

## SIS

# Boleias

## Relatório de SIS

Nº 2231439

Ivan Santos

Nº 2231448

Tiago Bernardino

Nº 2231445

André Inácio

## CONTEXTUALIZAÇÃO

A aplicação trata-se de uma plataforma de boleias, permitindo que utilizadores possam tanto oferecer como reservar viagens partilhadas. O objetivo principal é facilitar a ligação entre condutores e passageiros, proporcionando uma experiência prática de reserva de boleias, gestão de favoritos e avaliação de serviços. A aplicação distingue-se pelo seu sistema de reembolso proporcional, que ajusta o valor pago pelo passageiro consoante o número de pessoas que reservam a mesma boleia, incentivando a partilha de viagens e promovendo a eficiência do transporte. As operações CRUD que irão ser necessárias para android são a autenticação, as viaturas, as boleias, as reservas, a wishlist e a avaliação do condutor.

### Autenticação

#### Descrição:

Faz autenticação para obter o token, junto com as informações do perfil associado a esse user.

- Endpoint: `/api/auth`
- Método HTTP: `POST`

#### Exemplo de chamada com cURL

```
C:\Users\PC>curl -X POST http://localhost/PROJETOS/boleias/web/PSI-WEB/boleias/backend/web/api/auth ^
More? -H "Content-Type: application/json" ^
More? -d '{"username":"condutor","password":"123456789"}'
{"success":true,"token":"XVsne3mhfe4nCRI_2gJskXGmKi8Sd_I1","user_id":2,"perfil_id":10,"nomePerfil":"Ivan Fernandes dos Santos","condutor":1}
```

### Viaturas

#### Descrição:

POST das viaturas mandando o token para obter a autorização..

- Endpoint: `/api/viatura`
- Método HTTP: `POST`

#### Exemplo de chamada com cURL

```
C:\Users\PC>curl -X POST http://localhost/PROJETOS/boleias/web/PSI-WEB/boleias/backend/web/api/viatura ^
More? -H "Content-Type: application/json" ^
More? -H "Authorization: Bearer XVsne3mhfe4nCRI_2gJskXGmKi8Sd_I1" ^
More? -d '{"marca":"Toyota","modelo":"Corolla","matricula":"AA-11-BB","cor":"Preto","perfil_id":10}'
{"success":true,"message":"Viatura criada com sucesso","data":{"marca":"Toyota","modelo":"Corolla","matricula":"AA-11-BB","cor":"Preto","perfil_id":10,"id":18}}
```

**Descrição:**

GET das viaturas mandando o token para obter a autorização..

- Endpoint: `/api/viatura`
- Método HTTP: `GET`

**Exemplo de chamada com cURL**

```
C:\Users\PC>curl -X GET "http://localhost/PROJETOS/boleias/web/PSI-WEB/boleias/backend/web/api/viatura?perfil_id=10" ^
More? -H "Authorization: Bearer XVsne3mhfe4nCRI_2gJskXGmKi8Sd_I1"
{"success":true,"data":[{"id":18,"marca":"Toyota","modelo":"Corolla","matricula":"AA-11-BB","cor":"Preto","perfil_id":10}]}
```

**Descrição:**

PUT das viaturas mandando o token para obter a autorização..

- Endpoint: `/api/viatura`
- Método HTTP: `PUT`

**Exemplo de chamada com cURL**

```
C:\Users\PC>curl -X PUT http://localhost/PROJETOS/boleias/web/PSI-WEB/boleias/backend/web/api/viatura/18 ^
More? -H "Content-Type: application/json" ^
More? -H "Authorization: Bearer XVsne3mhfe4nCRI_2gJskXGmKi8Sd_I1" ^
More? -d "{\"marca\":\"Toyota\",\"modelo\":\"Corolla\",\"matricula\":\"AA-11-BB\",\"cor\":\"Azul\",\"perfil_id\":10}"
{"success":true,"message":"Viatura atualizada com sucesso","data":{"id":18,"marca":"Toyota","modelo":"Corolla","matricula":"AA-11-BB","cor":"Azul","perfil_id":10}}
```

**Descrição:**

DEL das viaturas mandando o token para obter a autorização..

- **Endpoint:** `/api/viatura`
- **Método HTTP:** `DEL`

**Exemplo de chamada com cURL**

```
C:\Users\PC>curl -X DELETE http://localhost/PROJETOS/boleias/web/PSI-WEB/boleias/backend/web/api/viatura/18 ^
More? -H "Authorization: Bearer XVsne3mhfe4nCRI_2gJskXGmKi8Sd_I1"
{"success":true,"message":"Viatura eliminada com sucesso"}
```

## MESSAGING

### Descrição:

A cada vez que será feita uma avaliação de um condutor, será notificado.

```
public function afterSave($insert, $changedAttributes)
{
    parent::afterSave($insert, $changedAttributes);

    $obj = new \stdClass();
    $obj->id = $this->id;
    $obj->descricao = $this->descricao;

    $json = json_encode($obj);

    $this->FazPublishNoMosquitto("AVALIACAO_FEITA", $json);
}
```

**Descrição:**

A cada vez que uma boleia será criada, apagada ou atualizada, será notificado.

```
public function afterSave($insert, $changedAttributes)
{
    parent::afterSave($insert, $changedAttributes);

    $obj = new \stdClass();
    $obj->id = $this->id;
    $obj->origem = $this->origem;
    $obj->destino = $this->destino;
    $obj->data_hora = $this->data_hora;
    $obj->lugares_disponiveis = $this->lugares_disponiveis;
    $obj->preco = $this->preco;
    $obj->viatura_id = $this->viatura_id;

    $json = json_encode($obj);

    if ($insert)
    {
        $this->FazPublishNoMosquitto("BOLEIA_INSERT", $json);
    }
    else
    {
        $this->FazPublishNoMosquitto("BOLEIA_UPDATE", $json);
    }
}
```

```
public function afterDelete()
{
    parent::afterDelete();

    $obj = new \stdClass();
    $obj->id = $this->id;

    $json = json_encode($obj);
    $this->FazPublishNoMosquitto("BOLEIA_DELETE", $json);
}
```

**Descrição:**

A cada vez que será feita uma validação de documentos de um condutor, será notificado.

```
public function afterSave($insert, $changedAttributes)
{
    parent::afterSave($insert, $changedAttributes);

    $obj = new \stdClass();
    $obj->id = $this->id;
    $obj->valido = $this->valido;

    $json = json_encode($obj);

    $this->FazPublishNoMosquitto("DOCUMENTOS_VALIDADOS", $json);
}
```



## SERVIDOR

IP -> 172.22.21.244

Credenciais -> username: grupo-pl-f // password: okvA!@#y4xav