Event Management System

Desenvolvimento WEB I – **M3**

Event Managment System - Melhorado

Para A nossa parte M3 decidimos Continuar o Trabalho na nossa API (Application Programming Interface) desenvolvida em M2 , (Sistema De Gestão de Eventos)

Esta passa a ser criada com recurso ao loopback 4 e apos isso a criaçao de uma interface mais intuitiva com recurso a ReactAdmin

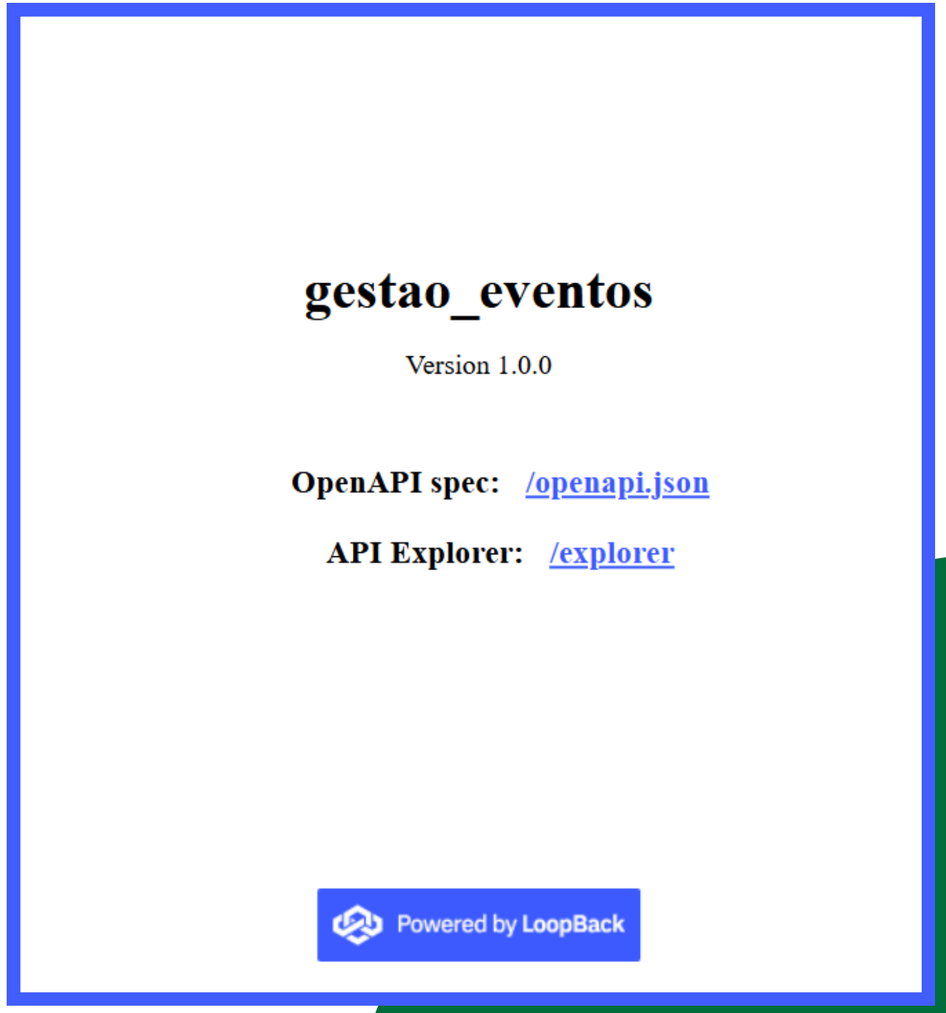
A API mantem os Quatro principais recursos:
(Organizadores, Eventos, Bilhetes e Participantes.)
Porem para alem dos Quatro Metodos Utilizados Anteriormente <GET>, <POST>, <PUT> e <DELETE>, esta Passa a Utilizar Tambem <PATCH>

Para Alem Apresenta ainda melhorias nas relaçoes o que melhora as funcionalidades de filtragem
Exemplo filtragem de um Ticket especifico de um Evento

Loopback4

Como dito Anteriormente desta vez de modo a melhorar-mos a nossa API utilizamos a tecnologia **LoopBack 4**, que simplificou significativamente todo o processo de desenvolvimento.

Com essa abordagem, conseguimos facilitar a criação de tabelas, o estabelecimento de relações entre elas e a conexão com o banco de dados MySQL, tornando o desenvolvimento mais ágil e eficiente.



Processo - Loopback

LoopBack

LB4 MODEL

(Criaçao de Modelo de Dados)

```
PS C:\Users\Utilizador\Desktop\M3_Exempo\m-3-exempo> lb4 model
(node:26340) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
? Model class name: Event
? Please select the model base class Entity (A persisted model with an ID)
? Allow additional (free-form) properties? No
Model Event will be created in src/models/event.model.ts

Let's add a property to Event
Enter an empty property name when done

? Enter the property name: id
? Property type: number
? Is id the ID property? Yes
? Is id generated automatically? Yes

Let's add another property to Event
Enter an empty property name when done

? Enter the property name: title
? Property type: string
? Is it required?: Yes

Let's add another property to Event
Enter an empty property name when done

? Enter the property name: description
? Property type: string
? Is it required?: Yes
```

Processo - Loopback

LB4 DATASOURCE

(Criaçao / Ligaçao Base de Dados)

```
PS C:\Users\Utilizador\Desktop\M3_Exempo\m-3-exempo> lb4 datasource
(node:10028) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
? Datasource name: sistema_gestao
? Select the connector for sistema_gestao:  MySQL (supported by StrongLoop)
? Connection String url to override other settings (eg: mysql://user:pass@host/db):
? host: localhost
? port: 3306
? user: root
? password: [hidden]
? database: sistema_gestao
   create src\datasources\sistema-gestao.datasource.ts
```

Processo - Loopback

LB4 REPOSITORY

(Gerar os Repositorios)

```
PS C:\Users\Utilizador\Desktop\M3_Exempo\m-3-exempo> lb4 repository
(node:11208) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
? Select the datasource SistemaGestaoDatasource
? Select the model(s) you want to generate a repository for Event
   create src\repositories\event.repository.ts

No change to package.json was detected. No package manager install will be executed.
   update src\repositories\index.ts

Repository EventRepository was/were created in src\repositories
```

Processo - Loopback

LB4 CONTROLLER

(Criaçaao de Controllers REST)

```
PS C:\Users\Utilizador\Desktop\M3_Exempo\m-3-exempo> lb4 controller
(node:21600) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
? Controller class name: Event
Controller Event will be created in src/controllers/event.controller.ts

? What kind of controller would you like to generate? REST Controller with CRUD functions
? What is the name of the model to use with this CRUD repository? Event
? What is the name of your CRUD repository? EventRepository
? What is the name of ID property? id
? What is the type of your ID? number
? Is the id omitted when creating a new instance? Yes
? What is the base HTTP path name of the CRUD operations? /events
   create src\controllers\event.controller.ts

No change to package.json was detected. No package manager install will be executed.
   update src\controllers\index.ts

Controller Event was/were created in src\controllers
```

Processo - Loopback

LB4 RELATION

(Relaçoes Entre modelos de Dados)

```
PS C:\Users\Utilizador\Desktop\M3_Exempo\m-3-exempo> lb4 relation
(node:26700) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
? Please select the relation type hasMany
? Please select source model Event
? Please select target model Event
? Foreign key name to define on the target model eventId
? Source property name for the relation getter (will be the relation name) events
? Allow Event queries to include data from related Event instances?  Yes
   create src\controllers\event-event.controller.ts

No change to package.json was detected. No package manager install will be executed.

Relation HasMany was/were created in src
```

React Adimn

Após a melhoria da nossa API com o uso do LoopBack 4, avançamos para a otimização da sua apresentação e interface, utilizando o **React Admin.**

Com essa implementação, foram adicionadas diversas funcionalidades para tornar a interface mais intuitiva e amigável para o usuário, proporcionando uma experiência de utilização mais fluida e eficiente.

Processo - React Admin

APP.JS

(Ficheiro Principal App)

App.js

App.js > default

```
1  import { Admin, Resource, ShowGuesser } from "react-admin";
2  import lb4Provider from "react-admin-lb4";
3  import { OrganizerCreate, OrganizerEdit, OrganizerList } from "./OrganizerList";
4  import { EventList, EventEdit, EventCreate } from "./EventList";
5  import { TicketList, TicketEdit, TicketCreate } from "./TicketList";
6  import { PartricipantList, PartricipantEdit, PartricipantCreate } from "./PartricipantList";
7
8  // Icons
9  import ConfirmationNumberIcon from "@mui/icons-material/ConfirmationNumber";
10 import UserIcon from "@mui/icons-material/Group";
11 import EventIcon from "@mui/icons-material/Event";
12 import WorkIcon from "@mui/icons-material/Work";
13
14 import Dashboard from "./Dashboard/Dashboard";
15 import { authProvider } from "./authProvider";
16 import "bootstrap/dist/css/bootstrap.min.css";
17
18 const dataProvider = lb4Provider("http://127.0.0.1:3000/");
19 const App = () => (
20   <Admin authProvider={authProvider} dataProvider={dataProvider} dashboard={Dashboard}>
21     <Resource name="organizers" list={OrganizerList} show={ShowGuesser} edit={OrganizerEdit} create={OrganizerCreate} icon={WorkIcon} />
22     <Resource name="events" list={EventList} show={ShowGuesser} edit={EventEdit} create={EventCreate} icon={EventIcon} />
23     <Resource name="tickets" list={TicketList} show={ShowGuesser} edit={TicketEdit} create={TicketCreate} icon={ConfirmationNumberIcon} />
24     <Resource name="partricipants" list={PartricipantList} show={ShowGuesser} edit={PartricipantEdit} create={PartricipantCreate} icon={UserIcon} />
25   </Admin>
26 );
27
28 export default App;
```

Processo - React Admin

EventList

(Dados que serão Apresntados)

```
export const EventList = () => (
    <List filters={eventFilters}>
        <Datagrid>
            <TextField source="id" />
            <TextField source="title" />
            <TextField source="description" />
            <DateField source="date" />
            <TextField source="time" />
            <TextField source="location" />
            <NumberField source="organizerId" label="Organizer ID" />
            <ReferenceField source="organizerId" reference="organizers" link="show" />
        </Datagrid>
    </List>
);
```

ShowGuesser

(Abetura Quando Clicado)

```
import { Admin, Resource, ShowGuesser } from "react-admin";
```

(Funcionalidade Padrão do React-Admin)

Processo - React Admin

EventEdit

(Ediçao de Eventos)

```
export const EventEdit = () => (
    <Edit>
        <SimpleForm>
            <TextInput source="id" disabled />
            <TextInput source="title" />
            <TextInput source="description" multiline rows={5} />
            <DateInput source="date" />
            <TextInput source="time" />
            <TextInput source="location" />
            <NumberInput source="organizerId" label="Organizer ID" />
            <ReferenceInput source="organizerId" reference="organizers">
                <SelectInput optionText="name" />
            </ReferenceInput>
        </SimpleForm>
    </Edit>
);
```

EventCreate

(Criaçao de Eventos)

```
export const EventCreate = () => (
    <Create>
        <SimpleForm>
            <TextInput source="title" />
            <TextInput source="description" multiline rows={5} />
            <DateInput source="date" />
            <TextInput source="time" />
            <TextInput source="location" />
            <NumberInput source="organizerId" label="Organizer ID" />
            <ReferenceInput source="organizerId" reference="organizers">
                <SelectInput optionText="name" />
            </ReferenceInput>
        </SimpleForm>
    </Create>
);
```

Processo - React Admin

EventFilters

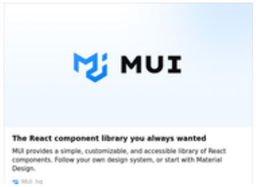
(Filtros em Eventos)

```
const eventFilters = [
    <TextInput source="q" label="Search" alwaysOn />,
    <ReferenceInput source="organizerId" label="Organizer" reference="organizers" perPage={100} />,
];
```

EventIcon

(Icon de Eventos)

```
import EventIcon from "@mui/icons-material/Event";
```



Processo - React Admin

DashBoard.JS

(Aba de Dashboard)

Graficos

```
const fetchChartData = async () => {
    try {
        const promises = resources.map((resource) =>
            dataProvider.getList(resource.name, { pagination: { page: 1, perPage: 1 }, sort: { field: "id", order: "ASC" } })
        );

        const results = await Promise.all(promises);

        const pieChartData = results.map((result, index) => ({
            name: resources[index].title,
            value: result.total,
        }));
        setPieData(pieChartData);

        const ticketsResponse = await dataProvider.getList("tickets", {
            pagination: { page: 1, perPage: 100 },
            sort: { field: "price", order: "ASC" },
        });

        const ticketPrices = ticketsResponse.data.map((ticket) => ticket.price);
        const barChartData = [
            { priceRange: "0-10€", tickets: ticketPrices.filter((price) => price <= 10).length },
            { priceRange: "11-20€", tickets: ticketPrices.filter((price) => price > 10 && price <= 20).length },
            { priceRange: "21-30€", tickets: ticketPrices.filter((price) => price > 20 && price <= 30).length },
            { priceRange: "31-50€", tickets: ticketPrices.filter((price) => price > 30 && price <= 50).length },
        ];
        setBarData(barChartData);
    } catch (error) {
        console.error("Error fetching chart data:", error);
    }
};

fetchChartData();
```

Botões

```
export const Dashboard = () => {
    const [sections, setSections] = useState([]);

    useEffect(() => {
        const fetchSections = async () => {
            const data = [
                {
                    title: "Organizers",
                    description: "Manage event organizers and their roles.",
                    buttonText: "View Organizers",
                    link: "/organizers",
                },
                {
                    title: "Events",
                    description: "View and manage all events.",
                    buttonText: "View Events",
                    link: "/events",
                },
```

Processo - React Admin

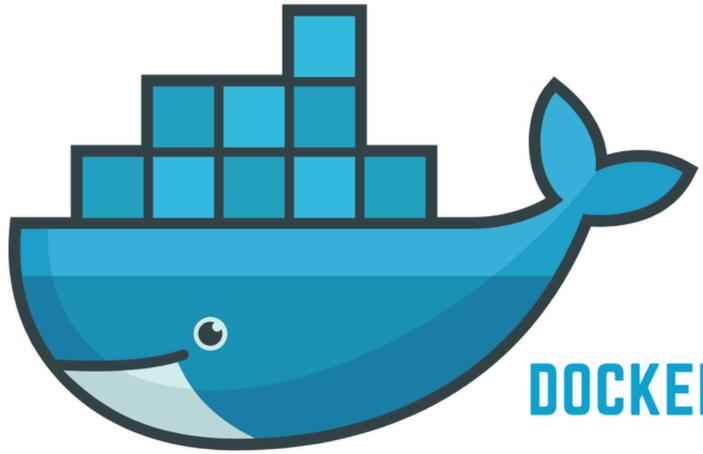
AuthProvider

(Login Inicio de Sessão)

```
export const authProvider = {
    // called when the user attempts to log in
    async login({ username, password }) {
        // accept all username/password combinations
        if (false) {
            throw new Error("Invalid credentials, please try again");
        }
        localStorage.setItem("username", username);
    },
    // called when the user clicks on the logout button
    async logout() {
        localStorage.removeItem("username");
    },
    // called when the API returns an error
    async checkError({ status }) {
        if (status === 401 || status === 403) {
            localStorage.removeItem("username");
            throw new Error("Session expired");
        }
    },
    // called when the user navigates to a new location, to check for authentication
    async checkAuth() {
        if (!localStorage.getItem("username")) {
            throw new Error("Authentication required");
        }
    },
```

DOCKER

Após a melhoria da nossa API com o uso do LoopBack 4 e a otimização da sua interface com o React Admin, avançamos para a próxima etapa: a containerização da aplicação utilizando o Docker. Com essa implementação, a aplicação ganhou mais portabilidade e escalabilidade. Agora, todos os seus serviços podem ser empacotados em contêineres isolados, garantindo maior consistência entre os ambientes de desenvolvimento, teste e produção. Essa abordagem simplifica o processo de implantação e manutenção, além de permitir uma gestão mais eficiente dos recursos utilizados pela aplicação.



DOCKERHUB

Processo - Docker

DOCKER-COMPOSE

(Conecta, organiza, gerencia contêineres.)

```yaml
version: '3.8'

Run All Services
services:
  Run Service
  mysql:
    image: mysql:8.0
    container_name: data
    environment:
      MYSQL_ROOT_PASSWORD: 12345678
    ports:
      - "3307:3306"
    volumes:
      - sql_data:/var/lib/mysql
      - ./SQL/init.sql:/docker-entrypoint-initdb.d/init.sql:ro
    healthcheck:
      test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
      interval: 10s
      timeout: 5s
      retries: 3
    networks:
      - app_network

  Run Service
  loopback:
    build:
      context: ./gestao_eventos
      dockerfile: Dockerfile
    container_name: loopback
    depends_on:
      mysql:
        condition: service_healthy
    ports:
      - "3000:3000"
    networks:
      - app_network


  Run Service
  node:
    image: node:18
    container_name: app
    working_dir: /usr/src/app
    volumes:
      - ./gestao_eventos_reactadmin:/usr/src/app
    command: "sh -c 'npm install && npm start'"
    ports:
      - "3001:3001"
    depends_on:
      mysql:
        condition: service_healthy
      loopback:
        condition: service_started
    networks:
      - app_network

networks:
  app_network:
    driver: bridge
volumes:
  sql_data:
```

Processo - Docker

DOCKERFILE

(Define, instala, configura ambiente Node.)

```
# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
# where available (npm@5+)
COPY --chown=node package*.json ./

RUN npm install

# Bundle app source code
COPY --chown=node . .

RUN npm run build

# Bind to all network interfaces so that it can be mapped to the host OS
ENV HOST=0.0.0.0 PORT=3000

EXPOSE ${PORT}
CMD [ "node", "." ]
```

Processo - Docker

DOCKER-COMPOSE UP -D --BUILD

(Comando de criação da docker.)

```
PS C:\Users\beatr\Desktop\Bia-m3-dw\M3 - C├┤pia> docker-compose up -d --build
time="2025-01-27T20:33:11Z" level=warning msg="C:\\Users\\beatr\\Desktop\\Bia-m3-dw\\M3 - C├┤pia\\docker-compo
ial confusion"
[+] Building 1.5s (13/13) FINISHED                                                    docker:desktop-linux
 => [loopback internal] load build definition from Dockerfile                                      0.1s
 => => transferring dockerfile: 684B                                                               0.0s
 => [loopback internal] load metadata for docker.io/library/node:18-slim                           1.1s
 => [loopback internal] load .dockerignore                                                         0.0s
 => => transferring context: 134B                                                                  0.0s
 => [loopback 1/7] FROM docker.io/library/node:18-slim@sha256:4cf853332a0358c135a3603745ec626c747c8e  0.0s
 => [loopback internal] load build context                                                         0.1s
 => => transferring context: 2.71kB                                                                0.0s
 => CACHED [loopback 2/7] RUN mkdir -p /home/node/app                                              0.0s
 => CACHED [loopback 3/7] WORKDIR /home/node/app                                                   0.0s
 => CACHED [loopback 4/7] COPY --chown=node package*.json ./                                       0.0s
 => CACHED [loopback 5/7] RUN npm install                                                          0.0s
 => CACHED [loopback 6/7] COPY --chown=node . .                                                    0.0s
 => CACHED [loopback 7/7] RUN npm run build                                                        0.0s
 => [loopback] exporting to image                                                                  0.1s
 => => exporting layers                                                                            0.0s
 => => writing image sha256:a5e0088ef862d75596d2578d892e629a0e9de0f46b8ebdde01ac967913da8f19       0.0s
 => => naming to docker.io/library/m3-cpia-loopback                                                0.0s
 => [loopback] resolving provenance for metadata file                                              0.0s
[+] Running 3/3
 ✓ Container data      Healthy                                                                    11.6s
 ✓ Container loopback  Started                                                                    11.0s
 ✓ Container app       Started                                                                    12.0s
PS C:\Users\beatr\Desktop\Bia-m3-dw\M3 - C├┤pia>
```

Processo - Docker

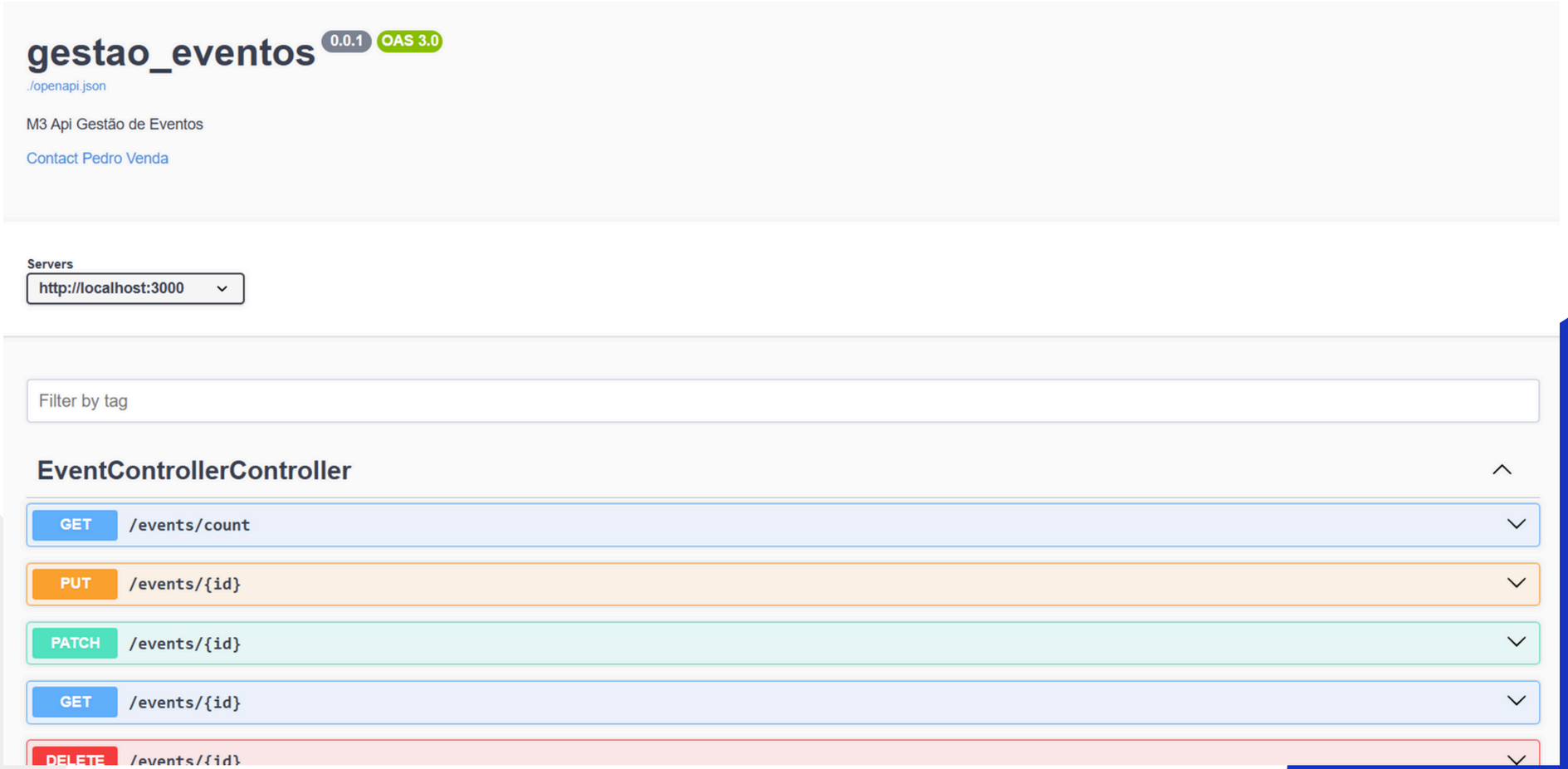
CONTAINER CRIADO NA DOCKER.

		Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
☐	⌄ ●	m3-cpia	-	-	-	0.48%	3 minutes ago	■ ⋮ 🗑
☐	●	app	96274d48001a	node:18	3001:3001	0.05%	3 minutes ago	■ ⋮ 🗑
☐	●	loopback	551ffc0abd4b	m3-cpia-loopback	3000:3000	0%	3 minutes ago	■ ⋮ 🗑
☐	●	data	e0ab88a6f3b9	mysql:8.0	3307:3306	0.43%	3 minutes ago	■ ⋮ 🗑

Processo - Docker

LOOPBACK

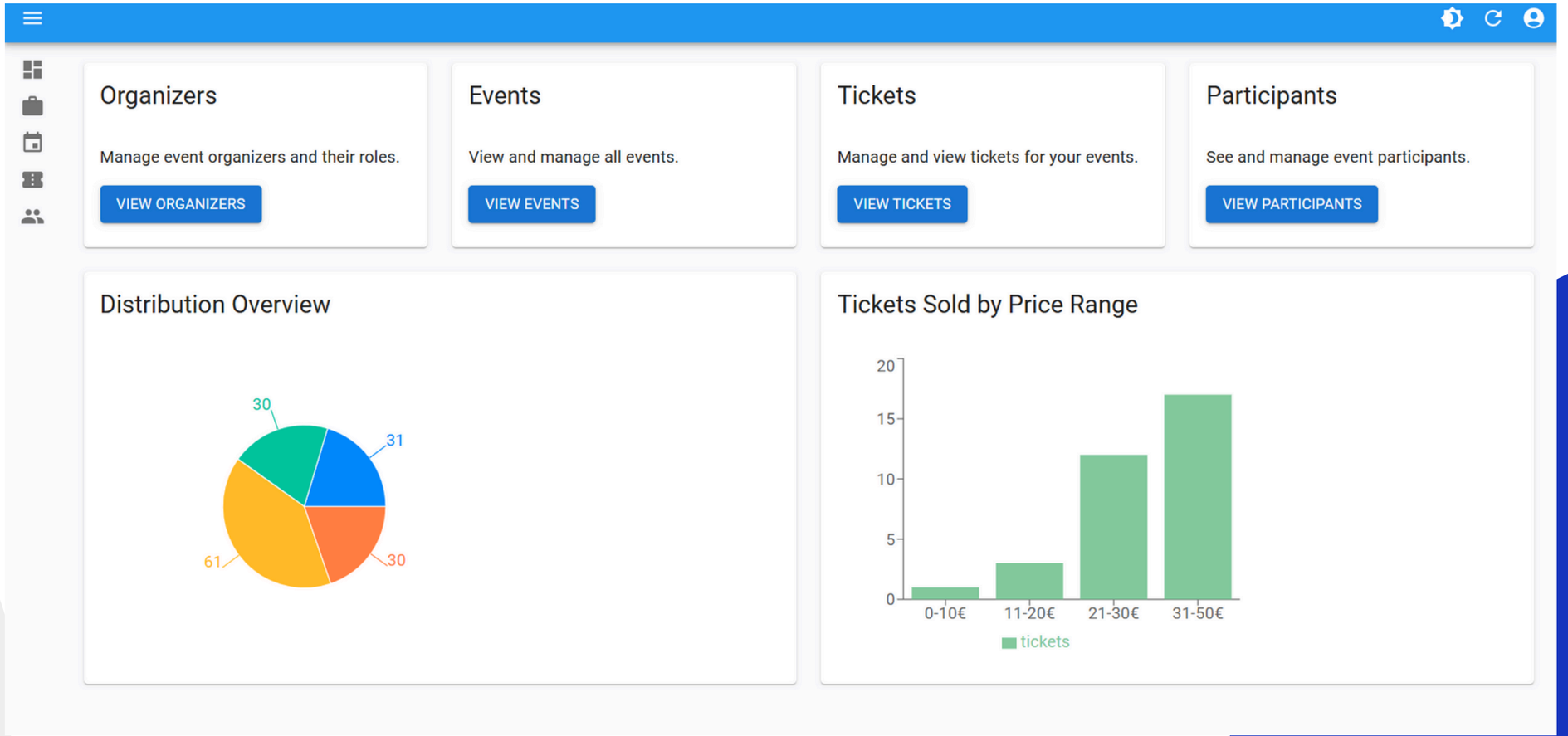
(http://localhost:3000/explorer/)



Processo - Docker

REACT APP

(http://localhost:3001/)



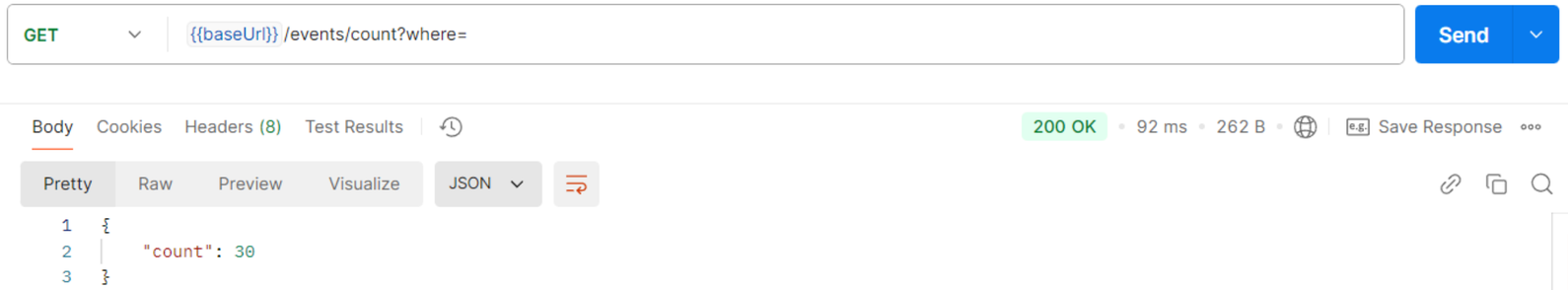
POSTMAN

Após concluida a API com as Ferramentas Anteriores por uma questao de revalidaçao do nosso trabalho inserimos este tambem no **Postman.**

Processo - Postman

COUNT

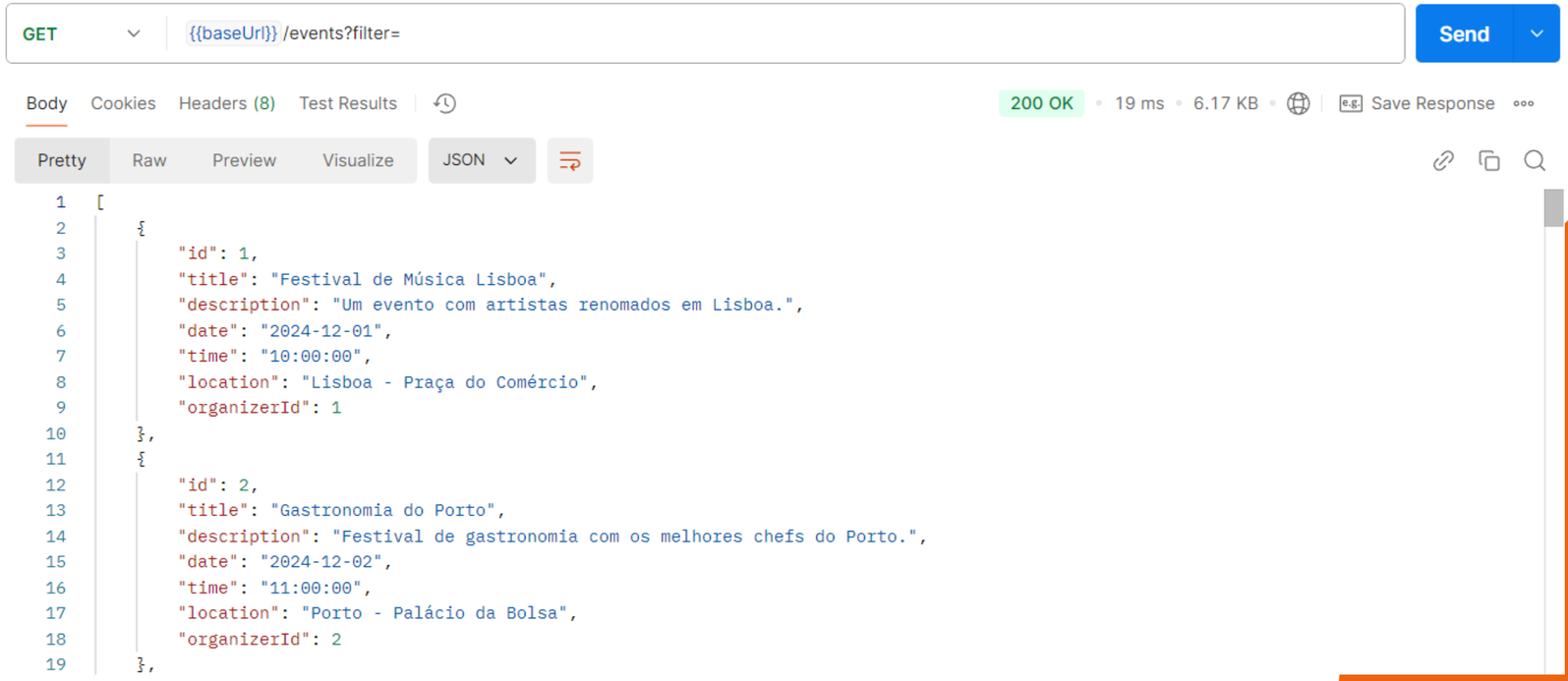
(Conta Numero Eventos Existentes)



Processo - Postman

GET EVENTS

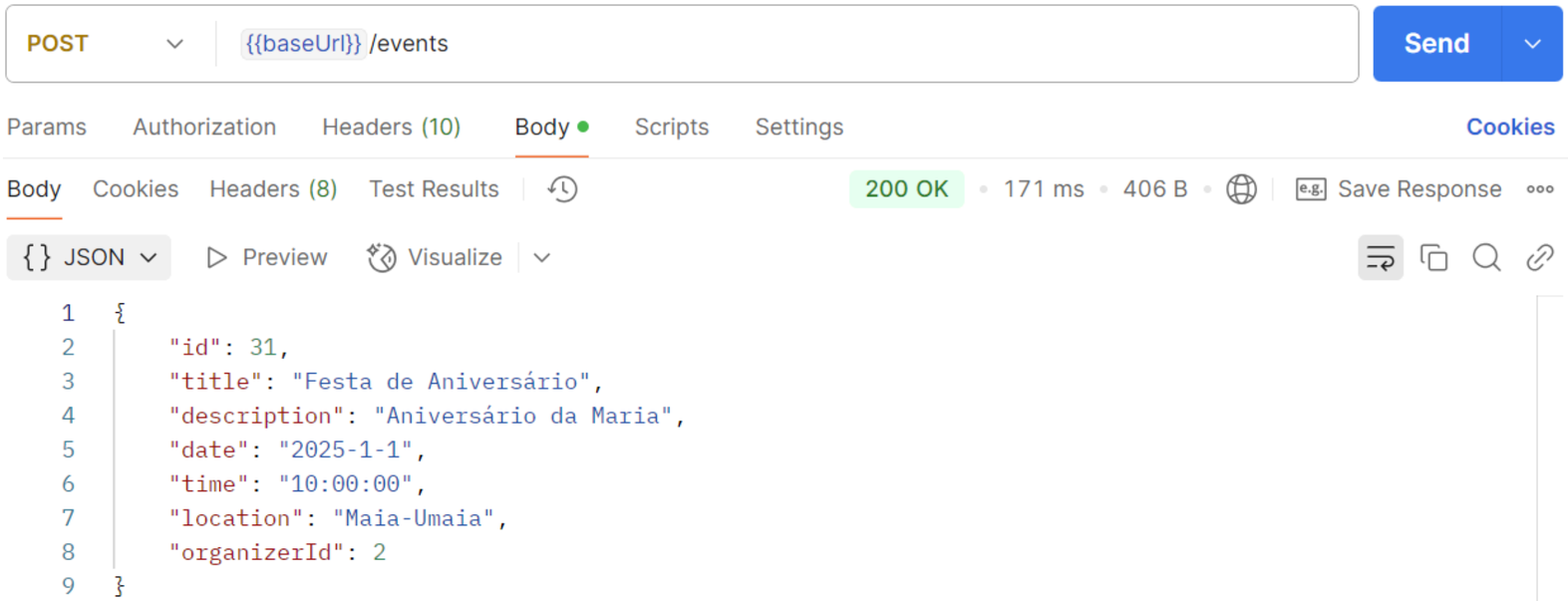
(Retorna Todos Os Eventos)



Processo - Postman

POST EVENT

(Cria Um Eventos)



Processo - Postman

PUT EVENT{ID}

(Atualiza Um Evento)

PUT {{baseUrl}} /events/31 Send

Params Authorization Headers (10) Body Scripts Settings Cookies

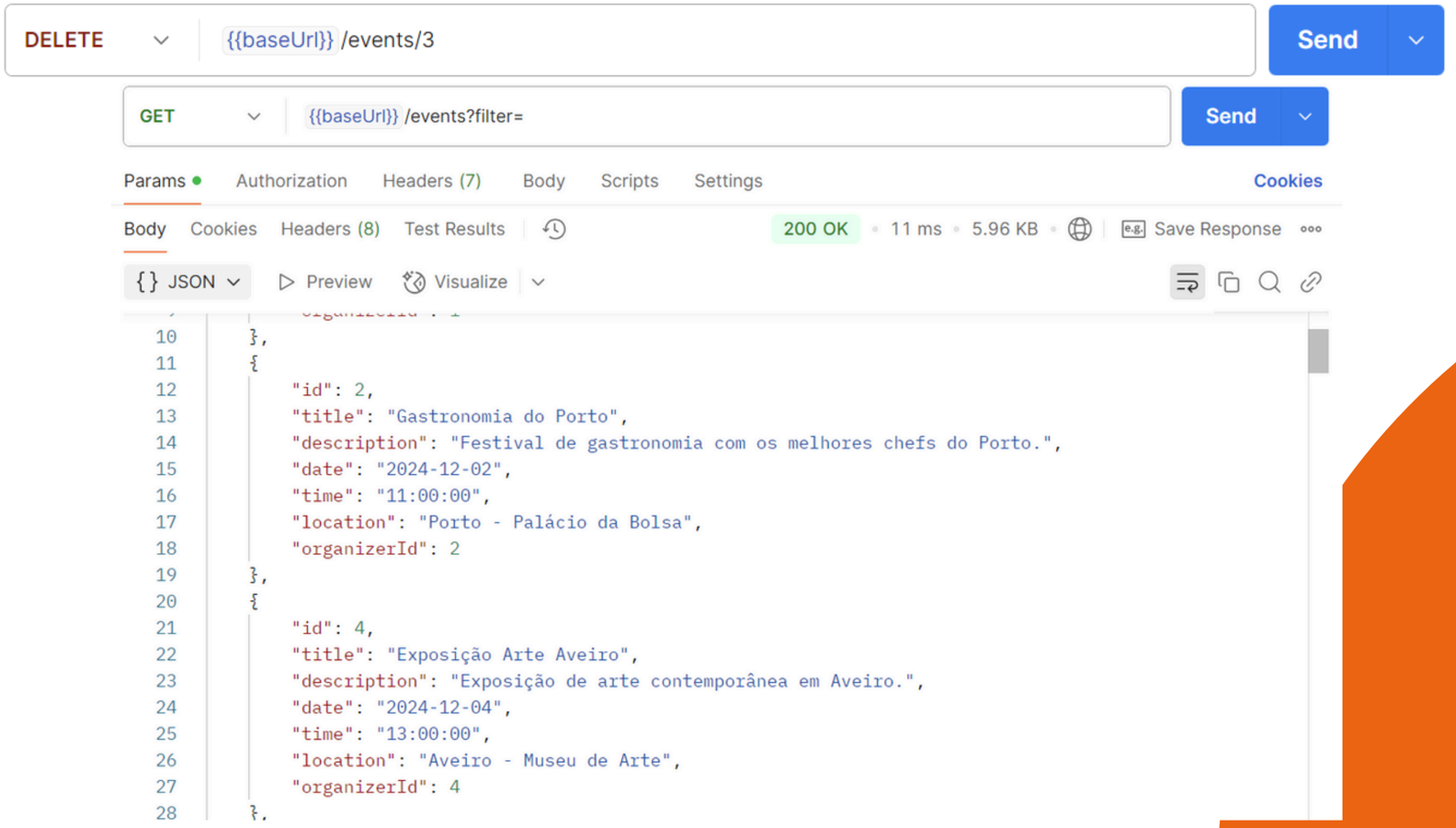
none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
{
  "title": "Festa de Aniversário",
  "description": "Aniversário da Maria",
  "date": "2025-1-2",
  "time": "10:00:00",
  "location": "Maia-Umaia",
  "organizerId": 2,
  "id": 31
}
```

Processo - Postman

DELETE EVENT{ID}

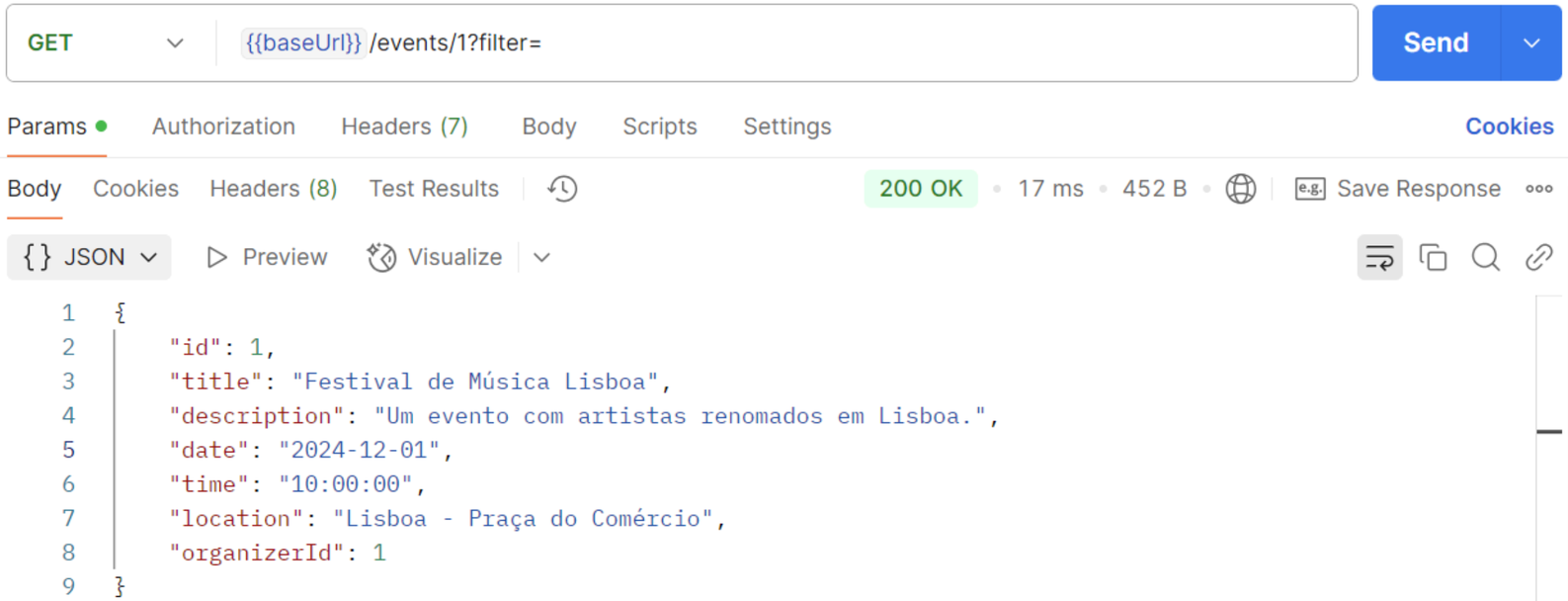
(Elimina Um Evento)



Processo - Postman

GET EVENT {ID}

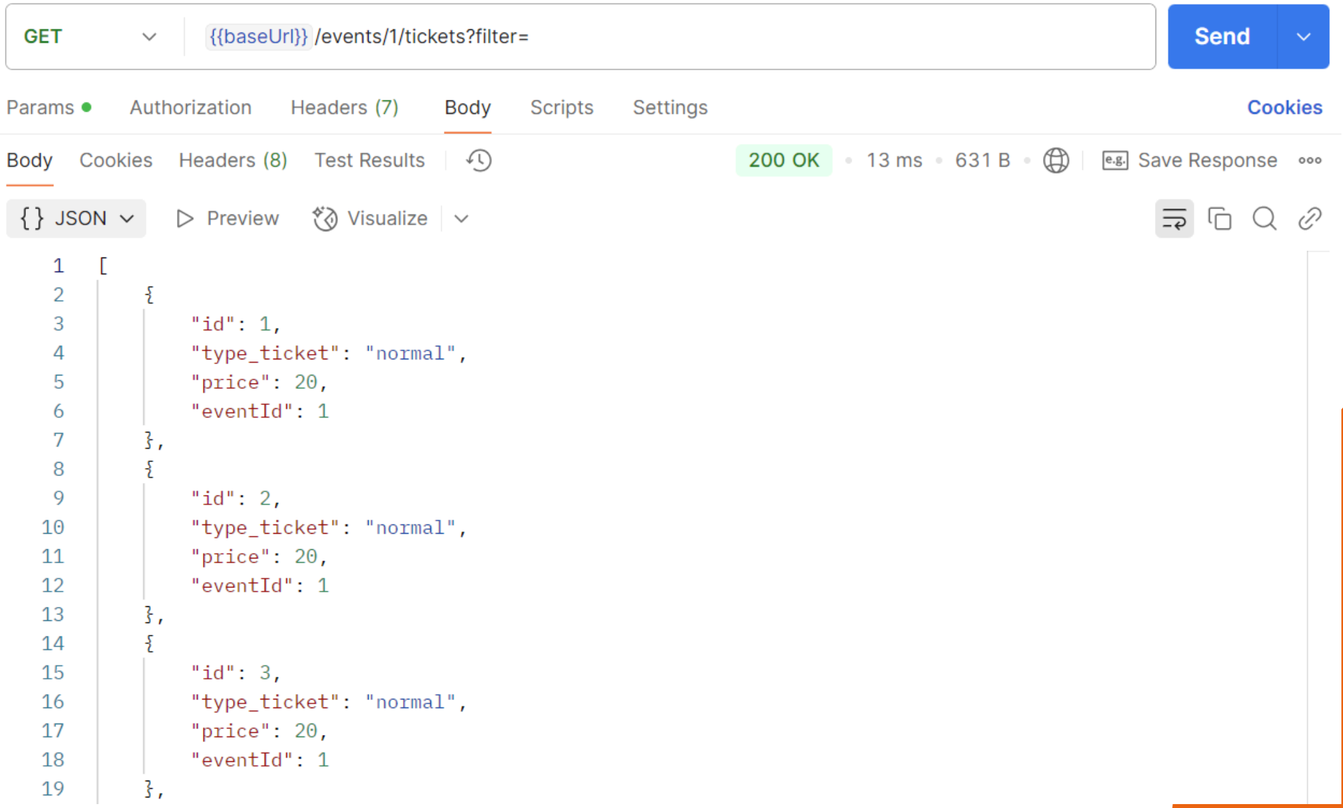
(Retorna Um Evento Especifico)



Processo - Postman

GET TICKET BY EVENT {ID}

(Retorna Todos os Bilhetes de um Evento)



Obrigado !

Beatriz Almeida
(A044416)
Carolina Fernandes
(A044897)
Pedro Venda
(A045464)