|  |  |  |  |
| --- | --- | --- | --- |
| Uma imagem com Tipo de letra, texto, Gráficos, captura de ecrã  Descrição gerada automaticamente | | Curso de TeSP em  Programação de Sistemas de Informação  Serviços e Interoperabilidade de Sistemas (SIS) | |
|  |  | |  |
| **Ano Letivo 2023/2024** |  | | **2º Ano, 1º Semestre** |
|  | | | |
| **Serviços e Interoperabilidade de Sistemas** | | | |

Uma imagem com logótipo, palmeira, fruta, texto

Descrição gerada automaticamente

**Relatório do projeto da Unidade Curricular de SIS**

***Lustitânia Travel***

|  |  |
| --- | --- |
|  |  |
| **Grupo:** PL1-G4 |  |
| **Nº** 2220863 | Diana Pereira Catarino |
| **Nº** 2211923 | Maria Inês Jesus |
|  |  |

ÍNDICE

[1 Contextualização do projeto 4](#_Toc155971650)

[2 Operações necessárias para aplicação android 5](#_Toc155971651)

[3 Detalhes da api restful 7](#_Toc155971652)

[3.1 Descrição Geral da API 7](#_Toc155971653)

[3.2 Endpoints e Exemplos com cURL 7](#_Toc155971654)

[4 Funcionalidades de messaging 14](#_Toc155971655)

[5 Informações de Hospedagem 16](#_Toc155971656)

# Contextualização do projeto

O presente documento foi realizado no âmbito da Unidade Curricular de Serviços e Interoperabilidade de Sistemas do Curso Técnico Superior Profissional de Programação em Sistemas de Informação. Este documento constata o relatório sobre o projeto realizado para a Unidade Curricular de Serviços e Interoperabilidade de Sistemas.

Este sistema pretende representar um modelo de gestão de uma agência de viagens. Esta agência reserva e permite ao utilizador reservar de forma autónoma estadias em Portugal Continental. O tipo de alojamentos possíveis de reservar são: hotéis, resorts e alojamento local. A empresa conta com 2 formas possíveis de reservar estadias, através da Mobile App (utilizado apenas por cliente e gerido pelos administradores), através da App Web (utilizado por clientes e funcionários, em cenário de agência local, e gerido pelos administradores). A utilização das Apps pelos clientes permite a reserva de férias e estadias em qualquer local e a qualquer momento, dependendo apenas da confirmação por parte do alojamento.

Com base naqueles que foram os conhecimentos lecionados durante o presente semestre, a proposta consiste no desenvolvimento de uma Mobile App, de uma Web App, de uma API Restfull e de uma Base de Dados que satisfaça as necessidades do negócio escolhido.

Para a realização deste projeto são utilizadas as ferramentas e software, baseadas pelas quais foram utilizadas em tempo letivo, como o WAMP (Windows) para estabelecimento de um servidor local, MySQL Workbench, e representações estruturais através de diagramas.

# Operações necessárias para aplicação android

Para o desenvolvimento da aplicação android serão necessárias operações que provêm da API RESTful. A aplicação irá consumir diversos dados necessários para o bom funcionamento da mesma estar assegurado. Exemplos destes dados provêm das seguintes entidades:

* Fornecedores

Todos os alojamentos deste sistema são entendidos como fornecedores. Denominam-se fornecedores devido ao facto de prestarem um serviço ao cliente. Deste modo, através da aplicação será possível obter: a listagem de fornecedores do tipo “Hotel”, a listagem dos fornecedores que estão localizados em Lisboa, a listagem de todos os comentários do fornecedor com o ID igual sete que sejam sido inseridos no dia 05-01-2023, a média da avaliação do fornecedor com o ID igual a um, a verificação da disponibilidade relativamente partindo do fornecedor e a contagem de todos os fornecedores que estejam localizados em Lisboa.

* Reservas

A reserva de estadias de forma autónoma é uma das funcionalidades presente na aplicação. O sistema móvel android irá consumir os seguintes dados: obter a listagem de todas as reservas confirmadas(por user?), atualizar o estado da reserva com o ID igual a sete para “Confirmada”, cancelar uma reserva com o ID igual a sete, calcular e demonstrar a taxa de reservas total do sistema e mostrar a listagem de detalhes da reserva com ID igual a um.

* Fatura

As operações necessárias à aplicação android desta entidade são: listar todos os detalhes de uma fatura em especifico, listar todas as faturas de um cliente especifico através do seu nome e gerar uma fatura apenas com o nome do cliente e o ID da reserva já existente.

* Carrinho

O carrinho é uma das principais funcionalidades para o bom funcionamento da reserva de estadias. Por essa razão, são consumidos os seguintes dados: cálculo do total do carrinho pelo nome especifico do cliente, inclusão no carrinho de uma reserva apenas e através do nome do fornecedor, remoção de uma reserva do carrinho pelo ID do fornecedor e finalização do carrinho e procedimento à fase seguinte através do identificador único do mesmo.

* User

O User é responsável por várias funcionalidades importantes e úteis ao utilizador. Como exemplo das funcionalidades, permite o registo de um novo utilizador, permite efetuar o login e consultar os dados de um utilizador especificado pelo seu username. O login é sem dúvida uma peca fundamental para um bom funcionamento de uma aplicação móvel. Para além da autenticação é fulcral no desempenho da autorização no acesso a certas partes do sistema.

# Detalhes da api restful

Na presente secção são abordados todos os detalhes da API implementada no sistema, de modo a sustentar e completar a aplicação móvel.

## Descrição Geral da API

A API desenvolvida pretende dar a consumir dados a uma aplicação móvel de modo a que seja possível a reserva de estadias e férias em Portugal Continental. Deste modo são tidos em conta os controladores dos Fornecedores, das Reservas, das Faturas, do Carrinho e dos Users.

## Endpoints e Exemplos com cURL

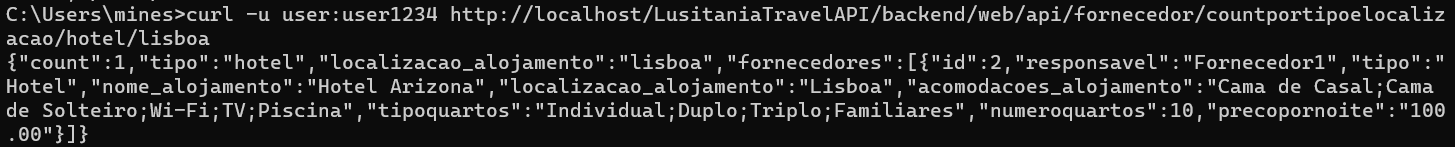
Todas as extrapatterns utilizadas têm valores associados, tokens, que devem seguir um conjunto de regras definidas na API. O projeto conta com controladores para as quatro já referidas anteriormente.

* Fornecedor

Todas as extrapatterns utilizadas têm valores associados, tokens, que devem seguir um conjunto de regras definidas na API. Os tokens associados a esta entidade são: o ID do alojamento (valor inteiro e positivo), o tipo do alojamento (caracteres alfanuméricos e hífens) e a localização do alojamento e a data (qualquer caractere exceto ‘/’).

1. GET countportipoelocalizacao/{tipo}/{localizacao\_alojamento}

Este endpoint efetua a contagem de todos os alojamentos por tipo e localização a serem definidos.



Neste exemplo, é calculado o valor de X obtido através de todos os Hóteis existentes na cidade de Lisboa.

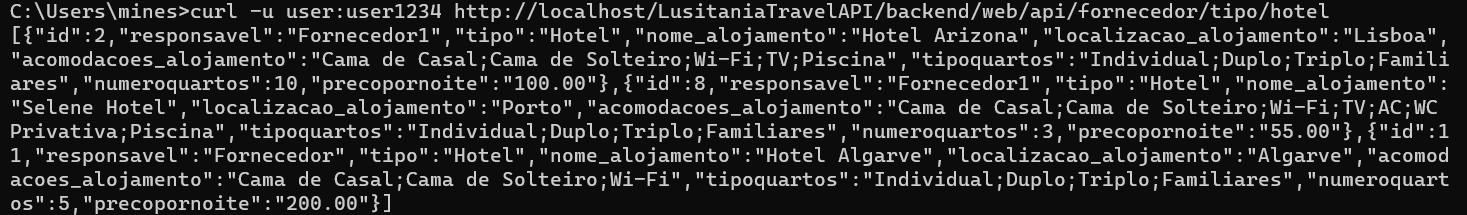
1. GET alojamentos

Através deste endpoint obtêm-se a listagem de todos os alojamentos que existem na aplicação.



1. GET tipo/{tipo}

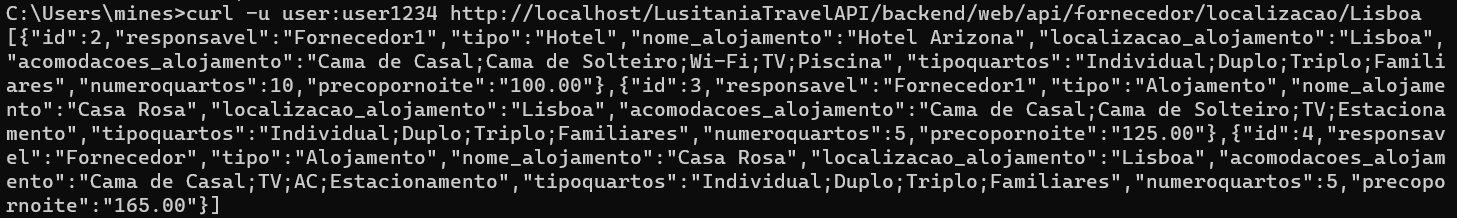
Nesta extrapattern, podemos obter a listagem de todos os alojementos do mesmo tipo. O tipo é um atributo que indica se o fornecedor é um hotel, um alojamento local, etc.



Neste exemplo, é obtida a listagem de todos os hotéis disponíveis no sistema.

1. GET localizacao/{localizacao\_alojamento}

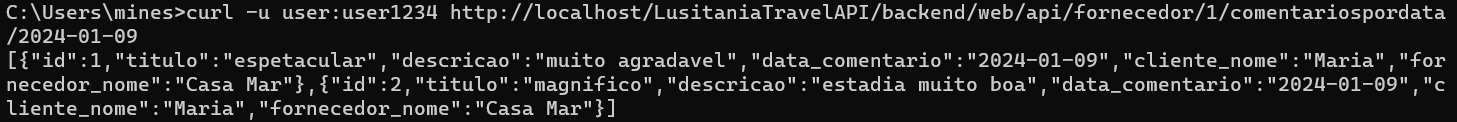
Este endpoint, indica a listagem de todos os fornecedores sediados numa cidade a ser definida pelo utilizador.



No exemplo a cima, obtêm-se a listagem de todos os alojamentos existentes da cidade de Lisboa.

1. GET {id}/comentariospordata/{data} => comentariospordata

A extrapattern devolverá a lista de todos os comentários efetuados num dia especifico a um fornecedor identificado através do seu ID.



No exemplo, é obtida a lista de dos comentários feitos no dia 2024-01-09 relativos ao fornecedor com ID igual a 1.

1. GET {id}/avaliacoesmedia => avaliacoesmedia

Este endpoint devolve a média de todas as avaliações referentes a um determinado alojamento.



A cima, foi calculada a média das avaliações atribuídas ao fornecedor com o ID igual a 1.

* Reserva

À semelhança da entidade Fornecedor o token relativo ao ID deve apresentar um valor inteiro e positivo.

1. PUT {id}/confirmar => confirmarreserva

Através desta extrapattern, é possível atualizar o estado da confirmação para “Confirmada” de uma reserva através do ID que define a reserva pretendida.



No exemplo a cima, é confirmada a reserva com o ID igual a 13.

1. PUT {id}/cancelar => cancelarreserva

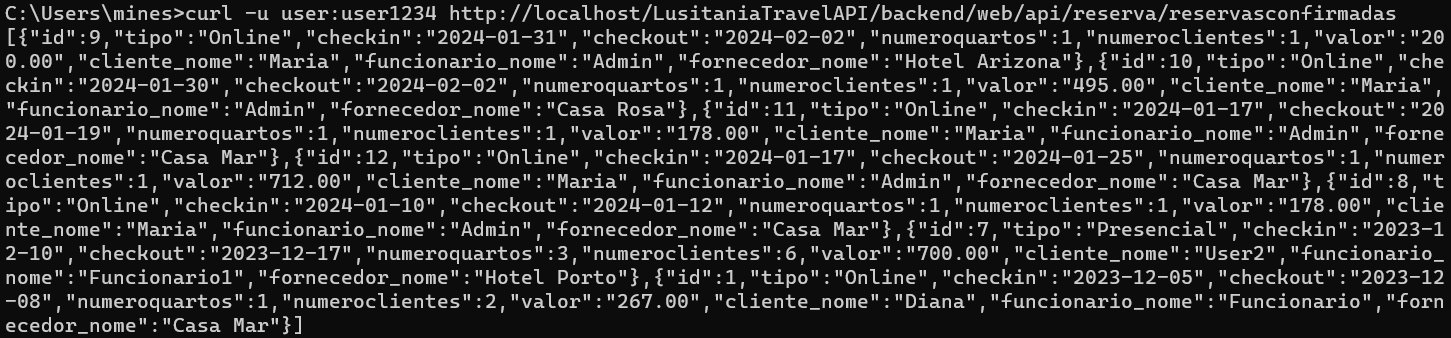
À semelhança do endpoint a cima, é pretendido atualizar o estado de confirmação de uma reserva para “Cancelada” através do ID da reserva escolhida.



No exemplo, é cancelada a reserva com o ID igual a 13.

1. GET reservasconfirmadas => reservasconfirmadas

Através deste endpoint é possível apresentar todas a lista de todas as reservas que já se encontram confirmadas.



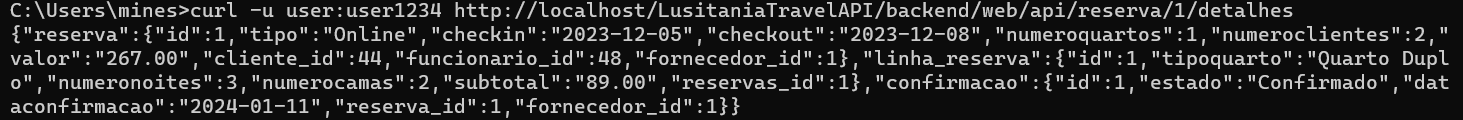
1. GET taxareservas => taxareservas

Nesta extra pattern é calculado a percentagem da taxa de reserva do sistema.



1. GET {id}/detalhes' => 'detalhesreserva

O endpoint lista de todos os detalhes de uma reserva com o ID a ser definido.



A cima, é obtida a lista de detalhes da reserva com o ID igual a X.

* Carrinho

Os tokens definidos para o carrinho são os seguintes: ID do carrinho e o ID do fornecedor (valores inteiros e positivos), o nome do cliente (deve conter apenas caracteres alfanuméricos).

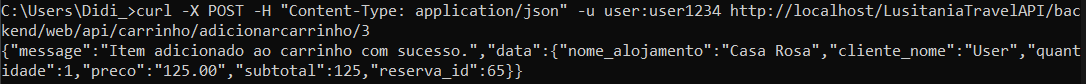
1. GET calculartotal/{nomecliente}=> calculartotal

Neste endpoint, é possível obter o valor do resultado do cálculo do total do carrinho através do nome do cliente.



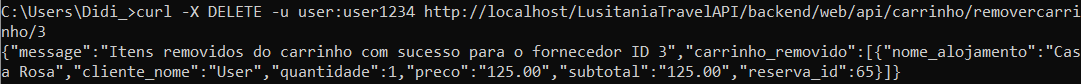
1. POST adicionarcarrinho/{fornecedorid} =>adicionarcarrinho

Esta extrapattern, torna possível adicionar um item ao carrinho através do ID do fornecedor escolhido.



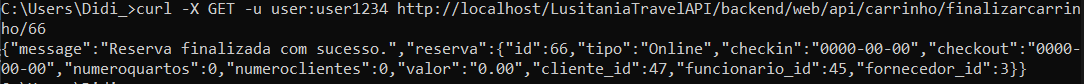
1. DELETE removercarrinho/{fornecedorid} => removercarrinho

Nesta extrapattern pretende-se eliminar um item do carrinho através do ID do fornecedor selecionado.



1. PUT finalizarcarrinho/{fornecedorid} => finalizarcarrinho

Este endpoint pretende atualizar o carrinho através do ID do fornecedor.

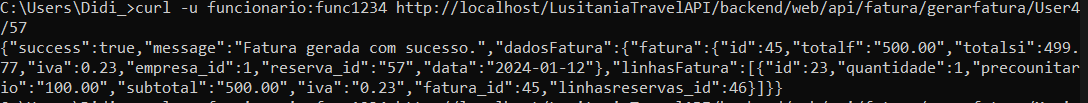


* Fatura

Os tokens da fatura são o ID da mesma, o ID da reserva associada e o nome do cliente. Ambos os valores relativos ao ID devem ser inteiros e positivos, enquanto que o nome deve conter apenas caracteres alfanuméricos.

1. 'GET gerarfatura/{nomecliente}/{reserva\_id}' => 'gerarfatura'

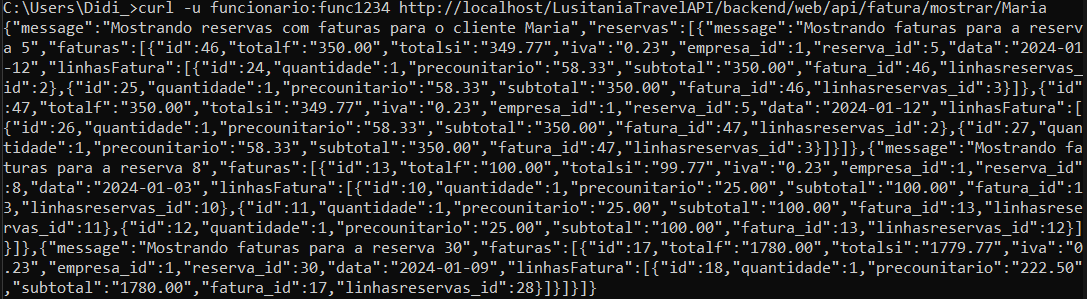
Neste endpoint é gerada a fatura através do nome do cliente e do ID da reserva à qual está associada.



No exemplo a cima é possível observar a fatura gerada através do utilizador com o nome igual a User4 e o ID da reserva igual a 57.

1. 'GET mostrar/{nomecliente}' => 'mostrarfatura'

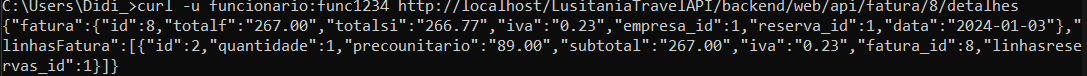
Através desta extrapattern é possível visualizar todas as faturas associadas ao cliente pesquisando apenas pelo nome.



Neste exemplo, obtém-se a listagem de todas as faturas através do nome do cliente.

1. 'GET {id}/detalhes' => 'detalhesfatura'

Esta extrapattern permite visualizar os detalhes de uma fatura espeicificada pelo ID da mesma.



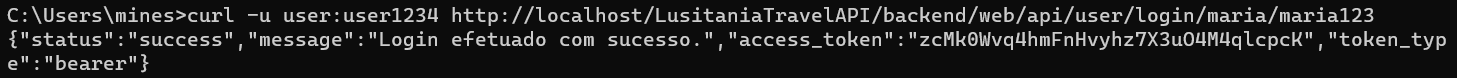
A cima, foi obtida a listagem dos detalhes de uma fatura especificada através do seu ID.

* User

O controlador do User possui três tokens, o ID, o username e password do Utilizador. O ID deve conter um valor inteiro e positivo e o username e a password deve conter letras e números.

1. 'GET login/{username}/{password}' => 'login'

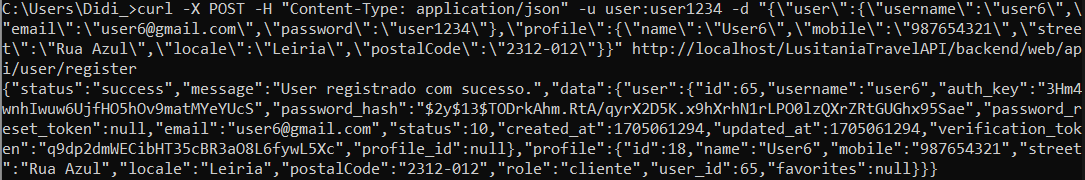
O endpoint a cima efetua o login de um utilizador através do seu username e password.



Neste exemplo, foi efetuado o login da conta do username com o valor “maria”.

1. 'POST register' => 'register'

Esta extrapattern, permite fazer um registo de um novo utilizador que pretenda utilizar o sistema.



1. 'GET mostrar/{username}' => 'mostraruser'

Este endpoint mostra o perfil do user através do seu nome de utilizador.



# Funcionalidades de messaging

Na presente secção são abordadas as funcionalidades de messaging que foram integradas no sistema desenvolvido. Durante o desenvolvimento do sistema, entendeu-se que os controladores que faria sentido implementar no sistema de messaging seriam a Reserva e a Fatura.

Deste modo, um dos principais objetivos é emitir uma notificação quando o estado de confirmação de uma reserva é alterado para um dos valores seguintes: “confirmada” ou “cancelada”.

* Reserva confirmada com sucesso

O nome dado ao canal para cancelar uma reserva é “reserva”. Ao tentar guardar o novo estado da reserva, é tentado estabelecer uma ligação com o servidor MQTT. Caso essa tentativa seja bem-sucedida aparecerá no canal uma mensagem publicada a dizer “Reserva confirmada com o ID: ” seguida do valor do ID da reserva, e a conecção com o servidor MQTT é fechada. Após o sucesso de todas a etapas a função “actionConfirmarReserva()” devolve ainda um array indicando que a reserva foi confirmada com sucesso.

* Reserva cancelada com sucesso

O nome dado ao canal para confirmar uma reserva é “reserva”. À semelhança da mensagem de sucesso ao confirmar a reserva, o cancelamento da reserva funciona exatamente da mesma forma. A diferença consiste apenas na mensagem que é publicada no canal. Em caso de sucesso a mensagem é “Reserva cancelada com o ID: ” seguida do valor do ID da reserva. A função “actionCancelarReserva()” devolve também um array indicando que a reserva foi cancelada com sucesso.

No caso do controlador da fatura, o objetivo do messaging foca-se apenas em emitir uma notificação de sucesso caso a fatura seja gerada com sucesso.

* Fatura gerada com sucesso

O canal para estas notificações denomina-se “fatura”. Na tentativa de guardar a fatura gerada, é efetuada a tentativa de conexão com o servidor MQTT. Se esta ligação for bem sucedida será publicada uma mensagem contento o texto “Fatura confirmada com o ID: ” De seguida a ligação ao servidor MQTT é concluída e a função “actionGerarFatura()” devolve um array que indica se a fatura foi gerada ou não com sucesso.

# Informações de Hospedagem

Na presente secção abordam-se todas as informações relativas à hospedagem no servidor de produção onde se encontra alojada a API.

Para o acesso ao servidor de produção não são necessárias quaisquer credencias. O endereço a aceder para efetuar o acesso é:

- <http://localhost/LusitaniaTravelAPI/backend/web/api>