

Documentação da API do Sistema Hospitalar

Gilmarcos Almeida – RU: 4305973

1. Paciente

1.1 POST /pacientes

Objetivo: Cadastrar um novo paciente no sistema.

URL: <http://127.0.0.1:8000/pacientes/>

Parâmetros e estrutura JSON:

```
{
  "nome": "Maria Eduarda",
  "data_nascimento": "1995-08-20",
  "historico_clinico": "Nenhum"
}
```

Resposta esperada (201 Created):

```
{
  "nome": "Maria Eduarda",
  "data_nascimento": "1995-08-20",
  "historico_clinico": "Nenhum",
  "id": 1
}
```

1.2 GET /pacientes

Objetivo: Listar todos os pacientes cadastrados.

URL: <http://127.0.0.1:8000/pacientes/>

Parâmetros: opcionais skip e limit (query params para paginação).

Resposta esperada (200 OK):

```
[
  {
    "nome": "Maria Eduarda",
    "data_nascimento": "1995-08-20",
    "historico_clinico": "Nenhum",
    "id": 1
  },
  {
    "nome": "Gilmarcos",
    "data_nascimento": "1997-04-04",
    "historico_clinico": "Nenhum",
    "id": 2
  }
]
```

1.3 GET /pacientes/{id}

Objetivo: Buscar os dados de um paciente específico.

URL: <http://127.0.0.1:8000/pacientes/2>

Resposta esperada (200 OK):

```
{
  "nome": "Gilmarcos",
  "data_nascimento": "1997-04-04",
  "historico_clinico": "Nenhum",
  "id": 2
}
```

1.4 GET /pacientes/{id}

Objetivo: Atualizar as informações de um paciente.

URL: <http://127.0.0.1:8000/pacientes/1>

Parâmetros e estrutura JSON:

```
{
  "nome": "Maria Eduarda Silva",
  "data_nascimento": "1995-08-20",
  "historico_clinico": "Hipertensão leve"
}
```

Resposta esperada (200 OK):

```
{
  "nome": "Maria Eduarda Silva",
  "data_nascimento": "1995-08-20",
  "historico_clinico": "Hipertensão leve",
  "id": 1
}
```

1.5 DELETE /pacientes/{id}

Objetivo: Remover um paciente do sistema.

URL: <http://127.0.0.1:8000/pacientes/2>

Resposta esperada (200 OK):

```
{
  "nome": "Gilmarcos",
  "data_nascimento": "1997-04-04",
  "historico_clinico": "Nenhum",
  "id": 2
}
```

2. Prontuário

2.1 POST /prontuario

URL: <http://127.0.0.1:8000/prontuario>

Objetivo: Cadastrar um novo prontuário para um paciente.

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "diagnostico": "Hipertensão arterial",
  "receita": "Losartana 50mg, 1x ao dia",
  "observacoes": "Paciente deve retornar em 30 dias."
}
```

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "receita": "Losartana 50mg, 1x ao dia",
  "diagnostico": "Hipertensão arterial",
  "observacoes": "Paciente deve retornar em 30 dias.",
  "id": 1
}
```

2.2 GET /prontuario

URL: <http://127.0.0.1:8000/prontuario>

Objetivo: Listar todos os prontuários cadastrados.

Resposta esperada (HTTP 200):

```
[
  {
    "paciente_id": 1,
    "profissional_id": 2,
    "receita": "Losartana 50mg, 1x ao dia",
    "diagnostico": "Hipertensão arterial",
    "observacoes": "Paciente deve retornar em 30 dias.",
    "id": 1
  }
]
```

2.3 GET /prontuario/{id}

URL: <http://127.0.0.1:8000/prontuario/2>

Objetivo: Obter os dados de um prontuário específico pelo ID.

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 0,
  "profissional_id": 0,
  "receita": "string",
  "diagnostico": "string",
  "observacoes": "string",
  "id": 2
}
```

2.3 PUT /prontuario/{id}

URL: <http://127.0.0.1:8000/prontuario/2>

Objetivo: Atualizar os dados de um prontuário.

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 0,
  "profissional_id": 0,
  "receita": "dipirona",
  "diagnostico": "dor forte",
  "observacoes": "sem alergias"
}
```

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 0,
  "profissional_id": 0,
  "receita": "dipirona",
  "diagnostico": "dor forte",
  "observacoes": "sem alergias",
  "id": 2
}
```

2.5 DELETE /prontuario/{id}

URL: <http://127.0.0.1:8000/prontuario/2>

Objetivo: Excluir um prontuário específico.

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 0,
  "profissional_id": 0,
  "receita": "dipirona",
  "diagnostico": "dor forte",
  "observacoes": "sem alergias",
  "id": 2
}
```

3. Receita

3.1 POST /receita

URL: <http://127.0.0.1:8000/receita>

Objetivo: Cadastrar uma nova receita médica.

Parâmetros e estrutura JSON:

```
{
  "paciente id": 1,
  "profissional id": 2,
  "prontuario id": 3,
  "medicamento": "Amoxicilina 500mg",
  "observacoes": "Tomar após as refeições",
  "dosagem": "3x ao dia por 7 dias",
  "validade": "2025-12-31"
}
```

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "observacoes": "Tomar após as refeições",
  "medicamento": "Amoxicilina 500mg",
  "validade": "2025-12-31",
  "prontuario_id": 3,
  "dosagem": "3x ao dia por 7 dias",
  "id": 1
}
```

3.2 GET /receita

URL: <http://127.0.0.1:8000/receita>

Objetivo: Listar todas as receitas cadastradas.

Resposta esperada (HTTP 200):

```
[
  {
    "paciente_id": 1,
    "profissional_id": 2,
    "observacoes": "Tomar após as refeições",
    "medicamento": "Amoxicilina 500mg",
    "validade": "2025-12-31",
    "prontuario_id": 3,
    "dosagem": "3x ao dia por 7 dias",
    "id": 1
  }
]
```

3.3 GET /receita/{id}

URL: <http://127.0.0.1:8000/receita/1>

Objetivo: Obter os dados de uma receita específica pelo ID.

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "observacoes": "Tomar após as refeições",
  "medicamento": "Amoxicilina 500mg",
  "validade": "2025-12-31",
  "prontuario_id": 3,
  "dosagem": "3x ao dia por 7 dias",
  "id": 1
}
```

3.4 PUT /receita/{id}

URL: <http://127.0.0.1:8000/receita/1>

Objetivo: Atualizar os dados de uma receita.

Parâmetros e estrutura JSON:

```
{
  "medicamento": "Ibuprofeno 600mg",
  "observacoes": "Tomar em caso de dor",
  "dosagem": "2x ao dia",
  "validade": "2026-01-15"
}
```

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "observacoes": "Tomar em caso de dor",
  "medicamento": "Ibuprofeno 600mg",
  "validade": "2026-01-15",
  "prontuario_id": 3,
  "dosagem": "2x ao dia",
  "id": 1
}
```

3.5 DELETE /receita/{id}

URL: <http://127.0.0.1:8000/receita/1>

Objetivo: Excluir uma receita específica.

Resposta esperada (HTTP 200):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "observacoes": "Tomar em caso de dor",
  "medicamento": "Ibuprofeno 600mg",
  "validade": "2026-01-15",
  "prontuario_id": 3,
  "dosagem": "2x ao dia",
  "id": 1
}
```

5. Profissional Saude

5.1 POST /profissionais-saude

Objetivo: Cadastrar um novo profissional da saúde no sistema.

URL: <http://127.0.0.1:8000/profissionais-saude/>

Parâmetros e estrutura JSON:

```
{
  "nome": "Dr. João Silva",
  "especialidade": "Cardiologista"
}
```

Resposta esperada (200 OK):

```
{
  "nome": "Dr. João Silva",
  "especialidade": "Cardiologista",
  "id": 1
}
```

5.2 GET /profissionais-saude

Objetivo: Listar todos os profissionais da saúde cadastrados.

URL: <http://127.0.0.1:8000/profissionais-saude/>

Resposta esperada (200 OK):

```
[
  {
    "nome": "Dr. João Silva",
    "especialidade": "Cardiologista",
    "id": 1
  },
  {
    "nome": "Dr. Carlos Silva",
    "especialidade": "Psiquiatra",
    "id": 2
  }
]
```

5.3 GET /profissionais-saude/{id}

Objetivo: Buscar os dados de um profissional da saúde específico.

URL: <http://127.0.0.1:8000/profissionais-saude/2>

Resposta esperada (200 OK):

```
{
  "nome": "Dr. Carlos Silva",
  "especialidade": "Psiquiatra",
  "id": 2
}
```

5.4 PUT /profissionais-saude/{id}

Objetivo: Atualizar as informações de um profissional da saúde.

URL: <http://127.0.0.1:8000/profissionais-saude/1>

Parâmetros e estrutura JSON:

```
{
  "nome": "Dr. João Silva",
  "especialidade": "Clínico Geral"
}
```


Resposta esperada (200 OK):

```
{
  "nome": "Dr. João Silva",
  "especialidade": "Clínico Geral",
  "id": 1
}
```

5.5 DELETE /profissionais-saude/{id}

Objetivo: Remover um profissional da saúde do sistema.

URL: <http://127.0.0.1:8000/profissionais-saude/1>

Resposta esperada (200 OK):

```
{
  "nome": "Dr. João Silva",
  "especialidade": "Clínico Geral",
  "id": 1
}
```

6. Consulta

6.1 POST /consultas

Objetivo: Agendar uma nova consulta entre um paciente e um profissional de saúde.

URL: <http://127.0.0.1:8000/consultas/>

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Agendada",
  "hora": "14:30:00"
}
```

Resposta esperada (200 OK):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Agendada",
  "hora": "14:30:00",
  "id": 1
}
```

6.2 GET /consultas

Objetivo: Listar todas as consultas agendadas no sistema.

URL: <http://127.0.0.1:8000/consultas/>

Resposta esperada (200 OK):

```
[
  {
    "paciente_id": 1,
    "profissional_id": 2,
    "status": "Agendada",
    "hora": "14:30:00",
    "id": 1
  },
  {
    "paciente_id": 2,
    "profissional_id": 3,
    "status": "Agendada",
    "hora": "16:30:00",
    "id": 2
  }
]
```

6.3 GET /consultas/{id}

Objetivo: Obter os dados de uma consulta específica.

URL: <http://127.0.0.1:8000/consultas/2>

Resposta esperada (200 OK):

```
{
  "paciente_id": 2,
  "profissional_id": 3,
  "status": "Agendada",
  "hora": "16:30:00",
  "id": 2
}
```

6.4 PUT /consultas/{id}

Objetivo: Atualizar o status ou horário de uma consulta.

URL: <http://127.0.0.1:8000/consultas/1>

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Cancelada",
  "hora": "16:00:00"
}
```

Resposta esperada (200 OK):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Cancelada",
  "hora": "16:00:00",
  "id": 1
}
```

6.5 DELETE /consultas/{id}

Objetivo: Cancelar uma consulta e removê-la do sistema.

URL: <http://127.0.0.1:8000/consultas/1>

Resposta esperada (200 OK):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Cancelada",
  "hora": "16:00:00",
  "id": 1
}
```

7. Telemedicina

7.1 POST /telemedicina

Objetivo: Agendar uma nova consulta via telemedicina entre um paciente e um profissional de saúde.

URL: <http://127.0.0.1:8000/telemedicina/>

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 0,
  "profissional_id": 0,
  "status": "Aguardando",
  "horario": "2025-05-13T07:43:56.054944",
  "link_acesso": "string"
}
```

Resposta esperada (200 OK):

```
{
  "paciente_id": 0,
  "profissional_id": 0,
  "status": "Aguardando",
  "horario": "2025-05-13T07:43:56.054944",
  "link_acesso": "string",
  "id": 1
}
```

7.2 GET /telemedicina

Objetivo: Listar todas as sessões de telemedicina cadastradas.

URL: <http://127.0.0.1:8000/telemedicina/>

Resposta esperada (200 OK):

```
[
  {
    "paciente_id": 0,
    "profissional_id": 0,
    "status": "Aguardando",
    "horario": "2025-05-13T07:43:56.054944",
    "link_acesso": "string",
    "id": 1
  }
]
```

7.3 GET /telemedicina/{id}

Objetivo: Consultar os dados de uma sessão de telemedicina específica.

URL: <http://127.0.0.1:8000/telemedicina/1>

Resposta esperada (200 OK):

```
{
  "paciente_id": 0,
  "profissional_id": 0,
  "status": "Aguardando",
  "horario": "2025-05-13T07:43:56.054944",
  "link_acesso": "string",
  "id": 1
}
```

7.4 PUT /telemedicina/{id}

Objetivo: Atualizar o status ou link de acesso da consulta por telemedicina.

URL: <http://127.0.0.1:8000/telemedicina/1>

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Finalizada",
  "horario": "2025-04-25T14:00:00",
  "link_acesso": "https://sistema.telemedicina/consulta123"
}
```

Resposta esperada (200 OK):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Finalizada",
  "horario": "2025-04-25T14:00:00",
  "link_acesso": "https://sistema.telemedicina/consulta123",
  "id": 1
}
```

7.5 DELETE /telemedicina/{id}

Objetivo: Cancelar e remover uma sessão de telemedicina do sistema.

URL: <http://127.0.0.1:8000/telemedicina/1>

Resposta esperada (200 OK):

```
{
  "paciente_id": 1,
  "profissional_id": 2,
  "status": "Finalizada",
  "horario": "2025-04-25T14:00:00",
  "link_acesso": "https://sistema.telemedicina/consulta123",
  "id": 1
}
```

8. Leitos

8.1 POST /leitos

Objetivo: Cadastrar um novo leito hospitalar no sistema.

URL: <http://127.0.0.1:8000/leitos/>

Parâmetros e estrutura JSON:

```
{
  "numero": 0,
  "tipo": "string",
  "status": true
}
```

Resposta esperada (200 OK):

```
{
  "numero": 0,
  "tipo": "string",
  "status": true,
  "id": 0
}
```

8.2 GET /leitos

Objetivo: Listar todos os leitos cadastrados no sistema.

URL: <http://127.0.0.1:8000/leitos/>

Resposta esperada (200 OK):

```
[
  {
    "numero": 0,
    "tipo": "string",
    "status": true,
    "id": 1
  },
  {
    "numero": 10,
    "tipo": "Enfermaria",
    "status": true,
    "id": 2
  }
]
```

8.3 GET /leitos/{id}

Objetivo: Obter os detalhes de um leito específico pelo ID.

URL: <http://127.0.0.1:8000/leitos/1>

Resposta esperada (200 OK):

```
{
  "numero": 0,
  "tipo": "string",
  "status": true,
  "id": 1
}
```

8.4 PUT /leitos/{id}

Objetivo: Atualizar as informações de um leito existente.

URL: <http://127.0.0.1:8000/leitos/1>

Parâmetros e estrutura JSON:

```
{
  "numero": 12,
  "tipo": "enfermaria",
  "status": false
}
```

Resposta esperada (200 OK):

```
{
  "numero": 12,
  "tipo": "enfermaria",
  "status": false,
  "id": 1
}
```

8.5 DELETE /leitos/{id}

Objetivo: Remover um leito do sistema.

URL: <http://127.0.0.1:8000/leitos/1>

Resposta esperada (200 OK):

```
{
  "ok": true
}
```

9. Internações

9.1 POST /internacoes

Objetivo: Cadastrar uma nova internação de paciente.

URL: <http://127.0.0.1:8000/internacoes/>

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 0,
  "leito_id": 0,
  "data_entrada": "2025-05-13",
  "data_saida": "2025-05-13",
  "status": "Ativo",
  "profissional_id": 0,
  "administracao_id": 0
}
```

Resposta esperada (200 OK):

```
{
  "paciente_id": 0,
  "leito_id": 0,
  "data_entrada": "2025-05-13",
  "data_saida": "2025-05-13",
  "status": "Ativo",
  "profissional_id": 0,
  "administracao_id": 0,
  "id": 1
}
```

9.2 GET /internacoes

Objetivo: Listar todas as internações registradas.

URL: <http://127.0.0.1:8000/internacoes/>

Resposta esperada (200 OK):

```
[
  {
    "paciente_id": 0,
    "leito_id": 0,
    "data_entrada": "2025-05-13",
    "data_saida": "2025-05-13",
    "status": "Ativo",
    "profissional_id": 0,
    "administracao_id": 0,
    "id": 1
  }
]
```


9.3 GET /internacoes/{id}

Objetivo: Obter os detalhes de uma internação específica pelo ID.

URL: <http://127.0.0.1:8000/internacoes/1>

Resposta esperada (200 OK):

```
{
  "paciente_id": 0,
  "leito_id": 0,
  "data_entrada": "2025-05-13",
  "data_saida": "2025-05-13",
  "status": "Ativo",
  "profissional_id": 0,
  "administracao_id": 0,
  "id": 1
}
```

9.4 PUT /internacoes/{id}

Objetivo: Atualizar as informações de uma internação (ex: registrar alta médica).

URL: <http://127.0.0.1:8000/internacoes/1>

Parâmetros e estrutura JSON:

```
{
  "paciente_id": 0,
  "leito_id": 0,
  "data_entrada": "2025-05-14",
  "data_saida": "2025-05-14",
  "status": "inativo",
  "profissional_id": 0,
  "administracao_id": 0
}
```

Resposta esperada (200 OK):

```
{
  "paciente_id": 0,
  "leito_id": 0,
  "data_entrada": "2025-05-14",
  "data_saida": "2025-05-14",
  "status": "inativo",
  "profissional_id": 0,
  "administracao_id": 0,
  "id": 1
}
```

9.5 DELETE /internacoes/{id}

Objetivo: Remover uma internação do sistema.

URL: http://127.0.0.1:8000/internacoes/1

Resposta esperada (200 OK):

```
{
  "ok": true
}
```

10. Administração

10.1 POST /administracao

URL da requisição: http://127.0.0.1:8000/administracao

Objetivo: Cadastrar um novo administrador responsável por gerenciar pacientes e internações.

Parâmetros e estrutura JSON:

```
{
  "nome": "Fernanda Souza",
  "cargo": "Coordenadora Administrativa"
}
```

- **Resposta esperada:**

```
{
  "nome": "Fernanda Souza",
  "cargo": "Coordenadora Administrativa",
  "id": 1
}
```

10.2 GET /administracao

URL da requisição: http://127.0.0.1:8000/administracao

Objetivo: Listar todos os administradores cadastrados.

Resposta esperada:

```
[
  {
    "nome": "Fernanda Souza",
    "cargo": "Coordenadora Administrativa",
    "id": 1
  },
  {
    "nome": "Caio Passo",
    "cargo": "Coordenador de Estoque",
    "id": 2
  }
]
```

10.3 GET /administracao/{id}

URL da requisição: http://127.0.0.1:8000/administracao/1

Objetivo: Buscar um administrador específico pelo ID.

Resposta

esperada:

```
{
  "nome": "Fernanda Souza",
  "cargo": "Coordenadora Administrativa",
  "id": 1
}
```

10.4 PUT /administracao/{id}

URL da requisição: http://127.0.0.1:8000/administracao/1

Objetivo: Atualizar os dados de um administrador.

Parâmetros e estrutura JSON:

```
{
  "nome": "Fernanda Souza",
  "cargo": "Diretora Administrativa"
}
```

Resposta esperada:

```
{
  "nome": "Fernanda Souza",
  "cargo": "Diretora Administrativa",
  "id": 1
}
```

10.5 DELETE /administracao/{id}

URL da requisição: http://127.0.0.1:8000/administracao/1

Objetivo: Excluir um administrador do sistema.

Resposta esperada:

```
{
  "ok": true
}
```

11. ProfissionalInternacao

11.1 POST /profissional-internacao

URL: http://127.0.0.1:8000/profissional-internacao

Objetivo: Cadastrar a participação de um profissional de saúde em uma internação.

Parâmetros e estrutura JSON (body):

```
1  {
2    "profissional_id": 0,
3    "internacao_id": 0
4  }
```

Resposta esperada (HTTP 201 Created):

```
1  {
2    "profissional_id": 0,
3    "internacao_id": 0,
4    "id": 2
5  }
```

11.2 GET /profissional-internacao

URL: http://127.0.0.1:8000/profissional-internacao

Objetivo: Listar todas as participações de profissionais em internações.

Parâmetros: nenhum.

Resposta esperada (HTTP 200 OK):

```
1  [  
2    {  
3      "profissional_id": 0,  
4      "internacao_id": 0,  
5      "id": 1  
6    },  
7    {  
8      "profissional_id": 0,  
9      "internacao_id": 0,  
10     "id": 2  
11  },
```

11.3 DELETE /profissional-internacao/{id}

URL: http://127.0.0.1:8000/profissional-internacao/{id}

Objetivo: Remover a associação entre um profissional e uma internação.

Resposta esperada (HTTP 200 OK):

```
{  
  "profissional_id": 0,  
  "internacao_id": 0,  
  "id": 1  
}
```

12.1 POST/ auth/register

URL: http://127.0.0.1:8000/auth/register

Objetivo: criar um novo usuário e senha para acesso a rota restritas

Parâmetros e estrutura JSON (body):

```
{  
  "email": "gilmarcos@example.com",  
  "senha": "teste"  
}
```

Resposta esperada (HTTP 200 OK):

Code	Details
200	<div>Response body</div> <div><pre>{ "id": 4, "email": "gilmarcos@example.com" }</pre></div>

12.1 POST/ auth/login

URL: http://127.0.0.1:8000/auth/login

Objetivo: Faz login com as credencias criadas anteriormente

Parâmetros e estrutura JSON (body):

<pre>{ "email": "gilmarcos@example.com", "senha": "teste" }</pre>

Resposta esperada (HTTP 200 OK):

Code	Details
200	<div>Response body</div> <div><pre>{ "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJnaWxtYXJjb3NAZXhhbXBsZS5jb20iLCJleHAiOjE3NDc2MDk2MDN9.V-phDF4jNwVoN-ZrX_FrodQdOYERpGd999IR-9m1ENp4", "token_type": "bearer" }</pre></div>