



Documentação de APIs: Módulo de Reconhecimento Facial e Biometria

Versão: 1.2.4

Data: 23 de Novembro de 2025

Autor: Manus AI

1. Visão Geral

Este documento detalha todas as APIs do **Módulo de Reconhecimento Facial e Biometria** do Sistema Guardião. Este módulo é responsável por cadastrar, verificar, autenticar e gerenciar a biometria facial dos usuários utilizando a API da Face++.

Base URL: /api/verificacao

🔒 Autenticação

Todas as APIs deste módulo requerem um token JWT válido no cabeçalho de autorização:

```
Authorization: Bearer SEU_TOKEN_AQUI
```

2. Endpoints da API

2.1 Cadastrar ou Atualizar Biometria Facial

Este endpoint cadastra uma nova biometria facial para um usuário ou atualiza uma existente.

- **Endpoint:** POST /verificacao/usuario/:id/biometria

- **Descrição:** Envia uma foto em base64 para a API da Face++, que detecta a face e gera um `face_token`. Este token é salvo no banco de dados. O usuário só pode cadastrar a sua própria biometria.

Requisição

```
{  
  "foto_base64": "..."  
}
```

Resposta de Sucesso (201 Created ou 200 OK)

```
{  
  "success": true,  
  "message": "Biometria facial cadastrada com sucesso",  
  "data": {  
    "id": 1, // ID do registro no banco  
    "face_token": "a1b2c3d4e5f67890...",  
    "face_rectangle": {  
      "top": 150,  
      "left": 100,  
      "width": 200,  
      "height": 200  
    }  
  },  
  "timestamp": "2025-11-23T21:00:00.000Z"  
}
```

Respostas de Erro ✨ MELHORADO

Código HTTP	Status	Descrição
400	Validation Error	<code>foto_base64</code> não fornecida.
403	Forbidden	Tentativa de cadastrar biometria para outro usuário.
404	Not Found	Usuário com o <code>id</code> fornecido não encontrado.
500	Internal Server Error	Nenhuma face detectada na imagem: A imagem não contém um rosto claro.
500	Internal Server Error	Muitas requisições simultâneas: Limite da API Face++ excedido. Tente novamente em alguns segundos.
500	Internal Server Error	Formato de imagem não suportado: Use JPG ou PNG.
500	Internal Server Error	Tamanho da imagem inválido: Use imagens entre 48x48 e 4096x4096 pixels.

2.2 Autenticar por Biometria Facial

Este endpoint autentica um usuário comparando uma nova foto com a biometria já cadastrada.

- **Endpoint:** `POST /verificacao/usuario/:id/biometria/autenticar`
- **Descrição:** Compara a face da nova foto com o `face_token` armazenado. Se a confiança for maior ou igual a 70%, a autenticação é bem-sucedida.

Requisição

```
{  
  "foto_base64": "..."  
}
```

Funcionalidade Adicional: Notificação por E-mail

- **Se Sucesso:** Um e-mail de “Login Bem-Sucedido” é enviado ao usuário com detalhes do acesso (IP, dispositivo).
- **Se Falha:** Um e-mail de “Tentativa de Login Falhou” é enviado, alertando o usuário sobre a atividade suspeita.

Resposta de Sucesso (200 OK)

```
{  
    "success": true,  
    "message": "Autenticação facial bem-sucedida",  
    "data": {  
        "autenticado": true,  
        "confianca": 98.765,  
        "usuario_id": 1  
    },  
    "timestamp": "2025-11-23T21:05:00.000Z"  
}
```

Resposta de Falha na Autenticação (401 Unauthorized)

```
{  
    "success": false,  
    "message": "Face não corresponde ao usuário cadastrado",  
    "timestamp": "2025-11-23T21:06:00.000Z"  
}
```

Respostas de Erro ✨ MELHORADO

Código HTTP	Status	Descrição
400	Validation Error	<code>foto_base64</code> não fornecida.
404	Not Found	Biometria facial não cadastrada para este usuário.
500	Internal Server Error	Nenhuma face detectada na imagem: A imagem enviada para autenticação não contém um rosto claro.
500	Internal Server Error	Muitas requisições simultâneas: Limite da API Face++ excedido. Tente novamente em alguns segundos.
500	Internal Server Error	Token de face inválido ou expirado: A biometria cadastrada pode estar corrompida. Cadastre novamente.

2.3 Verificar Biometria Facial

Verifica se um usuário possui biometria facial cadastrada.

- **Endpoint:** `GET /verificacao/usuario/:id/biometria`
- **Descrição:** Retorna o status do cadastro biométrico de um usuário.

Resposta de Sucesso (200 OK) - Biometria Cadastrada

```
{  
    "success": true,  
    "message": "Operação realizada com sucesso",  
    "data": {  
        "tem_biometria_facial": true,  
        "data_cadastro": "2025-11-23T21:00:00.000Z",  
        "data_atualizacao": "2025-11-23T21:00:00.000Z"  
    },  
    "timestamp": "2025-11-23T21:10:00.000Z"  
}
```

Resposta de Sucesso (200 OK) - Biometria Não Cadastrada

```
{  
    "success": true,  
    "message": "Operação realizada com sucesso",  
    "data": {  
        "tem_biometria_facial": false,  
        "data_cadastro": null,  
        "data_atualizacao": null  
    },  
    "timestamp": "2025-11-23T21:11:00.000Z"  
}
```

2.4 Remover Biometria Facial

Remove o cadastro de biometria facial de um usuário.

- **Endpoint:** DELETE /verificacao/usuario/:id/biometria
- **Descrição:** Apaga o registro de biometria do banco de dados. O usuário só pode remover a sua própria biometria.

Resposta de Sucesso (200 OK)

```
{  
    "success": true,  
    "message": "Biometria facial removida com sucesso",  
    "data": null,  
    "timestamp": "2025-11-23T21:15:00.000Z"  
}
```

Respostas de Erro

Código HTTP	Status	Descrição
403	Forbidden	Tentativa de remover biometria de outro usuário.
404	Not Found	Biometria não encontrada para este usuário.

2.5 Comparar Duas Fotos

Compara duas fotos e retorna a confiança de que pertencem à mesma pessoa.

- **Endpoint:** POST /verificacao/comparar-fotos
- **Descrição:** Útil para verificações manuais ou para comparar a foto de um documento com uma selfie.

Requisição

```
{  
  "foto_base64_1": "... ",  
  "foto_base64_2": "... "  
}
```

Resposta de Sucesso (200 OK)

```
{  
    "success": true,  
    "message": "As faces correspondem à mesma pessoa",  
    "data": {  
        "mesma_pessoa": true,  
        "confianca": 95.123,  
        "thresholds": {  
            "1e-3": 62.354,  
            "1e-4": 69.432,  
            "1e-5": 75.349  
        },  
        "face1": {  
            "face_token": "f1a2b3c4d5...",  
            "face_rectangle": { "top": 150, "left": 100, "width": 200,  
"height": 200 }  
        },  
        "face2": {  
            "face_token": "g1h2i3j4k5...",  
            "face_rectangle": { "top": 160, "left": 110, "width": 190,  
"height": 190 }  
        },  
        "timestamp": "2025-11-23T21:20:00.000Z"  
    }  
}
```

2.6 Status de Verificação do Usuário

Retorna o status completo de verificação de um usuário, incluindo documentos e biometria.

- **Endpoint:** GET /verificacao/usuario/:id/status
- **Descrição:** Agrega informações de várias tabelas para fornecer um resumo do perfil de verificação.

Resposta de Sucesso (200 OK)

```
{  
    "success": true,  
    "message": "Operação realizada com sucesso",  
    "data": {  
        "usuario": {  
            "id": 1,  
            "nome_completo": "Maria Souza",  
            "email": "maria.souza@email.com",  
            "perfil_verificado": true  
        },  
        "documentos": {  
            "total": 3,  
            "pendentes": 0,  
            "aprovados": 3,  
            "rejeitados": 0,  
            "lista": [ /* ... lista de documentos ... */ ]  
        },  
        "biometria_facial": {  
            "cadastrada": true,  
            "data_cadastro": "2025-11-23T21:00:00.000Z"  
        },  
        "pode_usar_sistema": true  
    },  
    "timestamp": "2025-11-23T21:25:00.000Z"  
}
```

2.7 Listar Usuários Verificados

Retorna uma lista paginada de usuários que já possuem o perfil verificado.

- **Endpoint:** GET /verificacao/usuarios/verificados
- **Descrição:** Útil para listar prestadores de serviço ou usuários confiáveis.

2.8 Listar Usuários Pendentes

Lista usuários que possuem documentos com status “Pendente” .

- **Endpoint:** GET /verificacao/usuarios/pendentes

- **Descrição:** Ideal para painéis de administração focados em aprovação de cadastros.
-

4. Estrutura de Dados (Model)

Tabela biometria_usuario

Coluna	Tipo	Descrição
<code>id</code>	INTEGER	Identificador único do registro (Chave Primária).
<code>usuario_id</code>	INTEGER	ID do usuário (Chave Estrangeira e Única).
<code>face_token</code>	VARCHAR(500)	Token da face gerado pela API Face++.
<code>dados_biometria</code>	TEXT	JSON com dados adicionais da detecção facial.
<code>data_cadastro</code>	DATETIME	Data e hora do cadastro da biometria.
<code>data_atualizacao</code>	DATETIME	Data e hora da última atualização da biometria.

|