

Documento de Rotas - Projeto SentinelGuard

Nome: Rodrigo Hendrick Monteiro Santos Fortes

Base de Dados{Nome: sentinel, Usuário: postregs, Host: localhost}

Rotas do Usuário:

- [POST User] - Criar um novo usuário

Endpoint: <http://localhost:3001/teste/users/create>

```
{
  "first_name": "John",
  "last_name": "Doe",
  "email": "john.doe@example.com",
  "password": "securepassword123",
  "tipo_user": "admin"
}
```

- [GET All_users] - Obter todos os usuários

Endpoint: <http://localhost:3001/teste/users>

- [GET User by Id] - Obter um usuário pelo ID

Endpoint: <http://localhost:3001/teste/users/:id>

- [PUT Update-User] - Atualizar um usuário pelo ID

Endpoint: <http://localhost:3001/teste/users/update/:id>

- [DELETE User] - Deletar um usuário pelo ID

Endpoint: <http://localhost:3001/teste/users/delete/:id>

Rotas dos Funcionários:

- [POST Func] - Criar um novo funcionário

Endpoint: <http://localhost:3001/teste/func>

```
{
  "first_name": "Jane",
  "last_name": "Smith",
}
```

```
"tipo_funcionario": "security",  
"image_face_funcionario":  
"..."  
}
```

- [GET All_func] - Obter todos os funcionários

Endpoint: http://localhost:3001/teste/all_func

- [GET Func by Id] - Obter um funcionário pelo ID

Endpoint: <http://localhost:3001/teste/func/:id>

- [GET Func by Image] - Procurar um funcionário através de reconhecimento facial (não implementado)

Endpoint: <http://localhost:3001/teste/func/image>

- [PUT Update_Func] - Atualizar um funcionário

Endpoint: http://localhost:3001/teste/update_func

- [DELETE Func] - Deletar um funcionário pelo ID

Endpoint: http://localhost:3001/teste/delete_funcionario/:id

Rotas de Vídeo:

- [POST Video] - Criar um novo vídeo

Endpoint: <http://localhost:3001/teste/videos>

```
{  
  "camara_id": 1,  
  "data_hora_evento": "2024-05-20T14:30:00Z",  
  "arquivo": "video_evento.mp4",  
  "tamanho": 1048576  
}
```

- [GET All_Videos] - Obter todos os vídeos

Endpoint: http://localhost:3001/teste/all_videos

- [GET Video by Id] - Obter um vídeo pelo ID

Endpoint: <http://localhost:3001/teste/videos/:id>

- [PUT Update_Video] - Atualizar um vídeo pelo ID

Endpoint: <http://localhost:3001/teste/videos/:id>

- [DELETE Video] - Deletar um vídeo pelo ID

Endpoint: <http://localhost:3001/teste/videos/delete/:id>

- [GET Video with Camera Location] - Obter um vídeo pelo ID com a localização da câmera

Endpoint: <http://localhost:3001/teste/videos/:id/location>

Rotas de Eventos:

- [POST Evento] - Criar um novo evento

Endpoint: <http://localhost:3001/teste/eventos>

```
{
  "usuario_id": 1,
  "camara_id": 2,
  "tipo": "movimentação",
  "descricao": "Movimentação detectada na entrada",
  "data_hora_evento": "2024-05-20T14:30:00Z"
}
```

- [GET All_Eventos] - Obter todos os eventos

Endpoint: http://localhost:3001/teste/all_eventos

- [GET Evento by Id] - Obter um evento pelo ID

Endpoint: <http://localhost:3001/teste/eventos/:id>

- [PUT Update_Evento] - Atualizar um evento pelo ID

Endpoint: <http://localhost:3001/teste/eventos/:id>

- [DELETE Evento] - Deletar um evento pelo ID

Endpoint: <http://localhost:3001/teste/eventos/delete/:id>

- [GET Evento with Camera Location] - Obter um evento pelo ID com a localização da câmera

Endpoint: <http://localhost:3001/teste/eventos/:id/location>

Rotas de Reconhecimento de Imagem:

- [POST Image Recognition] - Criar um novo reconhecimento de imagem

Endpoint: http://localhost:3001/teste/image_recognition

```
{  
  "camara_id": 1,  
  "id_funcionario": 2,  
  "desconhecido": false,  
  "acesso_authorized": "Yes"  
}
```

- [GET All_Image Recognitions] - Obter todos os reconhecimentos de imagem

Endpoint: http://localhost:3001/teste/image_recognition

- [GET Image Recognition by Id] - Obter um reconhecimento de imagem pelo ID

Endpoint: http://localhost:3001/teste/image_recognition/:id

- [PUT Update_Image Recognition] - Atualizar um reconhecimento de imagem pelo ID

Endpoint: http://localhost:3001/teste/image_recognition/:id

- [DELETE Image Recognition] - Deletar um reconhecimento de imagem pelo ID

Endpoint: http://localhost:3001/teste/image_recognition/:id

- [GET Func by Facial Id] - Obter informações do funcionário através do ID facial

Endpoint: http://localhost:3001/teste/image_recognition/facial/:id

Rotas de Notificação:

- [POST Notificacao] - <http://localhost:3001/teste/notificacoes>

- [GET All_Notificacoes] - http://localhost:3001/teste/all_notificacoes

- [GET Notificacao by Id] - <http://localhost:3001/teste/notificacoes/:id>

- [PUT Update_Notificacao] - <http://localhost:3001/teste/notificacoes/:id>

- [DELETE Notificacao] - <http://localhost:3001/teste/notificacoes/delete/:id>

Rotas de Camara:

- [POST Camara] - <http://localhost:3001/teste/sentinels/create>

```
{  
  "nome": "Câmera Entrada Principal",  
  "localizacao": "Entrada Principal",  
  "status": "Ativa"  
}
```

- [GET All_Camaras] - <http://localhost:3001/teste/sentinel/all-sentinel>
- [GET Camara by Id] - <http://localhost:3001/teste/sentinel/:id>
- [PUT Update_Camara] - <http://localhost:3001/teste/sentinel/update-sentinel/:id>
- [DELETE Camara] - http://localhost:3001/teste/sentinel/delete_sentinel/:id