

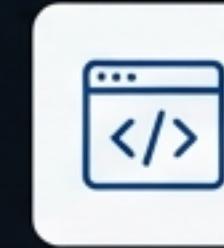
De Dados a Decisões: A Jornada da Integração Web com Business Intelligence

Como conectar um banco de dados na nuvem a um frontend web, passando pela criação de dashboards interativos com Power BI.



Power BI:

Mais de 25 milhões de usuários corporativos ativos.



APIs REST:

Movimentam mais de 80% do tráfego da web moderna.



Supabase:

Usado por mais de 500.000 desenvolvedores.

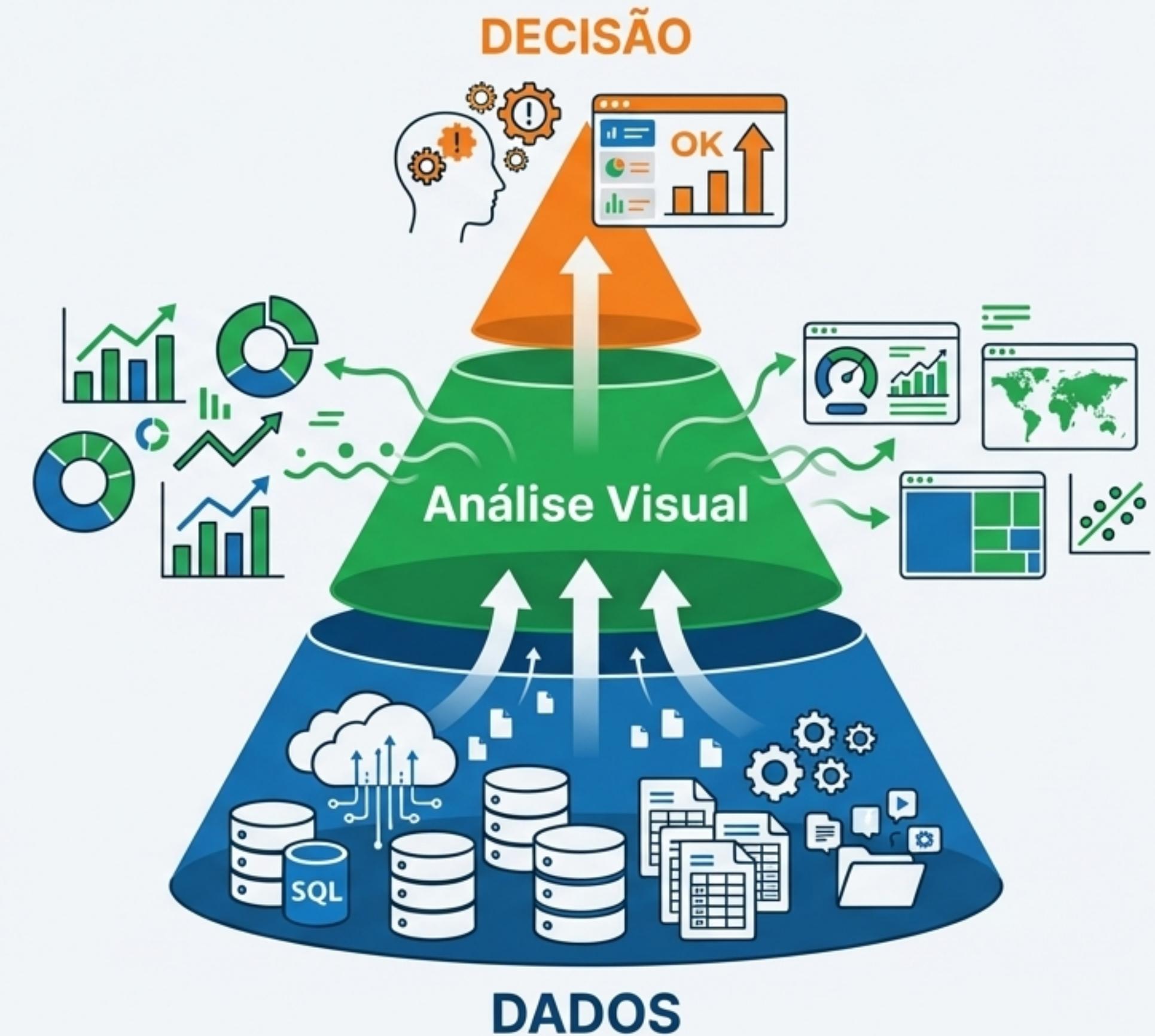
Fontes: Microsoft, Supabase. Dados de 2024.

O Que é Business Intelligence? A Análise Visual de Dados

BI organiza, analisa e compartilha informações para suportar a gestão. Ferramentas de BI operam no 'Nível de Visão', onde os dados já foram filtrados, agregados e relacionados para gerar insights. O objetivo é transformar o que aconteceu (Análise Descritiva) e por que aconteceu (Análise Diagnóstica) em decisões informadas.



O BI não é apenas sobre gráficos, mas sobre o caminho completo do armazenamento (SGBD) até a estratégia de negócio.



A Oficina de Dados: Do Banco ao Dashboard



1. Extração

Extração: O Power BI se conecta ao PostgreSQL (local ou na nuvem).



2. SQL

SQL: Consultas com `SELECT`, `JOIN` e agregações (`SUM`, `AVG`, `COUNT`) atuam como a ponte para extrair os dados.

ID	Nome	Vendas	Data
1	Nome	250.00	01/07/2022
2	Nome	230.00	00/07/2022
3	Nome	300.00	00/07/2023
4	Nome	200.00	00/07/2023
5	Nome	100.00	00/07/2023
6	Nome	300.00	00/07/2023

3. Tabela

Tabela: Os resultados da consulta formam uma tabela estruturada.



4. Dashboard

Dashboard: A tabela alimenta os indicadores visuais e gráficos interativos no Power BI.



A linguagem SQL é a ponte fundamental para transformar registros brutos em indicadores visuais claros.

Como o Power BI Acessa Seus Dados: Modos de Conexão

A escolha do modo de conexão depende do volume de dados, necessidade de atualização em tempo real e requisitos de governança.

Import

Características

Copia os dados para o modelo do Power BI. Desempenho alto e recursos completos de modelagem.

Quando usar

Datasets moderados que necessitam de atualização periódica.

DirectQuery

Características

Consulta diretamente a fonte de dados. Dados sempre atualizados, mas com algumas limitações de transformação.

Quando usar

Grandes volumes de dados e requisitos de análise em tempo quase real.

Live Connection

Características

Conecta-se a modelos semânticos externos (ex: Analysis Services) sem importação local.

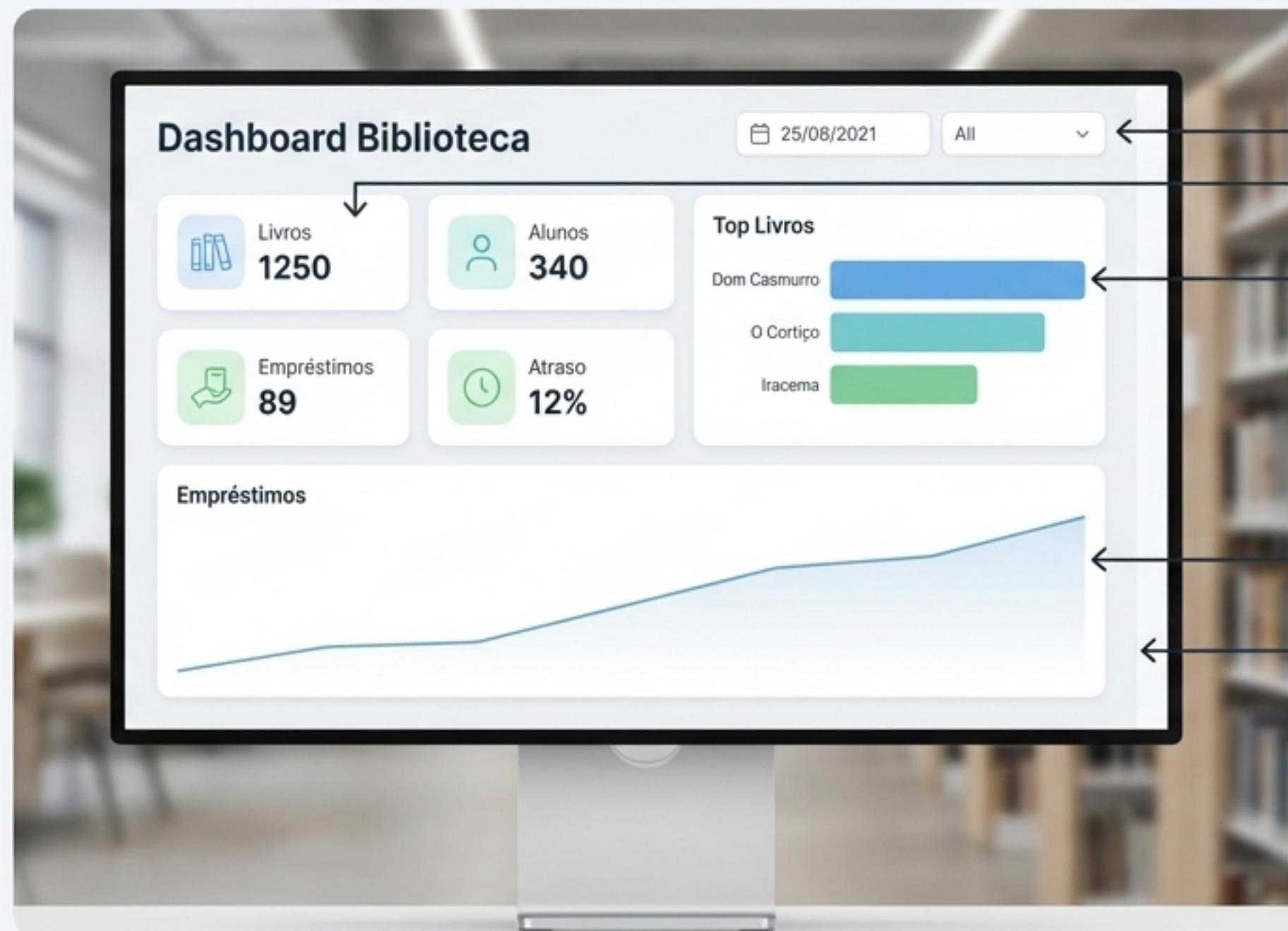
Quando usar

Governança centralizada e compartilhamento de modelos de dados corporativos.

Fonte: Microsoft Learn.

O Painel de Controle: Materializando a Visão dos Dados

Um dashboard é um painel interativo que consolida múltiplas visões de um mesmo conjunto de dados, permitindo monitorar e explorar resultados em tempo real.



Cartão (Card): Exibe um valor único (KPI), como "Total de Empréstimos".
Origem SQL: `SUM()`, `COUNT()`.

Gráfico de Barras: Compara volumes, como "Top Livros".
Origem SQL: `GROUP BY`.

Gráfico de Linha: Mostra a distribuição ao longo do tempo.
Origem SQL: `Agregação por período`.

Filtro: Permite selecionar subconjuntos de dados.
Origem SQL: Dimensões da cláusula `WHERE`.



Otimização com Views: Use `CREATE VIEW` para salvar consultas complexas como tabelas virtuais, simplificando as requisições do BI.

A Plataforma de Lançamento: Publicando o Banco na Nuvem com Supabase

DBaaS (Database as a Service) oferece um banco de dados gerenciado na nuvem, cuidando de escalabilidade, segurança e disponibilidade.

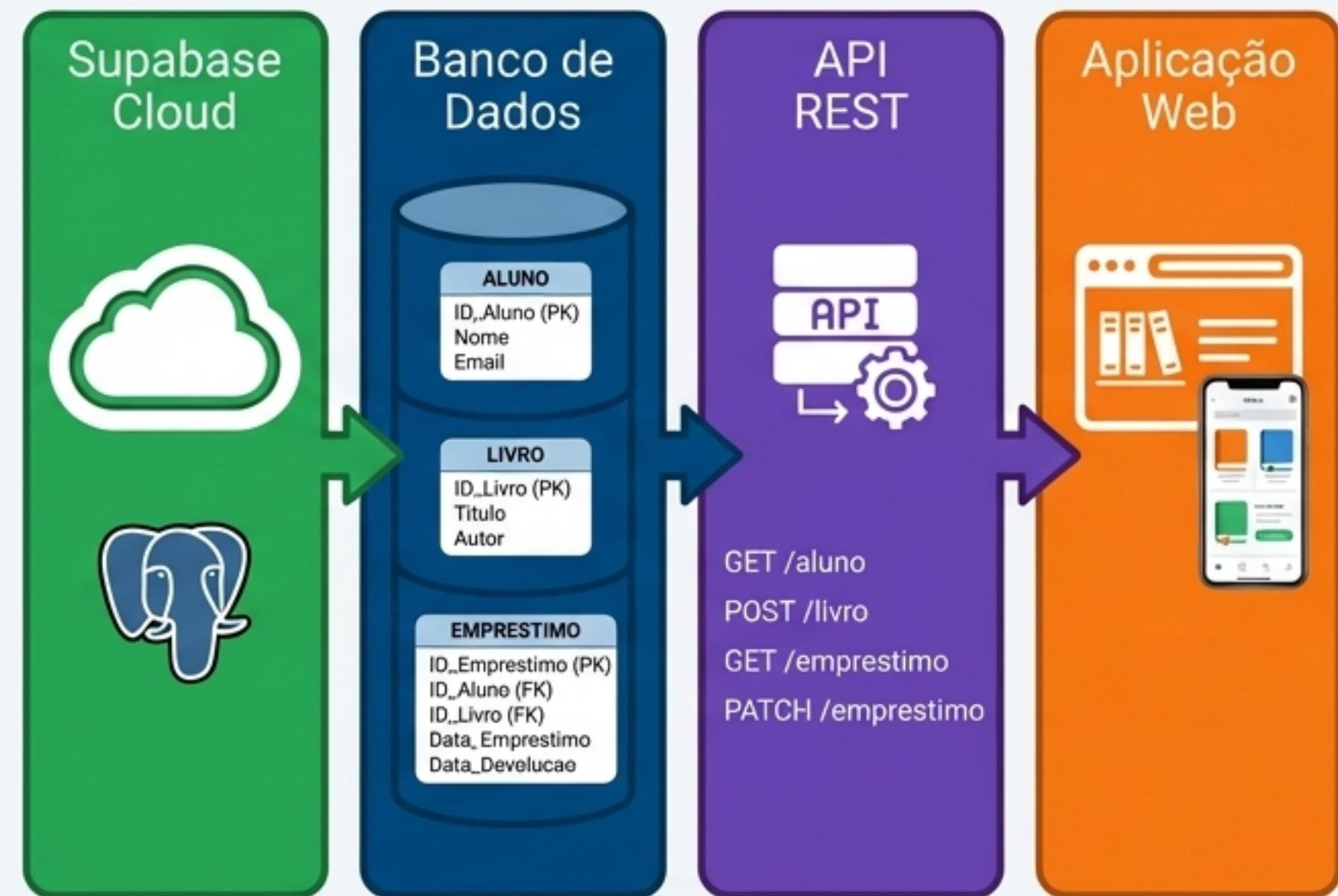
Apresentando Supabase

- Banco de dados gerenciado.
- API REST automática.
- Autenticação integrada.

Uma solução open-source baseada em PostgreSQL.

1. **Supabase Cloud**: O ambiente que hospeda o banco.

3. **API REST**: A camada que expõe os dados via endpoints HTTP.



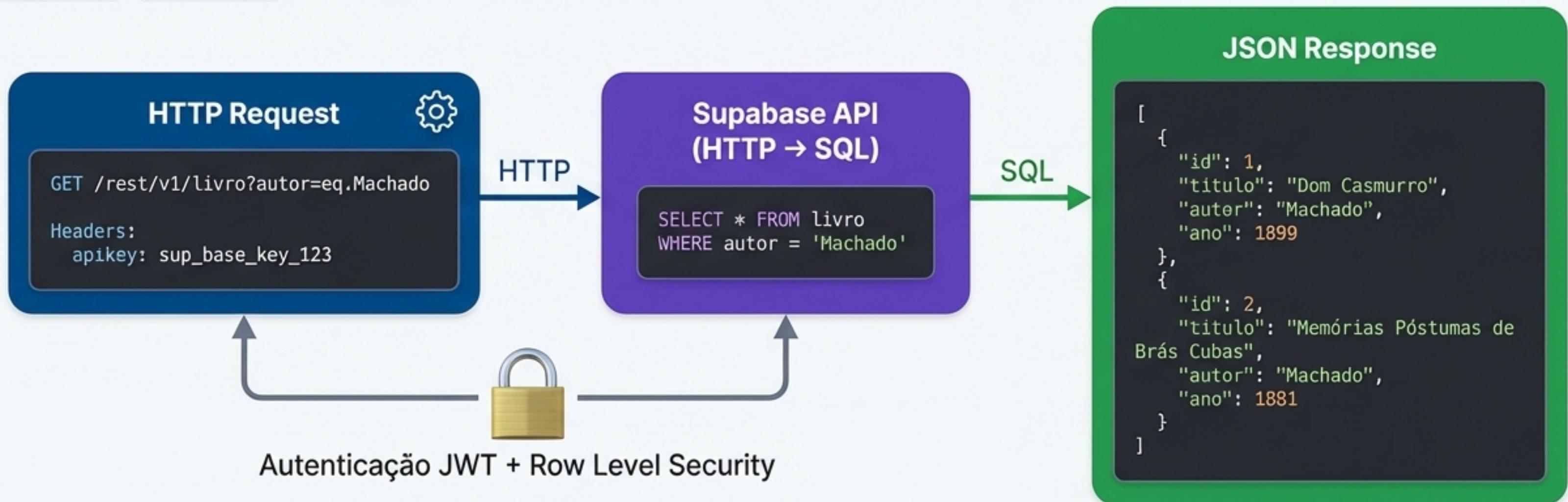
2. **Banco de Dados**: O PostgreSQL com nosso esquema.

3. **API REST**: A camada que expõe os dados via endpoints HTTP.

4. **Aplicação Web**: O cliente que consome os dados.

O Portal de Acesso: Traduzindo Requisições HTTP em Consultas SQL

O Supabase usa **PostgREST**, uma ferramenta que cria uma API RESTful a partir de um esquema PostgreSQL. Ele traduz requisições `GET`, `POST`, `PATCH` diretamente em comandos `SELECT`, `INSERT`, `UPDATE`.



****Destaque em Segurança**:** Acesso protegido por autenticação (JWT) e regras de segurança a nível de linha (Row Level Security).

Consultando a API: Filtros e Operadores Essenciais

Os dados podem ser filtrados diretamente na URL da requisição usando operadores do PostgREST.

Operador	Exemplo	Descrição
eq	?cpf=eq.12345678900	Igualdade
ilike	?nome=ilike.*silva*	Busca textual
gt / gte	?total=gt.100	Maior que
lt / lte	?quantidade=lte.10	Menor que
in	?status=in."novo", "pago"	Pertence ao conjunto
is	?data_entrega=is.null	Nulidade/Booleano

Exemplo Prático

Listar todos os exemplares com status 'disponível'.

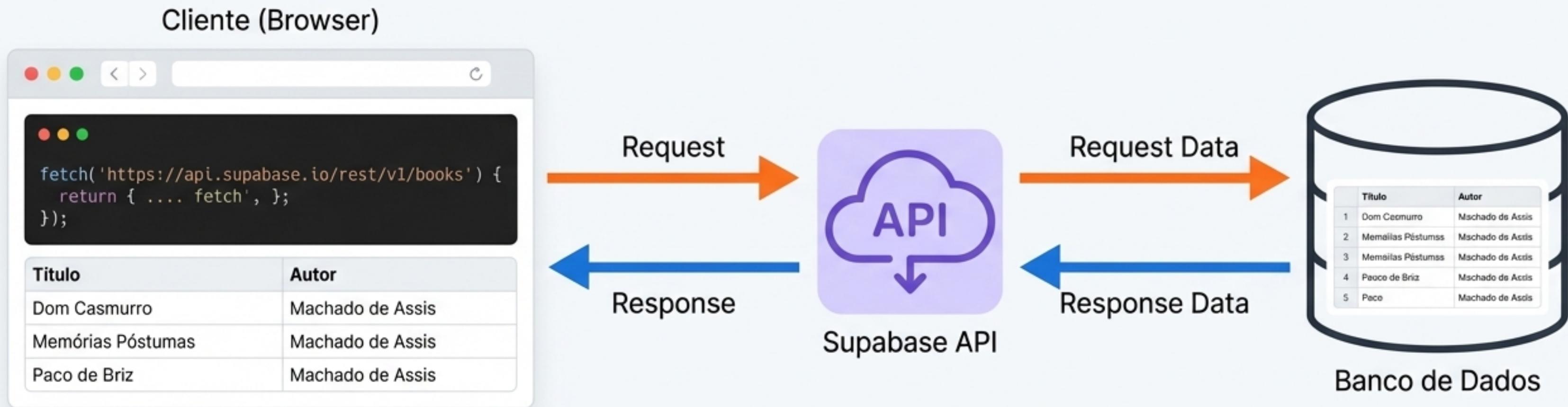
GET

<https://<projeto>.supabase.co/rest/v1/exemplar>?status=eq.
disponivel



Use este slide como um "cheatsheet" para construir suas requisições.

A Experiência Final: Construindo um Frontend com a API



1. O código JavaScript no navegador faz uma requisição (`fetch`) para a API do Supabase.
2. A API do Supabase processa o `Request` e consulta o banco de dados.
3. O banco retorna os `Response Data`.
4. A API envia a `Response` de volta para o cliente no formato JSON.
5. O JavaScript renderiza os dados em uma tabela HTML na página.

Este fluxo demonstra a arquitetura de três camadas em ação: Apresentação (Frontend), Negócio (API) e Dados (Banco).

Código em Ação: Listando Alunos da Biblioteca

Uma página simples em HTML e JavaScript que consome a API REST para exibir uma lista de alunos.

```
// Função para carregar os dados
async function carregarAlunos() {
    // URL do endpoint da API do Supabase
    const url = "https://<projeto>.supabase.co/rest/v1/aluno?select=matricula,nome";

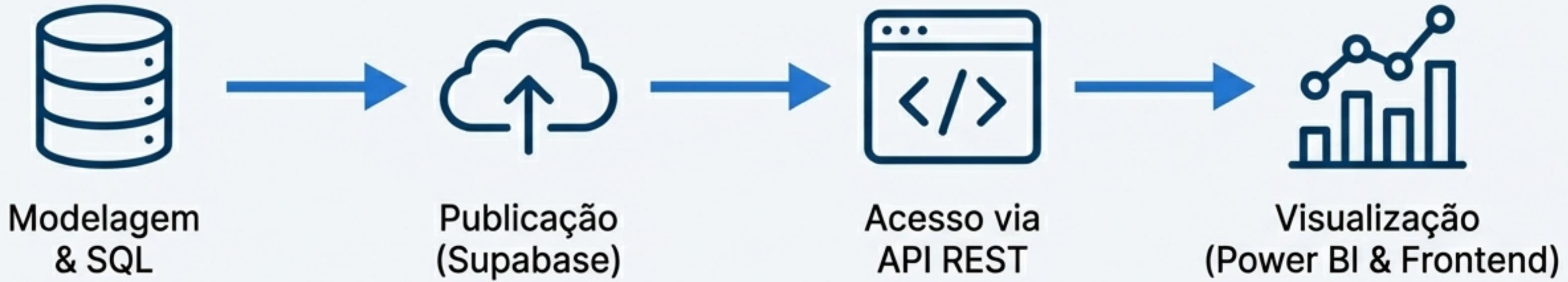
    // Requisição com fetch, passando a chave de API no cabeçalho
    const res = await fetch(url, {
        headers: { apikey: "<chave-anon-pública>" }
    });
    const alunos = await res.json(); ←

    // Injeta os dados na tabela HTML
    const tabela = document.getElementById("tabelaAlunos");
    alunos.forEach(a => {
        tabela.innerHTML += `<tr><td>${a.matricula}</td><td>${a.nome}</td></tr>`;
    });
}

carregarAlunos();
```

O método `fetch` do JavaScript é usado para fazer a requisição HTTP. A resposta (`res.json()`) é um array de objetos que pode ser facilmente iterado para construir a interface do usuário.

A Jornada Completa: Do Modelo à Visualização



Este capítulo conectou o banco de dados à web e às ferramentas de BI, fechando o ciclo completo: Modelagem → SQL → Otimização → Publicação → Visualização.

Business Intelligence: Transforma dados em insights visuais.

Power BI: Ferramenta para criar dashboards interativos.

DBaaS (Supabase): Banco de dados gerenciado na nuvem com API automática.

API REST: Interface web para acessar e manipular dados com segurança.

Frontend: Camada de apresentação que consome a API para o usuário final.

Agora é Sua Vez: Estudo de Caso Prático

Aplique os conceitos aprendidos para desenvolver seu próprio painel de análise e aprimorar a aplicação web.

Desafio Parte 1: Power BI

- Conecte o Power BI ao seu banco Supabase.
- Crie um dashboard mostrando:
 - Total de empréstimos por mês.
 - Top 5 livros mais emprestados.
 - Alunos com mais avaliações registradas.
 - Taxa de reservas concluídas (%)



Desafio Parte 2: Frontend Web

- Personalize a página HTML para mostrar uma lista de livros disponíveis.
- Adicione um botão 'Reservar' (a funcionalidade pode ser um mockup).