# API - Manutenção Tabelas SQL SERVER

# (CRUD Simples com JSON) POSTS (Entrada e Saída JSON)

# 

1. **Parte superior do formulário**

**1 - Objetivo**O objetivo principal deste recurso técnico é acessar **qualquer** **tabela** de um banco de dados **SQL** **Server** para **consultar, incluir, alterar ou excluir registros** através de apenas **1 EndPoint** e **1 Post (JSON).** O **EndPoint** sempre retorna o resultado da operação solicitada através um **JSON**.

Este recurso técnico pode ser utilizado em qualquer sistema, para qualquer banco de dados SQL Server e para qualquer tabela que se tenha criado no exato momento.

**Exemplos:**

* FontEnd para consulta e manutenção de cadastros com botões para chamada conforme a operação solicitada.
* FrontEnd com botões para consulta de tabelas domínio em telas de movimentações diversas.

**2 - Estrutura e Variação do EndPoint (POST)**

* **Estrutura**O exemplo utilizado no EndPoint é baseado na minha máquina localhost (<http://localhost:5000/>) e a estrutura fixa é [ADRRBR/cadastros/manutencao/tabelas/](http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/TABELA/CON)
* **Variação**

**Consulta**

[http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/**TABELA**/**CON**](http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/TABELA/CON) **Post (JSON) – Colunas, Ordem e Condições para Consulta**{

  "COLUNAS": "Coluna1,Coluna2,Coluna3",

  "ORDEM": "Coluna1",

  "CONDICOES": "Coluna1 > 0"

}

**Manutenção (Inclusão/Alteração/Exclusão)**

[http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/**TABELA**/**INC**](http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/TABELA/CON)[http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/**TABELA**/**ALT**](http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/TABELA/CON)[http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/**TABELA**/**EXC**](http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/TABELA/CON) **Post (JSON) – Colunas e Valores da Tabela para Manutenção**{

  "CHAVE\_PRIMARIA": Conteudo,

  "COLUNA1":  Conteudo,

  "COLUNA2": "Conteudo",

  "COLUNA3": "Conteudo"

}

**3 – Exemplo da utilização do EndPoint em uma INCLUSÃO na tabela CLIENTES**

<http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/clientes/inc>  
  
**Post (JSON) – Colunas e Valores da Tabela para Manutenção**{

  "PK\_CLIENTE": 0,

  "CODIGO": "100",

  "NOME": "GUILHERME RIBEIRO",

  "DATA\_INCLUSAO": "2024-07-18"

}

**Resultado (JSON)**[

    {

        "Mensagem:": "Registro inserido com sucesso!",

        "Quantidade:": 1

    },

    {

        "Registros:": [

            {

                "PK\_CLIENTE": 1,

                "CODIGO": "100",

                "NOME": "GUILHERME RIBEIRO",

                "DATA\_INCLUSAO": "2024-07-18 10:44:51.900000"

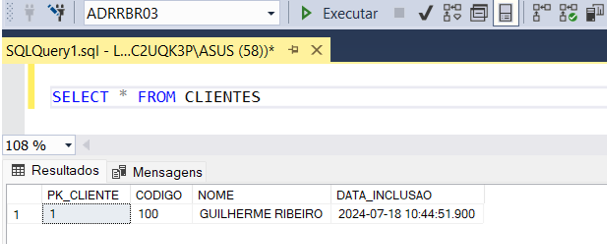
            }

        ]

    }

]

**Tabela SQL Server**

****

**4 – Exemplo da utilização do EndPoint em uma CONSULTA na tabela CLIENTES**

http://localhost:5000/ADRRBR/cadastros/manutencao/tabelas/clientes/con  
  
**Post (JSON) – Colunas, Ordem e Condições para Consulta contendo condição EXISTENTE**{

  "colunas": "CODIGO,nome",

  "ordem": "nome",

  "condicoes": "CODIGO = 100"

}

**Resultado (JSON)**[

    {

        "Mensagem:": "Consulta efetuada com sucesso",

        "Quantidade:": 1

    },

    {

        "Registros:": [

            {

                "CODIGO": "100",

                "nome": "GUILHERME RIBEIRO"

            }

        ]

    }

]

**Post (JSON) – Colunas, Ordem e Condições para Consulta contendo condição INEXISTENTE**

{

  "colunas": "CODIGO,nome",

  "ordem": "nome",

  "condicoes": "CODIGO <> 100"

}

**Resultado (JSON)**[

    {

        "Mensagem:": "Consulta efetuada com sucesso",

        "Quantidade:": 0

    },

    {

        "Registros:": []

    }

]

**5 – Observações Importantes**

* Este recurso técnico considera a premissa de chave primária única na tabela a ser consumida pelo EndPoint;
* Todos os índices exclusivos são validados e retornam mensagem de violação tratada;
* As chaves estrangeiras são validadas na integridade estabelecida no banco de dados;
* O EndPoint foi desenvolvido em Python, distribuído em Classes simplificadas e utiliza Procedimentos no SQL Server para a integração entre ambos;
* Todo o conteúdo deste recurso foi desenvolvido por Adriano Ribeiro.