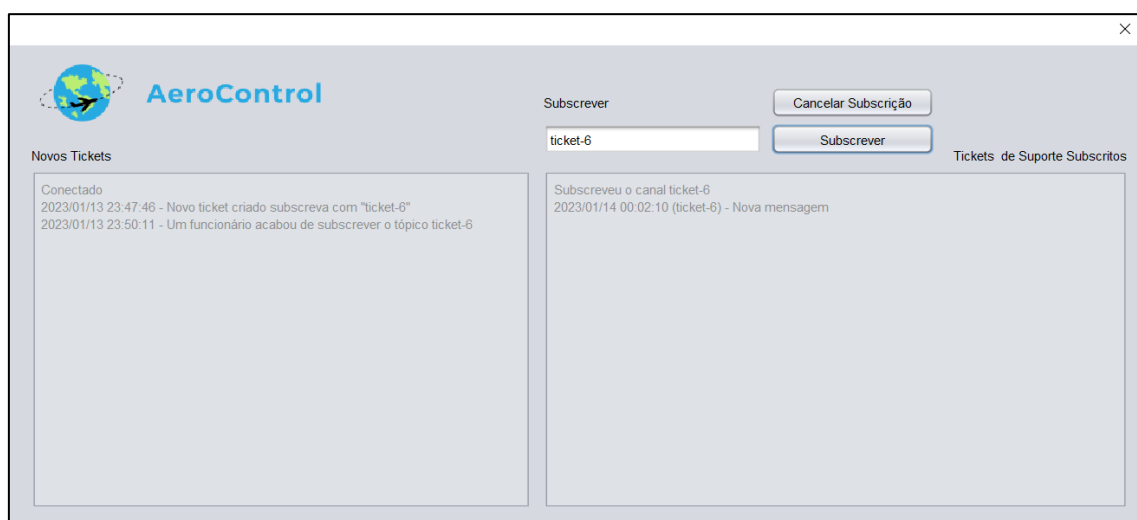


Figura 3 - Receção de notificação após mensagem no ticket de suporte

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.