

Disciplina	Programação Distribuída	Código	CP406TIN3
Nome	Marcelo Zaguet Junior	RA	180998
Nome	Bruno Cesar de Moraes	RA	180344
Nome	Felipe Siqueira Godoy	RA	190337
Nome	Guilherme Machado Crespo	RA	190501
Nome	Henrique César Pires Soares Almeida	RA	190804

Introdução

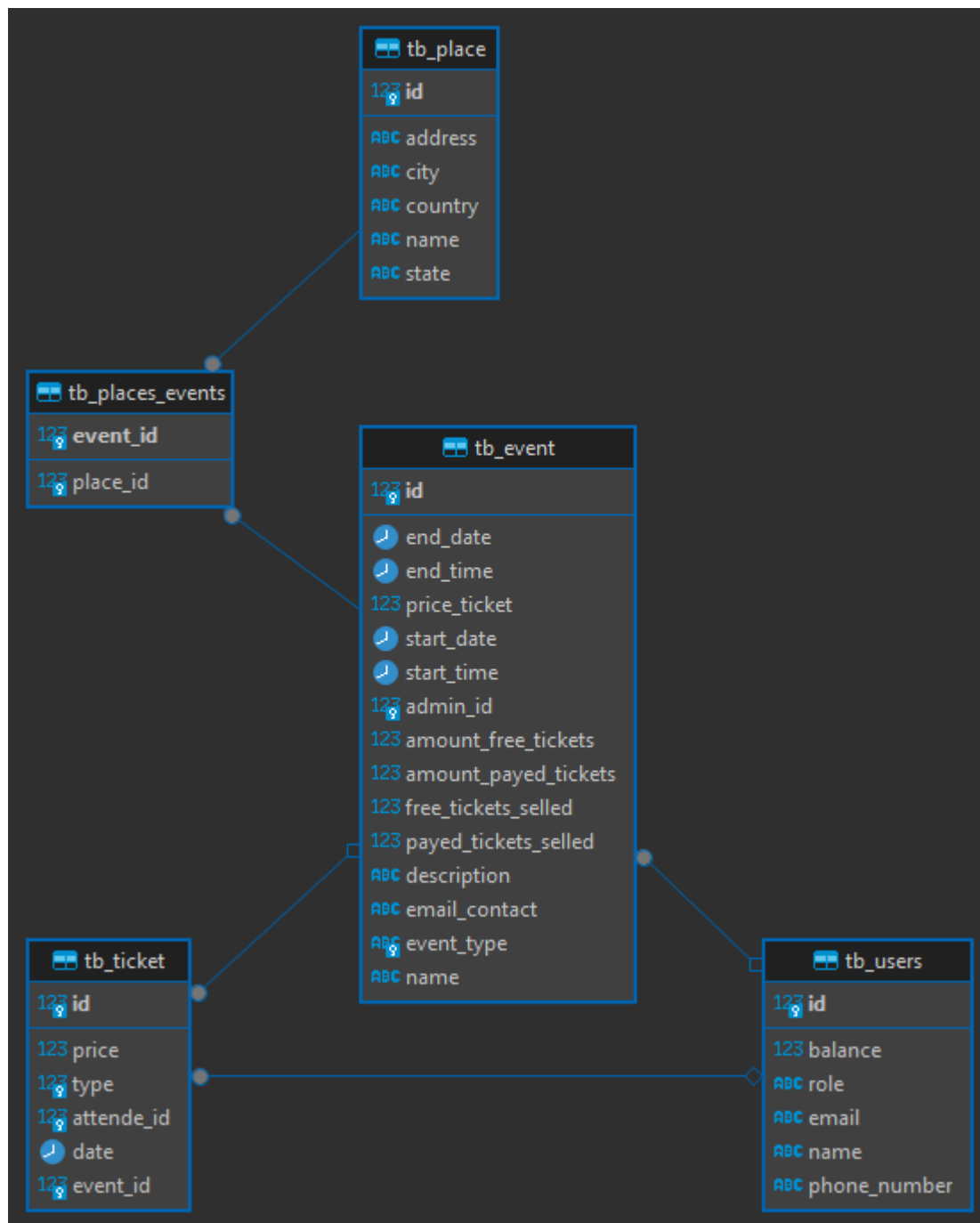
As APIs (Application Programming Interfaces) são construções disponíveis nas linguagens de programação que permitem aos desenvolvedores criar funcionalidades complexas mais facilmente. Tais construções abstraem o código mais complexo, proporcionando o uso de sintaxes mais simples em seu lugar.

Referência:

https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/Client-side_web_APIs/Introduction

Apresentação

De acordo com a organização Mozilla, Para esse projeto foi escolhido desenvolver um WebApp para Eventos podendo cadastrar endereços, eventos, adicionar eventos a endereços. Além disso, é possível comprar os ingressos além de ter categoria gratuita e paga para os ingressos. Existe também uma forma de buscar todos os eventos de uma cidade. Todas as 26 capitais e o Distrito Federal do Brasil possuem três eventos.



Desenvolvimento

Rotas

Há as seguintes rotas:

Events

- Get: `/events`

Busca todos os eventos, você pode passar os Query Params como page e limit para fazer paginação e seus detalhes são retornados nos Headers, e Query Params como name, description, emailContact ou startDate para filtrar os resultados ou fazer uma pesquisa.

Exemplo: /events?page=0&limit=1&name=Marcelo

Exemplo: /events?page=0&limit=1&startDate=2021-03-20

Exemplo: /events?page=0&limit=1&name=Marcelo

- Post: /events

Cria um novo evento.

```
{
  "name": "Evento do Joga18",
  "eventType": "MOVIE_THEATER",
  "description": "Será um evento legal",
  "place": "Facens",
  "emailContact": "180998@facens.br",
  "startDate": "2021-03-21",
  "endDate": "2021-03-28",
  "startTime": "16:30:00",
  "endTime": "19:00:00",
  "amountFreeTickets": 10,
  "amountPayedTickets": 90,
  "priceTicket": 60.99,
  "adminId": 1
}
```

- Get: /events/{eventId}

Busca um evento pelo seu ID.

- Put: /events/{eventId}

Atualiza um evento pelo seu ID.

```
{
  "name": "Evento do Joga11",
  "eventType": "THEATER",
  "description": "Será um evento legal para caramba.",
  "place": "Facens - LIGA",
  "emailContact": "18099823@etec.sp.gov.br",
  "startDate": "2021-03-20",
  "endDate": "2021-03-22",
  "startTime": "13:00:00",
  "endTime": "15:00:00",
  "amountFreeTickets": 5,
  "amountPayedTickets": 95,
  "priceTicket": 99.99,
}
```

```
"adminId": 2
}
```

- Delete: /events/{eventId}
- Remove um evento pelo seu ID.

Admins

- POST: /admins

Cria um admin.

```
{
  "name": "Marcelo Zaguette",
  "email": "180998@facens.br",
  "phoneNumber": "15988116120"
}
```

- PUT: /admins/{adminId}

Edita um admin.

```
{
  "name": "Marcelo Zaguette 2",
  "email": "1809982@facens.br",
  "phoneNumber": "15988116120"
}
```

- GET: /admins/{adminId}

Busca um admin.

- GET: /admins

Busca vários admins.

- DELETE: /admins/{adminId}

Remove um admin.

Attendees

- POST: /attendees

Cria um participante.

```
{
  "name": "Marcelo Zaguette",
  "email": "180998@facens.br"
}
```

PUT: /attendees/{attendeId}

Atualiza um participante.

```
{
  "name": "Marcelo Zaguette 2",
```

```
"email": "1809982@facens.br"
}
```

- GET: /attendees/{attendeeld}

Busca um participante.

- GET: /attendees

Busca vários participantes.

- DELETE: /attendees/{attendeeld}

Remove um participante.

```
{
  "name": "Marcelo Zaguetta 2",
  "email": "180998_2@facens.br"
}
```

Places

- POST: /places

Cria um lugar.

```
{
  "name": "Casa do Joga10",
  "address": "Rua Monsenhor Benedito Mário Calazans",
  "city": "Curitiba",
  "state": "PR",
  "country": "Brasil"
}
```

- PUT: /places/{placeld}

Atualiza um lugar.

```
{
  "name": "Casa do Joga10 2",
  "address": "Rua Monsenhor Benedito Mário Calazans 2",
  "city": "Curitiba",
  "state": "PR",
  "country": "Brasil"
}
```

- GET: /places/{placeld}

Busca um lugar.

- GET: /places

Busca vários lugares.

- DELETE: /places/{placeId}

Remove um lugar.

Events-places/

- GET: /events-places/city/{city}

Busca eventos em uma determinada cidade.

- POST: /events-places/{eventId}/{placeId}

Adiciona um lugar a um evento.

- DELETE: /events-places/{eventId}/{placeId}

Remove um lugar a um evento.

events/{eventId}/tickets

- GET: /events/{eventId}/tickets

Busca os Tickets de um evento.

- POST: /events/{eventId}/tickets

Vende um ticket para um attendee.

Separação feita entre tickets pagos e gratuitos através do campo "type", onde 0 é gratuito e 1 é pago.

```
{  
  "attendId": 5,  
  "type": 1  
}
```

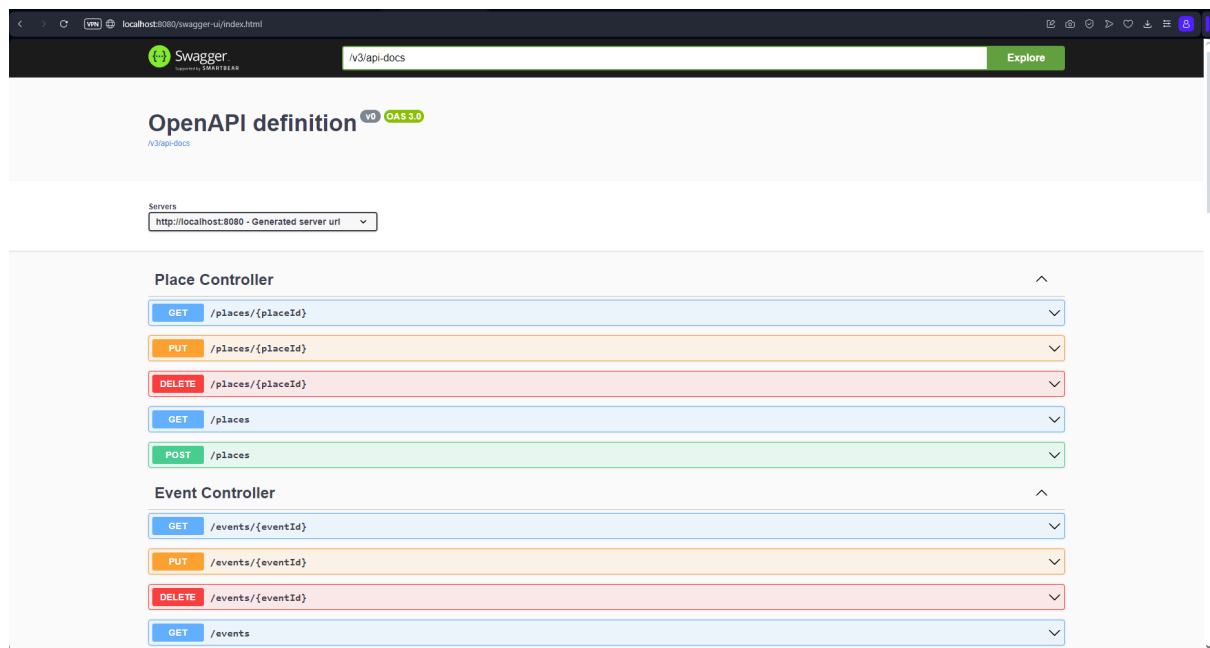
DEL: /events/{eventId}/tickets

Remove ticket vendido.

Separação feita entre tickets pagos e gratuitos através do campo "type", onde 0 é gratuito e 1 é pago.

```
{  
  "attendId": 5,  
  "type": 0  
}
```

Também está configurado o Swagger para auxiliar na utilização da WebApi



Para publicar o serviço basta executar o comando:

`mvn spring-boot:run`

Nota: É necessário possuir Java e também o Maven cadastrado no computador para conseguir subir o serviço na porta 8080 (<http://localhost:8080>)