



**Faculdade de Design,
Tecnologia e Comunicação**
Universidade Europeia

Overlap

<https://github.com/MarianaCGoncalves/Overlap>

Duarte Cavaleiro	20211026
Mariana Gonçalves	20210539
Miguel Bernardo	20211066

Índice

1-Descrição da app	3
1.1-Breve Descrição	3
1.2-Objetivos	3
1.3-Funcionalidades	3
1.4-Público Alvo	3
2-Apps semelhantes	3
3-Guião	5
4-Solução	6
4.1- Descrição genérica da solução implementada	6
4.2-Enquadramento das Unidades Curriculares	6
4.3-Requisitos técnicos	7
4.4-Arquitetura da solução	7
4.5-Tecnologias a utilizar	7
5-Calendarização do Projeto	9
6-Modelo de Domínio	9
7-Diagrama de Classes	10
8-Documentação REST.	11
9-Bibliografia	27

1-Descrição da app

1.1-Breve Descrição

Overlap é uma aplicação sobre Fórmula 1 onde os utilizadores poderão encontrar-se com outros para falarem sobre Fórmula 1 e ver as corridas de F1.

1.2-Objetivos

A ideia da aplicação surgiu devido ao aumento da popularidade de F1 e como ainda não há uma aplicação para encontrar pessoas que também gostam de F1, Overlap foi criada para responder a esta necessidade.

1.3-Funcionalidades

A aplicação Overlap consiste em muitas funções sendo a principal e mais importante, a opção de conseguir juntar-se a um grupo da sua equipa favorita, ou de um piloto favorito, podendo assim comunicar-se com outros fãs e partilhar as suas opiniões. Outra função presente na nossa aplicação é a disponibilidade das pessoas presentes no grupo conseguirem-se encontrar num local para ver as corridas ou até mesmo só para partilharem as suas opiniões pessoalmente.

1.4-Público Alvo

A nossa aplicação tem como público alvo pessoas entre os 18 e 50 anos de idade que tenham interesse em Fórmula 1.

2-Apps semelhantes

Como foi previamente referido, não existem aplicações daí o desenvolvimento da nossa.

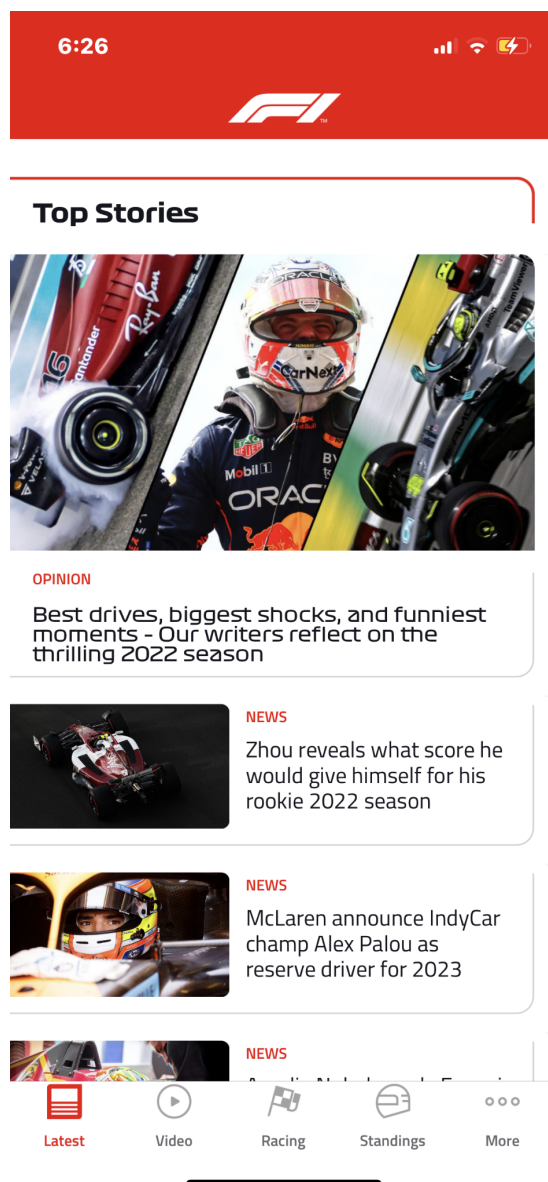
Algumas aplicações de Fórmula 1 que existem são as aplicações oficiais da FIA (Federação Internacional do Automóvel)

A aplicação mais importante e mais conhecida é a:

-Fórmula 1



Onde o utilizador tem acesso a notícias sobre Fórmula 1, as corridas que irão decorrer tal como um Live Feed da corrida a decorrer e apresenta também uma página onde demonstra os pontos que cada piloto e equipa têm.



3-Guião

Após entrar na aplicação, o utilizador vai encontrar uma caixa de texto onde irá inserir o nome, email e palavra-passe para a sua conta, em seguida o utilizador poderá escolher os desportos que deseja seguir e as equipas favoritas.

Após o registo na aplicação o utilizador poderá ver a Home Page onde se encontram os grupos aos quais se juntou, caso o utilizador não esteja em nenhum grupo, serão sugeridos grupos onde pessoas com gostos semelhantes se encontram.

O utilizador encontrará três botões na parte inferior do ecrã, uma pessoa, uma casa e um sinal GPS no qual poderá visualizar os eventos que estão a decorrer perto da sua localização.

Os eventos são criados nos grupos, nos quais os membros definem uma localização para o decorrer do mesmo.

Não será preciso um método de verificação que pertence ao grupo, pois o objetivo da aplicação é encontrar pessoas com o mesmo gosto desportivo, mesmo que a pessoa não esteja a utilizar o Overlap.

Breve introdução de personas:

- Gustavo Costa, 40 anos, é da Ferrari e da Red Bull e trabalha no bar SPOT no Bairro Alto.
- Manuel Gusmão, 24 anos, e torce também pela Ferrari e trabalha como sapateiro.
- Joana Pinto, 19 anos, gosta da Alpine e está a tirar um curso de fotografia no IADE.
- Manuel Gusmão, 24 anos, e torce também pela Ferrari e trabalha como sapateiro.
- Pablo Diaz, 29 anos natural de Sevilha, passa bastante tempo em Portugal devido ao trabalho. Gosta de assistir corridas de F1 casualmente, não tem uma equipa favorita.
- Maria Pereira, 27 anos, trabalha na UberEats. Torce pela Alpine.
- José Almeida, 25 anos, está a treinar para ser piloto de avião. Adora a Alpine.
- Frederico Pinheiro, 38 anos, trabalha nas Finanças. A Red Bull e a Ferrari são as suas equipas favoritas.

- Júlia Pinheiro, 44 anos, apresentadora na XIC. Seguiu os passos do marido e também torce pela Red Bull e Ferrari.

Um domingo, o Gustavo Costa estava a sair do seu turno da manhã do bar SPOT no Bairro Alto, mas não queria nada perder a corrida nem vê-la sozinho. A corrida começa às 15h em Portugal, então o Gustavo instala o Overlap em busca de companhia para ver a corrida. Faz o registo com o seu email, nome e palavra-passe. Após o registo é-lhe apresentada uma tela para selecionar a sua equipa favorita, na qual o Gustavo seleciona a Ferrari e Red Bull, e a Overlap recomenda-lhe grupos que têm como foco principal a equipa da Ferrari. Ao entrar num grupo da Ferrari, encontra Manuel Gusmão, que também está interessado em ver a corrida com alguém e criou um “evento” marcado para às 14:45h relativamente perto de Santos. Vendo o evento do Manuel, o Gustavo interessa-se e decide participar no evento. Manda mensagem no post do Manuel, a perguntar se poderia ir almoçar com ele enquanto via a corrida, combinam o ponto de encontro num restaurante em Santos. No fim de assistirem a corrida, eles seguem o perfil um do outro.

Joana Pinto saiu da aula Edição e Design às 17h, e ao sair da faculdade encontrou os habituais trabalhadores da UberEats e ao passar por eles, ouviu mencionarem a última corrida de fórmula 1. Joana não conteve o entusiasmo e mencionou o gosto pela equipa Alpine. Entre os funcionários da UberEats estava Maria Pereira, torcedora também da Alpine. Maria perguntou à Joana se estaria interessada em participar num evento organizado pelo José Almeida na Overlap de amanhã, exclusivo para torcedores da Alpine. Joana desconhecia a aplicação Overlap, e procedeu à criação da conta. Num clique junta-se ao grupo da Alpine, e partilha interesse em participar no evento do dia seguinte. No dia seguinte Joana e Maria destinam-se a Algés, a localização onde marcaram o evento. Encontram-se com o José e os outros e decidem ir ao café mais próximo falar das estratégias da Alpine para as próximas corridas.

4-Solução

4.1- Descrição genérica da solução implementada

Devido ao objetivo da nossa aplicação de ajudar os seus utilizadores a encontrarem outras pessoas com o mesmo de Fórmula 1, a nossa aplicação utiliza geolocalização para habilitar os utilizadores a encontrarem-se com outros para realizarem os seus eventos relacionados com Fórmula 1

4.2-Enquadramento das Unidades Curriculares

Para a realização da aplicação Overlap, será necessário enquadrar as diversas UCs. Necessitamos de uma base de dados para gerir os dados dos utilizadores (nome, email, idade) assim como armazenar dados das equipas. Uma interface para o utilizador interagir com a aplicação, feita no Android Studio, com todas as funcionalidades necessárias para satisfazer o usuário, por exemplo escolher a equipa favorita, juntar-se a grupos do seu interesse, alterar o seu perfil. Consequentemente para estabelecer uma comunicação entre estes dois (base de dados e interface da app), precisamos de uma API REST, pois é utilizada para estruturar qualquer modelo de aplicações web/móvel para os dias atuais, onde existe um alto volume de trocas de dados processados de forma assíncrona. Por fim, a unidade curricular Competências Comunicacionais terá um grande impacto nas apresentações que haverá ao longo do semestre.

4.3-Requisitos técnicos

Java: usado para desenvolver a REST API e a aplicação móvel.

SQLpostgres: para gerir os dados dos utilizadores.

4.4-Arquitetura da solução

Overlap estará dividida em três partes:

Base de Dados, onde serão armazenados os dados dos utilizadores, assim como os das equipas.

API, é utilizada para estruturar qualquer modelo de aplicações web/móvel para os dias atuais, onde existe um alto volume de trocas de dados processados de forma assíncrona.

Aplicação Móvel, utilizador vai estar em contato direto com a interface da aplicação, onde estarão presentes as funcionalidades como por exemplo: login, entrar num grupo, marcar um evento(...).

A API vai permitir a comunicação entre a base de dados e a aplicação móvel.

4.5-Tecnologias a utilizar

SpringBoot: <https://spring.io/projects/spring-boot>

VS Code: <https://code.visualstudio.com/>

Figma (MockUps): <https://www.figma.com/>

PgAdmin: <https://www.pgadmin.org/download/>

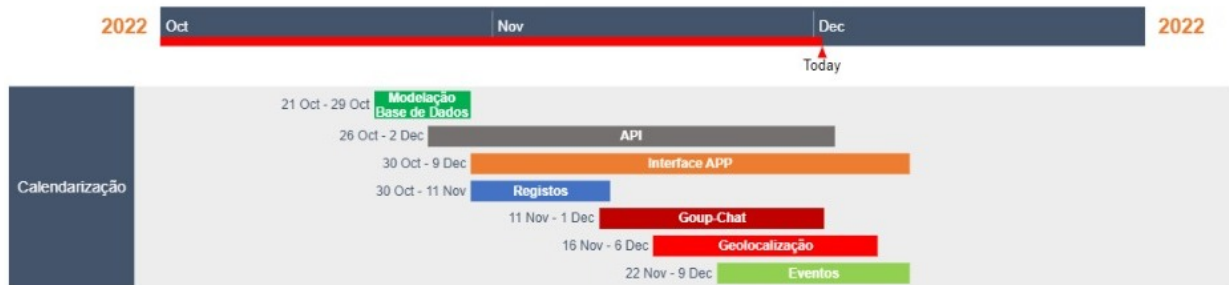
AndroidStudio: <https://developer.android.com/studio>

Retrofit2: <https://square.github.io/retrofit/>

Swagger: <https://swagger.io/>

5-Planeamento e calendarização

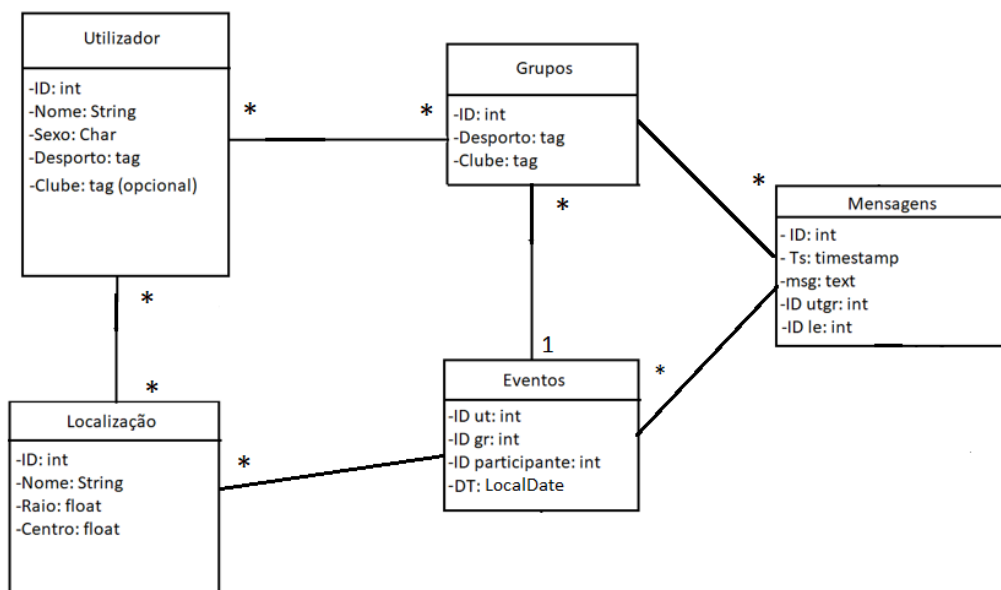
Calendarização do Overlap



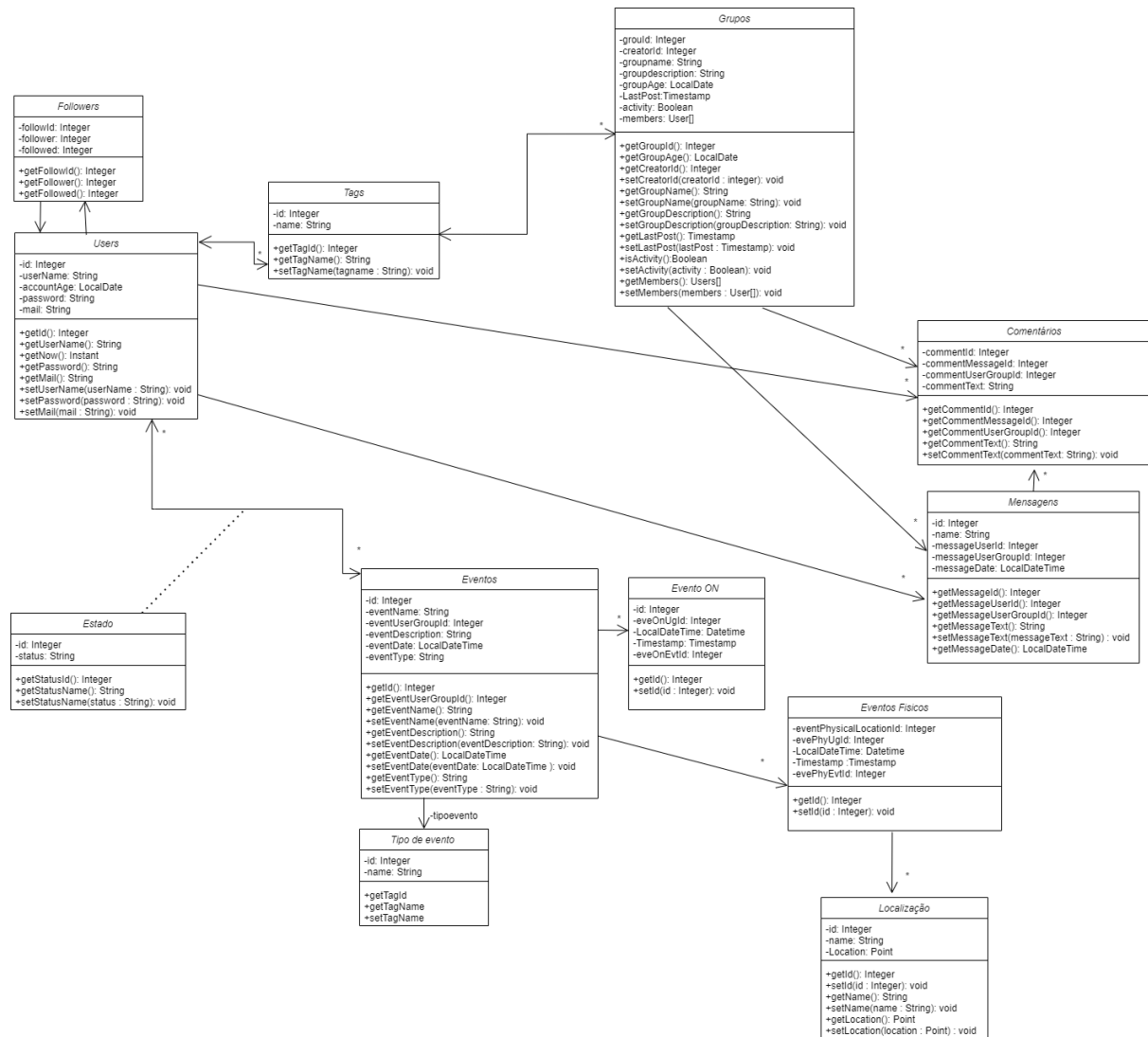
Os membros realizarão cada tarefa em conjunto.

6-Modelo do Domínio

Modelo do Domínio



7-Diagrama de Classes



8-Documentação REST.

Recurso Users (api/users)

Listar todos os users

Devolve a lista de todos os users. Funcionalidade necessária para pesquisa de users.

/api/users/ (**get**)

Sucesso (200):

```
[{"userName":"Gustavo Costa","password":"123","mail":"gcosta@gmail.com","id":1}, {"userName":"Joana Pinto","password":"123","mail":"joanapinto@gmail.com","id":2}, {"userName":"Francisco Andrade","password":"123","mail":"frandrade@gmail.com","id":3}, {"userName":"Manuel Gusmão","password":"123","mail":"MG@gmail.com","id":4}, {"userName":"Pablo Diaz","password":"123","mail":"PDiaz@gmail.com","id":5}, {"userName":"Maria Pereira","password":"123","mail":"maripe@gmail.com","id":6}, {"userName":"José Almeida","password":"123","mail":"JAlmeida@gmail.com","id":7}, {"userName":"Frederico Pinheiro","password":"123","mail":"pinheiro@gmail.com","id":8}, {"userName":"Júlia Pinheiro","password":"123","mail":"juliapinheiro@gmail.com","id":9}]
```

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Criar um user

Funcionalidade necessária para criar users.

/api/users/registro (**post**)

Dados:

```
{ "userName" : "João Pedro" , "password" : "password123" , "mail":  
"jp@gmail.com" }
```

Sucesso (200):

```
[ { "userName" : "João Pedro" , "password" : "password123" , "mail":  
"jp@gmail.com" } ]
```

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Pesquisar um user pelo id

Funcionalidade necessária para criar users.

/api/users/{use_id}

Parâmetros:

id: Os users mostrados têm de conter o texto neste parâmetro no id do user.

Sucesso (200):

```
[{"userName":"Gustavo Costa","password":"123","mail":"gcosta@gmail.com","id":1}]
```

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Pesquisar um user pelo nome

Funcionalidade necessária para criar users.

/api/users/{use_name}

Parâmetros:

nome: Os users mostrados têm de conter o texto neste parâmetro no nome do user.

Sucesso (200):

[{"userName":"Gustavo Costa","password":"123","mail":"gcosta@gmail.com","id":1}]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Pesquisar um user pelo email

Funcionalidade necessária para criar users.

/api/users/{use_mail}

Parâmetros:

mail: Os users mostrados têm de conter o texto neste parâmetro no mail do user.

Sucesso (200):

```
[{"userName":"Gustavo Costa","password":"123","mail":"gcosta@gmail.com","id":1}]
```

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Atualizar user pelo id

Funcionalidade necessária para atualizar o username e password do user pelo id.

/api/users/update/{use_id} (**put**)

Parâmetros:

id: Os users mostrados têm de conter o texto neste parâmetro no id do user.

Dados: username e password.

Sucesso (200):

["User updated"]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Eliminar user pelo id

Funcionalidade necessária para eliminar o user pelo id.

/api/users/delete/{use_id} (**delete**)

Parâmetros:

id(obrigatório): Para identificar o user que queremos eliminar precisamos do id.

Sucesso (200):

[{"userName":"Júlia Pinheiro","password":"123","mail":"juliapinheiro@gmail.com","id":9}]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Recurso Grupos (api/groups)

Listar todos os grupos

Devolve a lista de todos os grupos. Funcionalidade necessária para pesquisa de grupos.

/api/groups (**get**)

Sucesso (200):

```
[{"groupId":1,"creatorId":1,"groupName":"Grupo Ferrari","groupDescription":"Grupo que junta pessoas com interesse na equipa de f1 Ferrari.","groupAge":"2022-09-09","activity":true,"lastPost":"2022-09-10T08:30:25.000+00:00"}, {"groupId":2,"creatorId":1,"groupName":"Grupo Mercedes","groupDescription":"Grupo que junta pessoas com interesse na equipa de f1 Mercedes.","groupAge":"2022-09-09","activity":true,"lastPost":"2022-09-10T08:30:25.000+00:00"}, {"groupId":3,"creatorId":1,"groupName":"Grupo RedBull","groupDescription":"Grupo que junta pessoas com interesse na equipa de f1 RedBull.","groupAge":"2022-09-09","activity":true,"lastPost":"2022-09-10T08:30:25.000+00:00"}, {"groupId":4,"creatorId":1,"groupName":"Grupo McLaren ","groupDescription":"Grupo que junta pessoas com interesse na equipa de f1 McLaren.","groupAge":"2022-09-09","activity":true,"lastPost":"2022-09-10T08:30:25.000+00:00"} (...)]
```

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Criar um grupo

Funcionalidade necessária para criar grupos.

/api/groups/create (**post**)

Dados: {"creatorId":1,"groupName":"Grupo Ferrari","groupDescription":"Grupo que junta pessoas com interesse na equipa de f1 Ferrari.","groupAge":"2022-09-09","activity":true,"lastPost":"2022-09-10T08:30:25.000+00:00"}

Sucesso (200):
[{"Group created"}]

Erro:
500: Erro de Servidor

Alterar dados do grupo pelo id.

Funcionalidade necessária para modificar o nome e a descrição do grupo pelo id.

/api/groups/update/{gru_id} (**put**)

Parâmetros:

id: Para identificar o user que altera os dados, precisamos do id.

Sucesso (200): [{"groupId":1,"creatorId":1,"groupName":"Grupo Ferrari","groupDescription":"Grupo que junta pessoas com interesse na equipa de f1 Ferrari.","groupAge":"2022-09-09","activity":true,"lastPost":"2022-09-10T08:30:25.000+00:00"}]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Alterar a atividade do grupo

Altera a atividade do grupo dependendo da última mensagem mandada no grupo. (O grupo fica inativo quando a última mensagem tem mais de 3 meses.)

/api/groups/update/activity (**put**)

Parâmetros:

id: Para verificar a atividade do grupo precisamos do id do mesmo.

Sucesso (200):

[{true}]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Eliminar o grupo pelo id

Funcionalidade necessária para eliminar o grupo. (Apenas o moderador poderá eliminar o grupo).

/api/groups/delete/{gru_id} (**delete**)

Parâmetros:

id: Para verificar a atividade do grupo precisamos do id do mesmo.

Sucesso (200):

[{"Delete group with id 1"}]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Recurso Mensagem (/api/message)

Cria uma mensagem

Funcionalidade necessária para criar mensagens para comunicar nos grupos.

/api/message/create (**post**)

Dados: {"messageText": "Alguém quer almoçar para depois ir ver a corrida?"}

Sucesso (200): [{"messageId":1,"messageText":"Alguém quer almoçar para depois ir ver a corrida?","messageUserId":1,"messageUserGroupId":1,"messageDate":"2022-09-22T14:10:25"}]

Erro:

500: Erro de Servidor

Editar a mensagem

Funcionalidade necessária para editar a mensagem de users.

/api/message/edit/{mes_id} (**put**)

Parâmetros:

id: Para identificar a mensagem que queremos alterar, precisamos do id.

Sucesso (200):

[{"Message edited"}]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Eliminar a mensagem

Funcionalidade necessária para eliminar a mensagem.

/api/message/delete/{mes_id} (**delete**)

Parâmetros:

id: Para identificar a mensagem que queremos eliminar precisamos do id.

Sucesso (200):

[{"Delete message with id 1"}]

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

Recurso Tags (/api/tags)

Listar as tags

Funcionalidade necessária para mostrar todas as tags existentes.

/api/tags

Sucesso (200):

```
[{"tagId":1,"tagName":"Ferrari"}, {"tagId":2,"tagName":"Mercedes"}, {"tagId":3,"tagName":"RedBull"}, {"tagId":4,"tagName":"Mclaren"}, {"tagId":5,"tagName":"Alpine"}, {"tagId":6,"tagName":"Alfa Romeo"}, {"tagId":7,"tagName":"Aston Martin"}, {"tagId":8,"tagName":"Haas"}, {"tagId":9,"tagName":"Alphat auri"}, {"tagId":10,"tagName":"Williams"}]
```

Erro:

404: Not Found (O servidor não conseguiu encontrar o recurso solicitado)

500: Erro de Servidor

9-Bibliografia

Fórmula 1

app:https://play.google.com/store/apps/details?id=com.softpauer.f1timingapp2014.basic&hl=pt_PT&gl=US

SpringBoot:

<https://spring.io/projects/spring-boot>

VS Code:

<https://code.visualstudio.com/>

Figma (MockUps):

<https://www.figma.com/>

PgAdmin:

<https://www.pgadmin.org/download/>

AndroidStudio:

<https://developer.android.com/studio>

Gráfico de Gantt:

<https://www.officetimeline.com/online/gantt-chart-maker>

RetroFit2:

<https://square.github.io/retrofit/>