

PROJETO – TuneCatch

API do TuneCatch –.

A API do TuneCatch – *Maestro* – serve como ponte entre a interface do usuário (frontend) e sua lógica de negócios JavaScript com a camada de informações persistentes (banco de dados), sendo responsável por processar requisições e garantir a integridade das informações trafegadas.

A arquitetura da API segue princípios RPC e RESTful, com separação clara de responsabilidades por meio de verbos HTTP (**GET**, **POST**, **PATCH**, **PUT**, **DELETE**) e estruturação de rotas semânticas que representam os recursos principais do sistema — como usuários, playlists, autenticação e reprodução de músicas.

É possível segmentá-la em endpoints de esferas distintas, classificando-os de acordo com o recurso que ela está manipulando e/ou acessando.

Endpoints para Usuário

Rota: POST - maestro/usuario

Tipo: RESTful

Objetivo: Cadastrar um novo usuário no sistema

Requisição JSON:

```
{
  "username": input_usuario,
  "email/phone": input_usuario,
  "displayName": *opcional - omissão permitida*,
  "passwordHash": input_usuario,
  "avatarURL": *opcional - omissão permitida*
```

Resposta JSON:

201 - Created

```
{
  "message": "Usuário criado com sucesso!",
  "id": "u-0001",
  "username": "stargazer",
  "email/phone": "marco.email@dominio.com",
  "dateCreated": 01-06-2025,
```

400 - Bad Request{

```
  "error": "Campos obrigatórios vazios",
  "message": "Email/Telefone, Senha e Nome de Usuário são obrigatórios!"
```

}

400 - Bad Request{

```
  "error": "Validação de campos falhou",
  "message": "Um ou mais campos estão inválidos",
  "details": {
    "email": "Formato de Email/Telefone inválido.",
    "senha": "A senha deve ter pelo menos 6 caracteres.",
  }
```

}

409 - Conflict{

```
  "error": "Recurso já existente",
  "conflictField": *Campo que já existe no sistema*,
  "details": "0 {conflictField} já está em uso!"
```

}

Rota: POST maestro/usuario/fazer-login

Tipo: RPC-Styles

Objetivo: Enviar as credenciais para o sistema autenticar um usuário no sistema

Requisição JSON:

```
{
    "identifier": *email, telefone ou nome de usuário*
    "passwordHash": input_usuario_criptografado
}
```

Resposta JSON:

200 - Sucess

```
{
    "message": "Login Autorizado!",
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
    "user": {
        "id": "u-0001",
        "username": "stargazer"
    }
}
```

401 - Unauthorized{

```
    "error": "Credenciais inválidas",
    "message": "As credenciais enviadas estão incompletas ou incorretas!"
}
```

403 - Forbidden{

```
    "error": "Conta banida",
    "message": "Essa conta foi banida!"
}
```

Rota: GET maestro/usuario/{id_usuario}

Tipo: RESTful

Objetivo: Retornar todos os dados do usuário (exceto senha) pelo seu ID

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{
  "id": "u-0001 {id_usuario}",
  "username": "stargazer",
  "displayName": "marco",
  "email/phone": "marco@dominio.com.br",
  "playlistsCreated": ["p-000001", "p-000002"],
  "playlistsCaught": [
    {
      "playlist": {
        "id": "p-000003",
        "name": "NEW DANCE",
        "creator": "u-0002",
        "description": " ",
        "coverImageURL": "https://br.pinterest.com/pin/",
      },
      "caughtVisibleOnProfile": "true"
    },
    {
      "playlist": {
        "id": "p-000004",
        "name": "just tell me why",
        "creator": "u-0002",
        "description": " ",
        "coverImageURL": "https://br.pinterest.com/pin",
      },
      "caughtVisibleOnProfile": "false"
    }
  ],
  "followers": ["u-0002", "u-0002"],
  "following": ["u-0002", "u-0003"],
  "userBadges": [
    {
      "badge": {
        "id": "b-001",
        "name": "Fundador",
        "description": "Fez parte dos 10 primeiros usuários",
        "iconURL": "https://armazenamento/badge/founder-5dsp.jpg"
      },
      "isVisibleOnProfile": "true"
    }
  ],
  "avatarURL": "https://br.pinterest.com/pin/599682506668827088/",
  "dateCreated": 15-02-2025,
  "dataUpdated": 10-04-2025
}
```

```
400 - Bad Request{
  "error": "Formato de ID inválido",
  "message": "O ID enviado não está no formato aceito pelo sistema"
}
404 - Not Found{
  "error": "ID não existe",
  "message": "Essa conta não existe no sistema"
}
```

Rota: GET maestro/usuario/nome-usuario/{nome_usuario}

Tipo: RESTful

Objetivo: Retornar todos os dados públicos do usuário a partir do seu username

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{
  "id": "u-0001",
  "username": "stargazer" {nome_usuario},
  "displayName": "marco",
  "email/phone": "marco@dominio.com.br",
  "playlistsCreated": ["p-000001"],
  "playlistsCaught":
    [{
      "playlist": {
        "id": "p-000003",
        "name": "NEW DANCE",
        "creator": "u-0002",
        "description": " ",
        "coverImageURL": "https://br.pinterest.com/pin/",
      },
      "caughtVisibleOnProfile": "true"
    }
  ],
  "followers": ["u-0002", "u-0003"],
  "following": ["u-0002", "u-0003"],
  "userBadges":
    [{
      "badge": {
        "id": "b-001",
        "name": "Fundador",
        "description": "Fez parte dos 10 primeiros usuários",
        "iconURL": "https://armazenamento/badge/founder-5dsp.jpg"
      },
      "isVisibleOnProfile": "true"
    }
  ],
  "avatarURL": "https://br.pinterest.com/pin/599682506668827088/",
  "dateCreated": 15-02-2025,
  "dataUpdated": 10-04-2025
}
```

404 - Not Found{

```
  "error": "Username não existe",
  "message": "Esse username não existe no sistema"
}
```

Rota: GET maestro/usuario/nome-exibição?q={nome_exibição}

Tipo: RESTful

Objetivo: Retornar uma lista de usuário que compartilham o mesmo nome de exibição

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
[
  "users": [{
    "id": "u-0001",
    "username": "stargazer",
    "displayName": "marco {nome_exibição}",
    "avatarURL": "https://br.pinterest.com/pin",
  },
  {
    "id": "u-1285",
    "username": "WAVELENGTH",
    "displayName": "marco {nome_exibição}",
    "avatarURL": "https://br.pinterest.com/pin",
  }
]
```

200 - OK, but no data found

```
{
  "users": []
}
```

Rota: PATCH maestro/usuario/{id_usuario}

Tipo: RESTful

Objetivo: Modificar parcialmente o usuário com as informações enviadas por ele

Requisição JSON:

```
{
  "username": input_usuario *opcional*,
  "displayName": input_usuario *opcional*,
  "email/phone": input_usuario *opcional*,
  "password": {
    "currentPassword": input_usuario
    "newPassword": "input_usuario *opcional*"
  },
  "playlistsCaught":
    [{
      "playlist": {
        "id": "p-000003",
      },
      "caughtVisibleOnProfile": input_usuario *opcional*,
    },
    {
      "playlist": {
        "id": "p-000004",
      },
      "caughtVisibleOnProfile": input_usuario *opcional*,
    }
  ],
  "userBadges":
    [{
      "badge": {
        "id": "b-001",
      },
      "isVisibleOnProfile": input_usuario *opcional*,
    }
  ],
  "avatarURL": input_usuario *opcional*,
}
```

Resposta JSON:

```
200 - Sucess
{
  "message": "Alterações feitas com sucesso",
}

400 - Bad Request{
  "error": "Formato de ID inválido",
  "message": "O ID enviado não está no formato aceito pelo sistema"
}

400 - Bad Request{
  "error": "Atualização proibida",
  "unauthorizedField": *Campo que modificação não foi autorizada*,
  "details": "A atualização no campo {unauthorizedField} foi barrada!"
}
```



```
401 - Unauthorized{
  "error": "Senha incorreta",
  "message": "A senha atual não está correta"
}
404 - Not Found{
  "error": "Recurso não existe",
  "message": "O recurso que quer ser atualizado não existe!"
}
409 - Conflict{
  "error": "Recurso já existente",
  "conflictField": *Campo que já existe no sistema*,
  "details": "O {conflictField} já está em uso!"
}
```

Rota: POST maestro/usuario/seguir/{id_usuario}

Tipo: RPC-Style

Objetivo: Adicionar ou remover um usuário da lista de “seguindo”.

Requisição JSON: Não aplicável

Resposta JSON:

```
200 - Sucess
{
    "message": "Lista de usuários seguidos atualizada!",
    "user": {
        "id": "u-0001",
        "username": "stargazer",
        "following": ["u-0002 {id_usuario}"],
    }
}

400 - Bad Request{
    "error": "Formato de ID inválido",
    "message": "O ID enviado não está no formato aceito pelo sistema"
}

403 - Forbidden{
    "error": "ID de usuário é igual ao ID a ser seguido",
    "message": "Não é possível seguir a si mesmo!"
}

404 - Not Found{
    "error": "ID de usuário não existe",
    "message": "O usuário para ser seguido não existe."
}
```

Rota: POST maestro/usuario/banir-perfil/{id_usuario}

Tipo: RPC-Styles

Objetivo: Bane um usuário do sistema

Requisição JSON:

```
{
  "reason": input_usuario
}
```

Resposta JSON:

```
200 - Sucess
{
  "message": "Usuário banido com sucesso!",
  "message": 07-06-2025,
  "user": {
    "id": "u-0001",
    "username": "stargazer",
    "username": "Spam de playlists"
  }
}

400 - Bad Request{
  "error": "Formato de ID inválido.",
  "message": "O ID enviado não está no formato aceito pelo sistema."
}

401 - Unauthorized{
  "error": "Tentativa de banimento por pessoal não autorizado.",
  "message": "Somente administradores podem banir um usuário!"
}

404 - Bad Request{
  "error": "ID de usuário não existe.",
  "message": "A conta a ser banida não existe."
}

409 - Conflict{
  "error": "ID de usuário já banido.",
  "message": "Essa conta já foi banida."
}
```

Rota: DELETE maestro/usuario/{id_usuario}

Tipo: RESTful

Objetivo: Excluir um perfil de usuário do sistema.

Requisição JSON:

Resposta JSON:

204 - Sucess

```
{  
}
```

403 - Forbidden

```
{  
  "error": "Tentativa de excluir conta de terceiro.",  
  "message": "Não é possível excluir a conta de uma outra pessoa!"  
}
```

404 - Not Found{

```
  "error": "ID de usuário não existe.",  
  "message": "A conta a ser excluída não existe."  
}
```

Endpoints para Playlist

Rota: POST - maestro/playlist

Tipo: RESTful

Objetivo: Criar uma nova playlist

Requisição JSON:

```
{
  "name": input_usuario,
  "description": input_usuario *Opcional*,
  "isVisible": true/false *True por padrão*,
  "tags": input_usuario,
  "coverImageURL": input_usuario *Opcional*,
}
```

Resposta JSON:

```
201 - Created {
  "id": "p-000001",
  "name": "duskbreaker",
  "creator": "u-0001",
  "description": "i just wanna drive",
  "isVisible": true,
  "catches": 0,
  "tags": ["t-001", "t-005", "t-015", "t-031"],
  "tracks": []
  "coverImageURL": "https://br.pinterest.com/pin/593419688438357287/",
  "dateCreated": 24-05-2025,
  "dateModified": 29-05-2025
}

400 - Bad Request{
  "error": "Tags inválidas",
  "message": "Não é possível atribuir tags que não existem à playlist!",
  "invalidTags": [ *Lista com as Tags inválidas*]
}

400 - Bad Request{
  "error": "Nome vazio",
  "message": "O nome da playlist não pode estar vazio!"
}

400 - Bad Request{
  "error": "Playlist sem tags",
  "message": "Uma playlist precisa ter tags!"
}

413 - Payload Too Large{
  "error": "Número de tags excede o limite",
  "message": "Uma playlist pode ter até 10 tags!"
}
```

Rota: PATCH - maestro/playlist/{id_playlist}

Tipo: RESTful

Objetivo: Modificar parcialmente o usuário com as informações enviadas por ele

Requisição JSON:

```
{
  "name": input usuário *Opcional*,
  "description": input_usuario *Opcional*,
  "isVisible": true/false *Opcional*,
  "tags": input usuário *Opcional*,
  "coverImageURL": input_usuario *Opcional*,
}
```

Resposta JSON:

200 - Sucess

```
{
  "message": "Playlist modificada com sucesso"
  "playlist": {
    "id": "p-000001 {id_playlist}",
    "name": "Nighttearer",
    "creator": "u-0001",
    "description": "i just wanna drive",
    "isVisible": true,
    "catches": 1783,
    "tags": ["t-001", "t-005", "t-015", "t-031"],
    "tracks": ["spotify:track:crashing", "spotify:track:drive"]
    "coverImageURL": "https://br.pinterest.com/pin/",
    "dateCreated": 24-05-2025,
    "dateModified": 07-06-2025
  }
}

400 - Bad Request{
  "error": "Tags inválidas",
  "message": "Não é possível atribuir tags que não existem à playlist!",
  "invalidTags": [ *Lista com as Tags inválidas* ]
}

400 - Bad Request{
  "error": "Nome vazio",
  "message": "O nome da playlist não pode estar vazio!"
}

400 - Bad Request{
  "error": "Playlist sem tags",
  "message": "Uma playlist precisa ter tags!"
}

413 - Payload Too Large{
  "error": "Número de tags excede o limite",
  "message": "Uma playlist pode ter até 10 tags!"
}
```

Rota: DELETE - maestro/playlist/{id_playlist}

Tipo: RESTful

Objetivo: Excluir uma playlist do sistema.

Requisição JSON: Não aplicável

Resposta JSON:

204 - Sucess

```
{  
}
```

403 - Forbidden{

"error": "Tentativa de excluir playlist de terceiro",

"message": "Não é possível excluir a playlist de uma outra pessoa!"

```
}
```

404 - Not Found{

"error": "ID de playlist não existe",

"message": "A playlist a ser excluída não existe."

```
}
```

Rota: POST maestro/playlist/curtir/{id_playlist}

Tipo: RPC-Style

Objetivo: Adicionar ou remover uma playlist da lista de “curtidas”.

Requisição JSON: Não Aplicável

Resposta JSON:

```
200 - Sucess
{
    "message": "Lista de playlists curtidas atualizada!",
    "playlistsCaught": [{
        "playlist": {
            "id": "p-000003 {id_playlist}",
        },
    ]
}
400 - Bad Request{
    "error": "Formato de ID inválido",
    "message": "O ID enviado não está no formato aceito pelo sistema"
}
403 - Forbidden{
    "error": "ID de playlist é alguma criada pelo usuário",
    "message": "Não é possível curtir uma playlist de si mesmo!"
}
404 - Not Found{
    "error": "ID de playlist não existe",
    "message": "A playlist a ser curtida não existe."
}
```


Rota: POST maestro/playlist/ocultar-curtida/{id_playlist}

Tipo: RPC-Style

Objetivo: Modificar o status de visibilidade de uma playlist curtida no perfil do usuário

Requisição JSON:

```
{
    "visibilityStatus": True/False
}
```

Resposta JSON:

200 - Sucess

```
{
    "message": "Status de visibilidade da playlist curtidas atualizada!",
    "playlistsCaught": [{
        "playlist": {
            "id": "p-000003 {id_playlist}",
            "caughtVisibleOnProfile": True/False
        },
    ]
}
```

400 - Bad Request{

```
    "error": "Formato de ID inválido",
    "message": "O ID enviado não está no formato aceito pelo sistema."
}
```

}

400 - Bad Request{

```
    "error": "Status de visibilidade inválido",
    "message": "O Status de Visibilidade definido não é aceito pelo sistema."
}
```

}

403 - Forbidden{

```
    "error": "ID de playlist não foi curtida pelo usuário",
    "message": "Não é possível modificar o status de visibilidade de uma playlist que não foi curtida!"
}
```

}

404 - Not Found{

```
    "error": "ID de playlist não existe",
    "message": "A playlist que se quer modificar o status de visualização não existe."
}
```

}

Rota: POST maestro/playlist/adicionar-musica/{id_playlist}

Tipo: RPC-Style

Objetivo: Adicionar uma música à playlist

Requisição JSON:

```
{
  "trackSpotifyID": "spotify:track:crashing"
}
```

Resposta JSON:

200 - Sucess

```
{
  "message": "Música adicionada com sucesso a playlist",
  "playlist": {
    "id": "p-000003 {id_playlist}",
    "tracks": [
      {
        "track": {
          "spotifyID": "spotify:track:crashing",
          "trackName": "Crashing (with Kalis Uchi)",
          "artistName": "d4vd",
          "albumName": "WITHERED",
          "albumCoverURL": "https://spotify-api/image.jpg"
        }
      }
    ],
    "dateCreated": "24-05-2025",
    "dateModified": "05-06-2025"
  }
}
```

400 - Bad Request{

```
  "error": "Formato de ID inválido",
  "message": "O ID enviado não está no formato aceito pelo sistema"
}
```

403 - Forbidden{

```
  "error": "ID de playlist não pertence ao usuário",
  "message": "Não é possível adicionar músicas à playlists que não pertencem à você"
}
```

404 - Not Found{

```
  "error": "ID de playlist não existe",
  "message": "A playlist em que se quer adicionar músicas não existe."
}
```

404 - Not Found{

```
  "error": "ID de música não existe",
  "message": "O ID enviado não existe no spotify."
}
```

502 - Bad Gateway e 504 Gateway Timeout{

```
  "error": "Erro na API do Spotify",
  "message": "Não foi possível se conectar com a API do Spotify!"
}
```

Rota: DELETE maestro/playlist/remove-musica/{id_playlist}/{id_musica}

Tipo: RPC-Style

Objetivo: Remove uma música à playlist

Requisição JSON: Não Aplicável

Resposta JSON:

```
200 - Sucess
{
  "message": "Música removida com sucesso da playlist",
  "playlist": {
    "id": "p-000003 {id_playlist}",
    "tracks":
      [],
    "dateCreated": 24-05-2025,
    "dateModified": 06-06-2025
  }
}

400 - Bad Request{
  "error": "Formato de ID inválido",
  "message": "O ID enviado não está no formato aceito pelo sistema"
}

403 - Forbidden{
  "error": "ID de playlist não pertence ao usuário",
  "message": "Não é possível remover músicas de playlists que não
pertencem à você"
}

404 - Not Found{
  "error": "ID de playlist não existe",
  "message": "A playlist em que se quer remover músicas não existe."
}

404 - Not Found{
  "error": "ID de música não existe",
  "message": "O ID enviado não existe no spotify."
}

404 - Not Found{
  "error": "ID de música não existe na playlist",
  "message": "A música a ser deletada não está na playlist."
}

502 - Bad Gateway e 504 Gateway Timeout{
  "error": "Erro na API do Spotify",
  "message": "Não foi possível se conectar com a API do Spotify!"
}
```

Rota: GET maestro/playlist/{id_playlist}

Tipo: RESTful

Objetivo: Retorna todas as informações da playlist com base no ID

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{
  "id": "p-000003 {id_playlist}",
  "name": "duskbreaker",
  "creator": "u-0001",
  "description": "i just wanna drive",
  "isVisible": true,
  "catches": 1783,
  "tags": ["t-001", "t-005", "t-015", "t-031"],
  "tracks":
    [{
      "track": {
        "spotifyID": "spotify:track:crashing",
        "trackName": "Crashing (with Kali Uchis)",
        "artistName": "d4vd",
        "albumName": "WITHERED",
        "albumCoverURL": "https://spotify-api/image/d4vd-album2.jpg"
      },
      "track": {
        "spotifyID": "spotify:track:ilysmih",
        "trackName": "ILYSMIH",
        "artistName": "Kali Uchis",
        "albumName": "Sincerely,",
        "albumCoverURL": "https://spotify-api/image/kali-album2.jpg"
      }
    ]
  "coverImageURL": "https://br.pinterest.com/pin/593419688438357287/",
  "dateCreated": 24-05-2025,
  "dateModified": 29-05-2025
}

400 - Bad Request{
  "error": "Formato de ID inválido",
  "message": "O ID enviado não está no formato aceito pelo sistema"
}

404 - Not Found{
  "error": "ID de playlist não existe",
  "message": "A playlist em que se quer remover músicas não existe."
}
```

Rota: GET maestro/playlist/nome-playlist/{id_criador}

Tipo: RESTful

Objetivo: Retorna playlists públicas que um determinado usuário criou.

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Success

```
{
  "playlists": [{
    "id": "p-000013",
    "name": "fastlane",
    "creator": "u-0001 {id_creator}",
    "description": "drop a gear and disappear",
    "isVisible": true,
    "catches": 1783,
    "tags": ["t-013", "t-005"],
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 24-05-2025,
    "dateModified": 29-05-2025
  },
  {
    "id": "p-000001",
    "name": "duskbreaker",
    "creator": "u-0001 {id_creator}",
    "description": "just wanna relax",
    "isVisible": true,
    "catches": 13,
    "tags": ["t-001", "t-005"],
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 02-06-2025,
    "dateModified": 05-06-2025
  }
]
```

200 - Ok, but no content

```
{
  "playlists": []
}
```

404 - Not Found{

```
  "error": "Nome de usuário não existe",
  "message": "0 usuário buscado não existe."
}
```

Rota: GET maestro/playlist/nome-playlist?q={nome_playlist}

Tipo: RESTful

Objetivo: Retorna playlists públicas com base no seu nome

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Success

```
{
  "playlists": [{
    "id": "p-000012",
    "name": "MAKE IT POP {nome_playlist}",
    "creator": "u-0002",
    "description": " ",
    "isVisible": true,
    "catches": 1783,
    "tags": ["t-016", "t-019"],
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 24-05-2025,
    "dateModified": 29-05-2025
  },
  {
    "id": "p-000016",
    "name": "MAKE IT POP {nome_playlist}",
    "creator": "u-0001",
    "description": "from across the world",
    "isVisible": true,
    "catches": 13,
    "tags": ["t-016", "t-019", "t-022"],
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 02-06-2025,
    "dateModified": 05-06-2025
  }
]
```

}

200 - Ok, but no content

```
{
  "playlists": []
}
```

Rota: GET maestro/playlist/tag?q={tags}

Tipo: RESTful

Objetivo: Retorna playlists públicas que possuem determinada(s) tag(s)

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Success

```
{
  "playlists": [{
    "id": "p-000044",
    "name": "no ending darkness",
    "creator": "u-0001",
    "description": "if one of us die, i hope i die first",
    "isVisible": true,
    "catches": 152,
    "tags": ["t-003", "t-025"] {tags},
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 24-05-2025,
    "dateModified": 29-05-2025
  },
  {
    "id": "p-000081",
    "name": "rainbow shatter",
    "creator": "u-0002",
    "description": " ",
    "isVisible": true,
    "catches": 13,
    "tags": ["t-003", "t-005"] {tags},
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 02-06-2025,
    "dateModified": 05-06-2025
  }
]
```

}

200 - Ok, but no content

```
{
  "playlists": []
}
```

Rota: GET maestro/playlist/musica?q={spotify_id}

Tipo: RESTful

Objetivo: Retorna playlists públicas que possuem determinada música

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Success

```
{
  "playlists": [{
    "id": "p-000034",
    "name": "the last sunset",
    "creator": "u-0002",
    "description": "the sun - it is not coming back...",
    "isVisible": true,
    "catches": 568,
    "tags": ["t-001", "t-025", "t-030"],
    "tracks": [
      {
        "track": {
          "spotifyID": "spotify:track:crashing{spotify_id}",
          "trackName": "Crashing (with Kalis Uchi)",
          "artistName": "d4vd",
        }
      }
    ]
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 24-05-2025,
    "dateModified": 29-05-2025
  },
  {
    "id": "p-000948",
    "name": "light thread",
    "creator": "u-0001",
    "description": "the last drop of hope",
    "isVisible": true,
    "catches": 13,
    "tags": ["t-005", "t-009"],
    "tracks": [
      {
        "track": {
          "spotifyID": "spotify:track:crashing{spotify_id}",
          "trackName": "Crashing (with Kalis Uchi)",
          "artistName": "d4vd",
        }
      }
    ]
    "coverImageURL": "https://br.pinterest.com/pin",
    "dateCreated": 02-06-2025,
    "dateModified": 05-06-2025
  }
]
```

200 - Ok, but no content

```
{
  "playlists": []
}
```

Endpoints para Tags:

Rota: GET maestro/admin/tag

Tipo: RESTful

Objetivo: Permite retornar todas as tags cadastradas no sistema.

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{
  "tags": [{
    "tag": {
      "id": "t-001",
      "name": "Late Night Drive",
      "category": "Ocasião",
      "iconEmoji": "🌙🚗"
    },
    "tag": {
      "id": "t-002",
      "name": "Chill",
      "category": "Vibe",
      "iconEmoji": "🧊"
    },
    "tag": {
      "id": "t-003",
      "name": "Pop",
      "category": "Gênero Musical",
      "iconEmoji": "🎵"
    }
  ]
}
```

Rota: GET maestro/admin/tag/{id_tag}

Tipo: RESTful

Objetivo: Permite retornar uma tag cadastrada a partir do seu ID.

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{  
  "id": "t-001 {id_tag}",  
  "name": "Late Night Drive",  
  "category": "Ocasão",  
  "iconEmoji": "🌙🚗"  
}
```

400 - Bad Request{

```
"error": "Formato de ID inválido",  
"message": "O ID enviado não está no formato aceito pelo sistema"
```

}

404 - Not Found{

```
"error": "Tag não existe",  
"message": "A tag buscada não existe."
```

}

Rota: GET maestro/admin/tag/{nome_tag}

Tipo: RESTful

Objetivo: Permite retornar uma tag cadastrada a partir do seu nome.

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{
  "id": "t-001",
  "name": "Late Night Drive {nome_tag}",
  "category": "Ocasão",
  "iconEmoji": "🌙🚗"
}
```

200 - OK, but no content

```
{
  []
}
```

Rota: POST maestro/admin/tag

Tipo: RESTful

Objetivo: Permite que os administradores cadastrem uma nova tag ao sistema.

Requisição JSON:

```
{
  "name": input,
  "category": input,
  "iconEmoji": input
}
```

Resposta JSON:

200 - Sucess

```
{
  "message": "Tag adicionada com sucesso ao sistema",
  "tags": {
    "id": "t-001",
    "name": "Late Night Drive",
    "category": "Ocasião",
    "iconEmoji": "🌙🚗"
  }
}
```

400 - Bad Request{

```
  "error": "Icon inválido",
  "message": "O ícone escolhido para a tag é inválido"
```

}

400 - Bad Request{

```
  "error": "Categoria inválida",
  "message": "A categoria escolhida é inválida."
```

}

409 - Conflict{

```
  "error": "Nome de Tag já usada",
  "message": "Esse nome de tag já existe no sistema!"
```

}

Rota: PATCH maestro/admin/tag/{id_tag}

Tipo: RESTful

Objetivo: Permite que os administradores modifiquem uma tag ao sistema.

Requisição JSON:

```
{
  "name": input *opcional*,
  "category": input *opcional*,
  "iconEmoji": input *opcional*
}
```

Resposta JSON:

200 - Sucess

```
{
  "message": "Tag modificada com sucesso",
  "tags": {
    "id": "t-001 {id_tag}",
    "name": "Night Drive",
    "category": "Ocasão",
    "iconEmoji": "🌙🚗"
  }
}
```

400 - Bad Request{

```
"error": "Icon inválido",
"message": "O ícone escolhido para a tag é inválido"
```

}

400 - Bad Request{

```
"error": "Categoria inválida",
"message": "A categoria escolhida é inválida."
```

}

400 - Bad Request{

```
"error": "Formato de ID inválido",
"message": "O ID enviado não está no formato aceito pelo sistema"
```

}

404 - Not Found{

```
"error": "Tag não encontrada",
"message": "A tag que quer ser modificada não existe."
```

}

409 - Conflict{

```
"error": "Nome de Tag já usada",
"message": "Esse nome de tag já existe no sistema!"
```

}

Rota: DELETE maestro/admin/tag/{id_tag}

Tipo: RESTful

Objetivo: Permite que os administradores deletem uma tag ao sistema.

Requisição JSON:

Resposta JSON:

204 - Sucess

```
{  
}
```

400 - Bad Request{

"error": "Formato de ID inválido",

"message": "O ID enviado não está no formato aceito pelo sistema"

```
}
```

404 - Not Found{

"error": "Tag não encontrada",

"message": "A tag referenciada não existe."

```
}
```

Endpoints para Badges:

Rota: GET maestro/admin/badge

Tipo: RESTful

Objetivo: Permite retornar todas as badges cadastradas no sistema.

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
"badges": [{
  "badge": {
    "id": "b-001",
    "name": "Fundador",
    "description": "Fez parte dos 10 primeiros usuários",
    "iconURL": "https://armazenamento/badge/founder-5dsp.jpg"
  },
  "badge": {
    "id": "b-002",
    "name": "Colecionador",
    "description": "Crie 25 Playlist",
    "iconURL": "https://armazenamento/badge/collector-8zdz.jpg"
  },
  "badge": {
    "id": "b-003",
    "name": "Eclético",
    "description": "Adicione 5 tags de gênero à uma playlist",
    "iconURL": "https://armazenamento/badge/eclectic-6bmn.jpg"
  }
}]
```

Rota: GET maestro/admin/badge/{id_badge}

Tipo: RESTful

Objetivo: Permite retornar uma badge cadastrada a partir do seu ID.

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{  
    "id": "b-003 {id_badge}",  
    "name": "Eclético",  
    "description": "Adicione 5 tags de gênero à uma playlist",  
    "iconURL": "https://armazenamento/badge/eclectic-6bmn.jpg"  
}
```

400 - Bad Request{

```
"error": "ID em formato inválido",  
"message": "O ID está em um formato inválido para o sistema."
```

}

404 - Not Found{

```
"error": "Badge não existe",  
"message": "A badge buscada não existe."
```

}

Rota: GET maestro/admin/badge/{nome_badge}

Tipo: RESTful

Objetivo: Permite retornar uma badge cadastrada a partir do seu ID.

Requisição JSON: Não Aplicável

Resposta JSON:

200 - Sucess

```
{
  "id": "b-003",
  "name": "Eclético {nome_badge}",
  "description": "Adicione 5 tags de gênero à uma playlist",
  "iconURL": "https://armazenamento/badge/eclectic-6bmn.jpg"
}

200 - Ok, but no content{
  []
}
```

Rota: POST maestro/admin/badge/{id_badge}

Tipo: RESTful

Objetivo: Permite cadastrar uma nova badge ao sistema.

Requisição JSON:

```
{
    "name": input,
    "category": input,
    "iconEmoji": input
}
```

Resposta JSON:

200 - Sucess

```
{
    "message": "Badge criada com sucesso",
    "badge": {
        "id": "b-002 {id_badge}",
        "name": "Colecionador",
        "description": "Crie 25 Playlist",
        "iconURL": "https://armazenamento/badge/collector-8zdz.jpg"
    }
}
```

400 - Bad Request{

```
    "error": "Icon inválido",
    "message": "O ícone escolhido para a badge é inválido"
```

}

400 - Bad Request

```
{
    "error": "Formato de ID inválido",
    "message": "O ID enviado não está no formato aceito pelo sistema"
}
```

409 - Conflict{

```
    "error": "Nome de Badge já usada",
    "message": "Esse nome de badge já existe no sistema!"
```

}

Rota: PATCH maestro/admin/badge/{id_badge}

Tipo: RESTful

Objetivo: Permite modificar uma badge ao sistema.

Requisição JSON:

```
{
    "name": input *opcional*,
    "category": input *opcional*,
    "iconEmoji": input *opcional*
}
```

Resposta JSON:

200 - Sucess

```
{
    "message": "Badge modificada com sucesso",
    "badge": {
        "id": "b-002 {id_badge}",
        "name": "Colecionador",
        "description": "Crie 30 Playlist",
        "iconURL": "https://armazenamento/badge/collector-8zdj.jpg"
    }
}
```

400 - Bad Request{

```
    "error": "Icon inválido",
    "message": "O ícone escolhido para a badge é inválido!"
}
```

}

400 - Bad Request

```
{
    "error": "Formato de ID inválido",
    "message": "O ID enviado não está no formato aceito pelo sistema"
}
```

404 - Not Found{

```
    "error": "Badge não encontrada",
    "message": "A badge referenciada não existe."
}
```

}

409 - Conflict{

```
    "error": "Nome de Badge já usada",
    "message": "Esse nome de badge já existe no sistema!"
}
```

}

Rota: DELETE maestro/admin/badge/{id_badge}

Tipo: RESTful

Objetivo: Permite deletar uma badge ao sistema.

Requisição JSON:

Resposta JSON:

204 - Sucess

```
{  
}
```

400 - Bad Request

```
{  
  "error": "Formato de ID inválido",  
  "message": "O ID enviado não está no formato aceito pelo sistema"  
}
```

404 - Not Found{

```
  "error": "Badge não encontrada",  
  "message": "A badge referenciada não existe."  
}
```