

# Rehmo API v1 —

---

## Guia de Integração Completo

---

Base URL: /v1 • Contrato v1 congelado (sem breaking changes)

## 0. Visão Executiva

A Rehmo API v1 oferece integração segura (token Bearer), multi-tenant e orientada a eventos clínicos para ingestão, avaliação e monitoramento baseado em dados de wearables. O acesso é controlado por escopos (scopes) e permissão explícita por paciente.

### 0.1 Capacidades-Chave

- Autenticação com token opaco (Bearer) e expiração.
- Permissão explícita por paciente.
- Leitura de dados brutos (RawData) e geração de eventos clínicos.
- Avaliação (evaluate) e resumo (summary) com abordagem defensiva.
- Cálculo de Risk Score com base exclusivamente em eventos persistidos.

## 1. Conceitos Essenciais

### 1.1 Multi-tenant (Isolamento por Empresa)

Cada token pertence a um cliente. O isolamento é obrigatório e aplica-se a todos os dados.

### 1.2 Permissão por Paciente (Acesso Clínico)

Mesmo dentro da mesma empresa, o token só acessa pacientes explicitamente liberados.

### 1.3 Pipeline Clínico

Componentes e fluxo lógico:

- RawData → dados brutos recebidos do wearable (Health Connect e outras fontes).
- Evaluate/Summary → avaliação de regras clínicas e geração de eventos.
- Alerts → persistência dos eventos clínicos.
- RiskScore → score calculado exclusivamente a partir de eventos de alerta.

## 2. Autenticação

### 2.1 Emitir token

Endpoint: POST /auth/token

Headers: Content-Type: application/json

Body (JSON):

```
{ "username": "test_client", "password": "senha" }
```

Resposta (200) inclui:

```
access_token,  
token_type (Bearer),
```

expires\_at e scope (lista de escopos).

## 2.2 Usar token em endpoints protegidos

Enviar em todas as requisições protegidas:

Authorization: Bearer <access\_token>

## 3. Fluxo Recomendado de Integração (Passo a Passo)

Etapa	Endpoint / Requisito
Passo A — Verificar disponibilidade	GET /ping (sem token)
Passo B — Efetuar Autenticação	POST /auth/token
<b>Todos abaixo necessitam do Token</b>	
<b>Authorization: Bearer &lt;access_token&gt;</b>	
Passo C — Descobrir pacientes permitidos	GET /pacientes
Passo D — Buscar alertas persistidos	GET /medical-alerts
Passo E — Avaliar e gerar alertas	POST /medical-alerts/evaluate
Passo F — Histórico de alertas por paciente	GET /medical-alert-history
Passo G — Dados brutos (RawData)	GET /medical-raw
Passo H — Resumo clínico	GET /medical-summary
Passo I — Risk Score	GET /medical-risk-score

## 4. Especificação de Endpoints

### 4.1 GET /ping

Objetivo: verificar se a API está online e retornando informações da versão.

Autenticação não é requerida.

Resposta (200):

```
{
  "success": true,
  "data": {
    "service": "Rehmo API",
    "version": "v1",
    "status": "online",
    "time": "YYYY-MM-DD HH:MM:SS"
  },
  "meta": { "timestamp": "..." }
}
```

## 4.2 POST /auth/token

Objetivo: Autenticar e receber informações da conexão.

Json de entrada :

```
{
  "username": "string (obrigatório)",
  "password": "string (obrigatório)"
}
```

Resposta (200):

```
{
  "success": true,
  "data": {
    "access_token": "string",
    "token_type": "Bearer",
    "expires_at": "YYYY-MM-DD HH:MM:SS",
    "scope": ["patients.read", "scope2"]
  },
  "meta": { "timestamp": "..." }
}
```

## 4.3 GET /pacientes

Escopo: **patients.read**.

Retorno: lista de pacientes liberados para o token (fonte de verdade de acesso).

Campos típicos: id, nome, device, numero\_prontuario.

Autenticação necessária:

**Authorization: Bearer <access\_token>**

**Content-Type: application/json**

Resposta (200):

```
{
```

```
"success": true,
"data": [
  {
    "id": 1,
    "nome": "Paciente",
    "device": "DEVICE_ID",
    "numero_prontuario": "ABC123"
  },
  {
    "id": 2,
    "nome": "Paciente",
    "device": "DEVICE_ID",
    "numero_prontuario": "ABC123"
  }
],
"meta": { "timestamp": "..."}
}
```

#### 4.4 GET /medical-alerts

Escopo: **medical.alerts.read**.

Query params (opcionais): paciente\_id, severity (low|medium|high), resolved (0|1), from, to, page, per\_page.

Retorno: itens + paginação.

Autenticação necessária:

**Authorization:** Bearer <access\_token>

**Content-Type:** application/json

Parâmetros de entrada (opcionais):

Nome do campo	Tipo
paciente_id	Int
severity	low medium high
resolved	0 1
from	datetime
to	datetime
page	Int (default 1)
per_page	int( default 25, máx 100)

Resposta (200):

```
{
  "success": true,
  "data": {
    "items": [
      {
        "id": 15,
        "paciente_id": 17,
        "device": "5519123456789",
        "type": "no_recent_data",
        "severity": "medium",
        "metric": null,
        "value": null,
        "message": "Nenhum dado recebido nos últimos 60 minutos",
        "detected_at": "2026-02-05 10:21:56",
        "resolved": 0,
        "resolved_at": null
      }
    ],
    "pagination": {
      "page": 1,
      "per_page": 25,
      "total": 100
    }
  },
  "meta": { "timestamp": "..." }
}
```

#### 4.5 POST /medical-alerts/evaluate

Escopo: medical.alerts.evaluate.

Body (JSON): device (obrigatório), window\_minutes (default 60), with\_ai (default false).

Retorno: alerts (array) e ai (objeto ou null).

Notas: pode gerar no\_recent\_data se não houver RawData recente; IA é opcional e defensiva.

Autenticação necessária:

**Authorization: Bearer <access\_token>**

**Content-Type: application/json**

Parâmetros de entrada:

```
{
  "device": "string (obrigatório)",
  "window_minutes": 60,
  "with_ai": false,
  "summary": {
    "metrics": ["heartrate", "steps"],
    "with_sleep": false,
    "with_calculated": true
  }
}
```

Resposta (200):

```
{
  "success": true,
  "data": {
    "alerts": [
      {
        "type": "tachycardia",
        "severity": "high",
        "metric": "heartrate",
        "value": "135",
        "message": "Frequência cardíaca elevada",
        "datetime": "YYYY-MM-DD HH:MM:SS",
        "source": "realtime"
      }
    ],
    "ai": null
  },
  "meta": { "timestamp": "..." }
}
```

Se "with\_ai" : true

```
"ai": { "analysis": "resposta da IA", }
```

#### 4.6 GET /medical-alert-history

Escopo: medical.alerts.read.

Query params: paciente\_id (obrigatório), from, to, page (default 1), per\_page (default 20).

Autenticação necessária:

**Authorization:** Bearer <access\_token>

**Content-Type:** application/json

Parâmetros de entrada:

Nome do campo	Tipo
paciente_id (obrigatório)	Int
from	datetime
to	datetime
page	Int (default 1)
per_page	int( default 25, máx 100)

Resposta (200):

```
{
  "success": true,
  "data": {
    "items": [
      {
        "id": 15,
        "company_id": 2,
        "api_client_id": 1001,
        "paciente_id": 17,
        "device": "5519123456789",
        "type": "no_recent_data",
        "severity": "medium",
        "metric": null,
        "value": null,
        "message": "Nenhum dado recebido nos últimos 60 minutos",
        "detected_at": "2026-02-05 10:21:56",
        "resolved": 0,
        "resolved_at": null,
        "resolved_by": null,
        "source": "realtime",
        "status": "active",
        "created_at": "2026-02-05 07:21:56",
        "updated_at": "2026-02-05 07:21:56"
      }
    ],
    "pagination": {
      "page": 1,
      "per_page": 20
    }
  },
  "meta": { "timestamp": "..." }
}
```

#### 4.7 GET /medical-raw

Escopo: medical.raw.read.

Query params: device (obrigatório), from (obrigatório), until (obrigatório), params (ALL ou lista), page, limit (máx 500).

Regras: intervalo máximo 31 dias; métrica inválida em params (retorna erro)

Parâmetros validos: **ALL** (retorna todas as métricas), ou nomes das métricas definido na coluna **Métrica (params)** da tabela do Indicadores válidos no item 5 deste documento.

Retorno: array com received e colunas por métrica (ou data\_json quando params=ALL).

Parâmetros de entrada:

Nome do campo	Tipo
device	string
from	datetime
until	datetime
params	string
page	Int
limit	int( máx 500)

Resposta (200):

```
{
  "success": true,
  "data": [
    {
      "received": "YYYY-MM-DD HH:MM:SS",
      "heartrate": 72,
      "steps": 120
    }
  ],
  "meta": { "timestamp": "..."}
}
```

#### 4.8 GET /medical-summary

Escopo: medical.summary.read.

Query params: device, from, until (obrigatórios). params (opcional): métricas RAW + **sleep** + **calculated**.

**Params** pode ser as métricas conforme tabela do item 5, mais sleep, que trata o sono de forma diferente

Regra defensiva: avalia/persiste alertas somente se calculated existir e não estiver vazio.

Parâmetros de entrada:

Nome do campo	Tipo	Exemplo
device	string	5519123456789
from	datetime	2025-07-01
until	datetime	2025-07-30
params	string	heartRate, steps, oxygenSaturation, sleep, calculated

Se Sleep estiver definido, o retorno contempla em minutos total de sono, média, sessões e estágios .

Se calculated estiver definido ele retorna indicadores de qualidade do sono, risco cardiovascular, mobilidade, índice Geral de Saúde (IGS) e um store de risco à saúde indicado as penalidades.

Resposta (200):

```
{
  "success": true,
  "data": {
    "period": {
      "from": "2025-07-01",
      "until": "2025-07-30",
      "days": 29
    },
    "raw": {
      "heartrate": {
        "min": 40,
        "avg": 67.86,
        "max": 113
      },
      "steps": {
        "min": 41,
        "avg": 1474.08,
        "max": 3717
      },
      "oxygensaturation": {
        "min": 0,
        "avg": 5.65,
        "max": 96
      },
      "sleep": {
        "total_minutes": 3560,
        "avg_per_night": 712,
        "sessions": 5,
        "stages": {
          "light": 2570,
          "deep": 0,
          "rem": 244,
          "awake": 746
        }
      }
    }
  },
  "calculated": {
```

```

"sleep": {
  "quality_score": 32,
  "classification": "ruim"
},
"cardio": {
  "resting_hr": 0,
  "rmsd_avg": null,
  "rmsd_classification": null,
  "risk_level": "low"
},
"mobility": {
  "steps_avg": null,
  "trend": "pending",
  "risk_level": "pending"
},
"igs": {
  "score": 72,
  "classification": "regular"
},
"risk_score": {
  "score": 75,
  "classification": "regular",
  "penalties": [
    "Qualidade do sono ruim"
  ]
}
}
},
"meta": {
  "timestamp": "2026-02-05T17:03:58+00:00"
}
}

```

#### 4.9 GET /medical-risk-score

Escopo: medical.risk.read.

Query params: paciente\_id (obrigatório), window\_hours (default 72; min 1; max 720).

Validações: paciente\_id ausente → 422; window\_hours fora do range → 422; acesso via PacienteAccessService (404/403).

Parâmetros de entrada:

Nome do campo	Tipo
paciente_id	string
window_hours	Int (1 a 720)

Resposta (200):

```
{
  "success": true,
  "data": {
    "paciente_id": 17,
    "window_hours": 72,
    "score": 19.8,
    "level": "high",
    "alerts_considered": 2,
    "details": [
      {
        "alert_id": 2,
        "type": "no_recent_data",
        "severity": "medium",
        "weight": 3,
        "factor": 1,
        "partial_score": 3,
        "detected_at": "2026-02-05 10:21:56"
      },
      {
        "alert_id": 1,
        "type": "no_recent_data",
        "severity": "medium",
        "weight": 3,
        "factor": 0.1,
        "partial_score": 3,
        "detected_at": "2026-02-04 14:31:22"
      }
    ]
  },
  "meta": {
    "timestamp": "2026-02-05T11:10:13+00:00"
  }
}
```

## 5. Indicadores (Métricas) Válidos — Registry Oficial Health Connect

As métricas abaixo são as únicas aceitas para o parâmetro **params** em /medical-raw e para métricas RAW em /medical-summary, conforme HealthConnectMetricService::METRIC\_MAP. As chaves são case-insensitive (normalizadas para lowercase).

**Métrica (params)**

**Descrição**

<b>activecaloriesburned</b>	Active calories burned (kcal)
<b>totalcaloriesburned</b>	Total calories burned (kcal)
<b>distance</b>	Distance (m)
<b>elevationgained</b>	Elevation gained (m)
<b>floorsclimbed</b>	Floors climbed (count)
<b>steps</b>	Steps count
<b>heartrate</b>	Heart rate samples (bpm series)
<b>restingheartrate</b>	Resting heart rate (bpm)
<b>heartratevariabilityrmssd</b>	HRV RMSSD (ms)
<b>oxygensaturation</b>	Oxygen saturation (%)
<b>respiratoryrate</b>	Respiratory rate samples (breaths/min series)
<b>bloodpressure</b>	Blood pressure (systolic/diastolic)
<b>bloodglucose</b>	Blood glucose (unit depends on source)
<b>bodytemperature</b>	Body temperature (°C)
<b>basalbodytemperature</b>	Basal body temperature (°C)
<b>weight</b>	Weight (kg)
<b>height</b>	Height (m)
<b>bodyfat</b>	Body fat (%)
<b>leanbodymass</b>	Lean body mass (kg)
<b>bodywatermass</b>	Body water mass (kg)
<b>bonemass</b>	Bone mass (kg)
<b>basalmetabolicrate</b>	Basal metabolic rate (kcal/day)
<b>sleepsession</b>	Sleep session (complex payload)
<b>exercisesession</b>	Exercise session (complex payload)
<b>nutrition</b>	Nutrition (complex payload)
<b>hydration</b>	Hydration (unit depends on source)
<b>vo2max</b>	VO2 Max (ml/kg/min)

## 5.1 Observações de Consumo

- Métricas complex (ex.: `sleepsession`, `exercisesession`, `nutrition`) não entram em SELEÇÃO direta no `/medical-raw`; são tratadas por serviços quando aplicável.
- Métricas series (ex.: `heartrate`, `respiratoryrate`) podem exigir `params=ALL` para consumo completo, dependendo do modo de extração/retorno do projeto.
- `bloodpressure` é um objeto com `systolic` e `diastolic`.

## 6. Tratamento de Erros e Respostas

Status	Como tratar
401 Unauthorized	Token ausente, inválido/revogado ou expirado. Ação: obter novo token e corrigir header.
403 Forbidden	Escopo insuficiente ou sem permissão explícita de paciente. Ação: solicitar escopo/liberação.
404 Not Found	Paciente inexistente ou fora do tenant. Ação: validar paciente_id e company_id.
422 Validation Error	Parâmetros obrigatórios ausentes, range inválido ou datas inválidas. Ação: corrigir request.
429 Rate limit	Excesso de requisições por token/cliente. Ação: backoff (1-5s), reduzir concorrência, ampliar janelas.

## 7. Recomendações de Integração (Produção)

- Não embutir tokens em apps cliente sem proteção; preferir backend intermediário.
- Implementar rotação de token e expiração planejada.
- Evitar polling agressivo e respeitar rate limits; usar backoff exponencial quando aplicável.
- Para alertas near-real-time, preferir webhook/event bus quando habilitado.
- Logar apenas identificadores (ids) e metadados; nunca registrar tokens.

### 7.1 Segurança e Conformidade (B2B)

- A contratante é responsável por proteger credenciais e tokens, e por garantir que o uso de dados clínicos esteja conforme LGPD.
- A Rehmo mantém isolamento multi-tenant e controles de acesso por escopo e permissão por paciente.
- Mudanças de contrato não ocorrem na v1 (versão congelada); evoluções devem ser publicadas em nova versão (ex.: v2).

## 8. Exemplo de Sequência End-to-End

- 1) GET /ping
- 2) POST /auth/token
- 3) GET /pacientes
- 4) GET /medical-alerts?resolved=0

5. 5) GET /medical-summary?device=...&from=...&until=...&params=heartrate,oxygensaturation,calculated
6. 6) GET /medical-risk-score?paciente\_id=...&>window\_hours=72

## 9. Referências

OpenAPI: /docs/openapi-rehmo-v1.yaml

Postman collection: Rehmo\_API\_v1.postman\_collection.json

Security: SECURITY.md

Ops: OPERATIONS.md