**1. Endpoints da API**

**Clientes**

1.1 **GET** - Obter todos os clientes  
**Endpoint**: GET /clientes  
**Descrição**: Recupera a lista de todos os clientes.

json

{

"idClientePk": 1,

"email": "cliente@exemplo.com",

"telefone": "123456789",

"celular": "987654321",

"nome": "João Silva",

"usuario": "joaosilva",

"senha": "12345",

"idFisicoFK": 1,

"idJuridicoFK": null,

"endereco": "Rua Exemplo, 123",

"complemento": "Apto 101",

"cidade": "Cidade Exemplo",

"estado": "SP"

}

1.2 **POST** - Criar novo cliente  
**Endpoint**: POST /clientes  
**Descrição**: Cria um novo cliente.  
**Corpo da requisição**:

json

{

"email": "novo@cliente.com",

"telefone": "123123123",

"celular": "456456456",

"nome": "Novo Cliente",

"usuario": "novocliente",

"senha": "12345",

"idFisicoFK": 1,

"idJuridicoFK": null,

"endereco": "Rua Nova, 456",

"complemento": "Casa",

"cidade": "Cidade Nova",

"estado": "RJ"

}

1.3 **PUT** - Atualizar cliente  
**Endpoint**: PUT /clientes/{id}  
**Descrição**: Atualiza os dados de um cliente existente.  
**Corpo da requisição**:

json

{

"email": "cliente@novomail.com",

"telefone": "111222333",

"celular": "444555666",

"nome": "João Silva Atualizado",

"usuario": "joaosilva\_atualizado",

"senha": "nova\_senha",

"idFisicoFK": 1,

"idJuridicoFK": null,

"endereco": "Rua Atualizada, 789",

"complemento": "Casa 202",

"cidade": "Cidade Atualizada",

"estado": "MG"

}

1.4 **DELETE** - Excluir cliente  
**Endpoint**: DELETE /clientes/{id}  
**Descrição**: Exclui um cliente.

**Pedidos**

2.1 **GET** - Obter todos os pedidos  
**Endpoint**: GET /pedidos  
**Descrição**: Recupera a lista de todos os pedidos.  
**Resposta**:

json

{

"idPedido": 1,

"idSalaFK": 2,

"idClienteFK": 1,

"formpgment": "Cartão",

"dataEntrada": "2025-03-01",

"dataSaida": "2025-03-05"

}

2.2 **POST** - Criar novo pedido  
**Endpoint**: POST /pedidos  
**Descrição**: Cria um novo pedido.  
**Corpo da requisição**:

json

{

"idSalaFK": 3,

"idClienteFK": 2,

"formpgment": "Boleto",

"dataEntrada": "2025-03-10",

"dataSaida": "2025-03-12"

}

2.3 **PUT** - Atualizar pedido  
**Endpoint**: PUT /pedidos/{id}  
**Descrição**: Atualiza os dados de um pedido existente.  
**Corpo da requisição**:

json

{

"idSalaFK": 2,

"idClienteFK": 1,

"formpgment": "Cartão de Crédito",

"dataEntrada": "2025-03-05",

"dataSaida": "2025-03-07"

}

2.4 **DELETE** - Excluir pedido  
**Endpoint**: DELETE /pedidos/{id}  
**Descrição**: Exclui um pedido.

**Pessoa Física**

3.1 **GET** - Obter todas as pessoas físicas  
**Endpoint**: GET /pessoafisicas  
**Descrição**: Recupera a lista de todas as pessoas físicas.  
**Resposta**:

json

{

"idFisicoPk": 1,

"CPF": "123.456.789-00"

}

3.2 **POST** - Criar nova pessoa física  
**Endpoint**: POST /pessoafisicas  
**Descrição**: Cria uma nova pessoa física.  
**Corpo da requisição**:

json

{

"CPF": "123.456.789-00"

}

3.3 **DELETE** - Excluir pessoa física  
**Endpoint**: DELETE /pessoafisicas/{id}  
**Descrição**: Exclui uma pessoa física.

**Pessoa Jurídica**

4.1 **GET** - Obter todas as pessoas jurídicas  
**Endpoint**: GET /pessoajuridicas  
**Descrição**: Recupera a lista de todas as pessoas jurídicas.  
**Resposta**:

json

{

"idJuridicoPk": 1,

"cnpj": "12.345.678/0001-99"

}

4.2 **POST** - Criar nova pessoa jurídica  
**Endpoint**: POST /pessoajuridicas  
**Descrição**: Cria uma nova pessoa jurídica.  
**Corpo da requisição**:

json

{

"cnpj": "12.345.678/0001-99"

}

4.3 **DELETE** - Excluir pessoa jurídica  
**Endpoint**: DELETE /pessoajuridicas/{id}  
**Descrição**: Exclui uma pessoa jurídica.

**Salas**

5.1 **GET** - Obter todas as salas  
**Endpoint**: GET /salas  
**Descrição**: Recupera a lista de todas as salas.  
**Resposta**:

json

{

"idSalaPk": 1,

"tipoSala": "Sala de Reunião",

"capacPessoas": 10,

"precoBase": 150.00,

"nome\_sala": "Reunião Executiva",

"descricao\_sala": "Sala com recursos avançados para reuniões",

"foto\_sala": "imagem\_reuniao.jpg"

}

5.2 **POST** - Criar nova sala  
**Endpoint**: POST /salas  
**Descrição**: Cria uma nova sala.  
**Corpo da requisição**:

json

Copiar

{

"tipoSala": "Sala de Conferências",

"capacPessoas": 20,

"precoBase": 500.00,

"nome\_sala": "Conferência Global",

"descricao\_sala": "Espaço para grandes eventos",

"foto\_sala": "imagem\_conferencia.jpg"

}

5.3 **DELETE** - Excluir sala  
**Endpoint**: DELETE /salas/{id}  
**Descrição**: Exclui uma sala.

**2. Melhorias e Recomendações**

**2.1 Validação de Dados**

Verifique se os dados enviados nas requisições (como CPF, CNPJ e e-mail) estão corretos e no formato adequado antes de salvar ou atualizar registros.

**2.2 Autenticação e Autorização**

Utilize tokens de autenticação , para garantir que apenas usuários autenticados possam acessar as rotas protegidas.

**2.3 Códigos de Status HTTP**

Certifique-se de retornar os códigos de status corretos para cada operação:

* **200 OK**: Sucesso em GET ou PUT.
* **201 Created**: Sucesso em POST.
* **404 Not Found**: Recurso não encontrado.
* **400 Bad Request**: Erros de validação.
* **500 Internal Server Error**: Falhas no servidor.

**2.4 Paginação**

Para consultas como GET /clientes ou GET /pedidos, considere implementar a paginação:

GET /clientes

**2.5 Relacionamento entre Entidades**

Um cliente pode ter Pessoa Física ou Pessoa Jurídica. Utilize chaves estrangeiras (idFisicoFK ou idJuridicoFK) para modelar esse relacionamento.

**2.6 Tratamento de Erros**

Forneça mensagens de erro detalhadas no backend para ajudar os desenvolvedores a identificar problemas durante a integração.