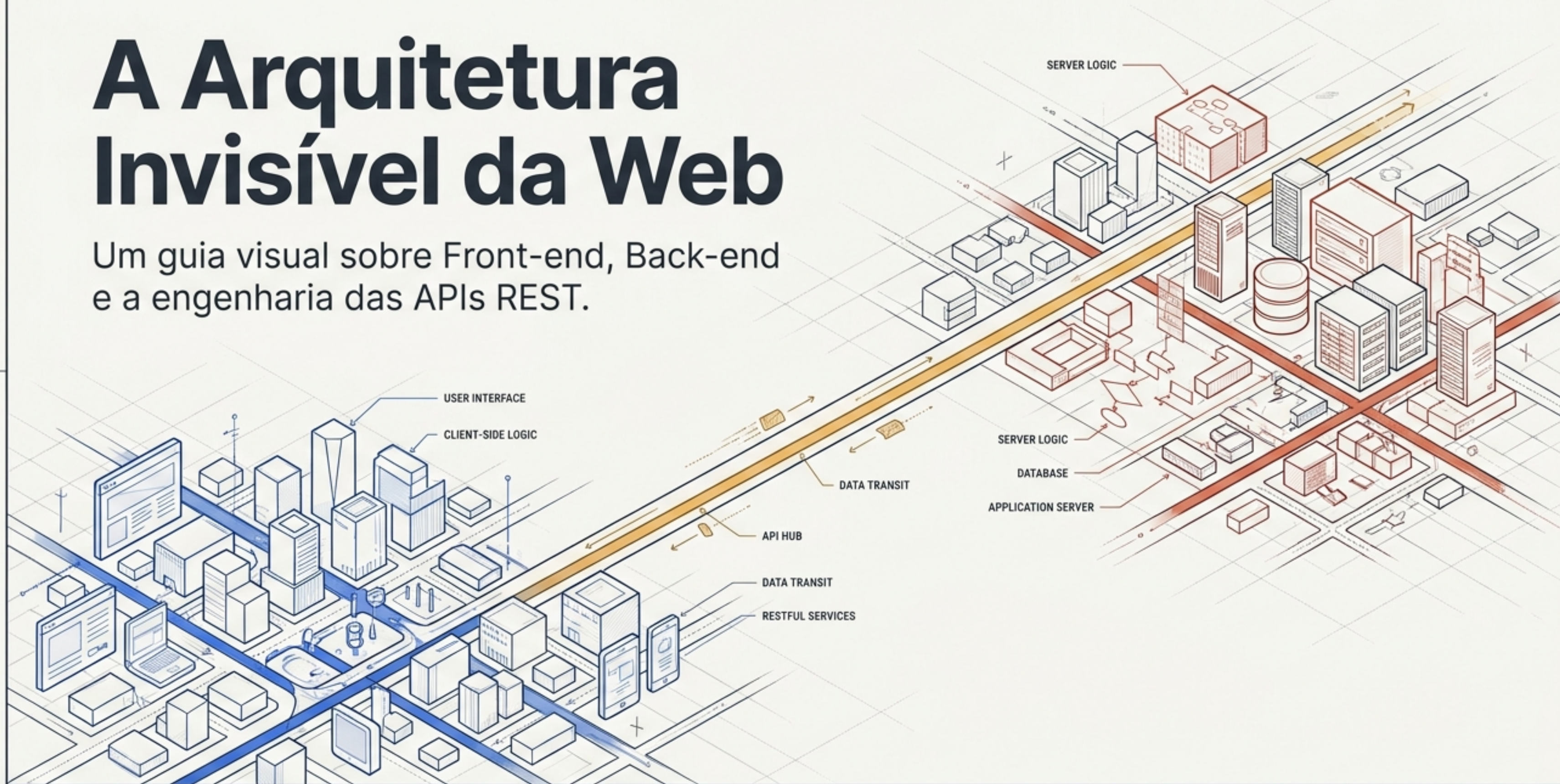
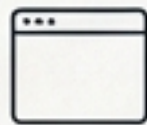


A Arquitetura Invisível da Web

Um guia visual sobre Front-end, Back-end e a engenharia das APIs REST.

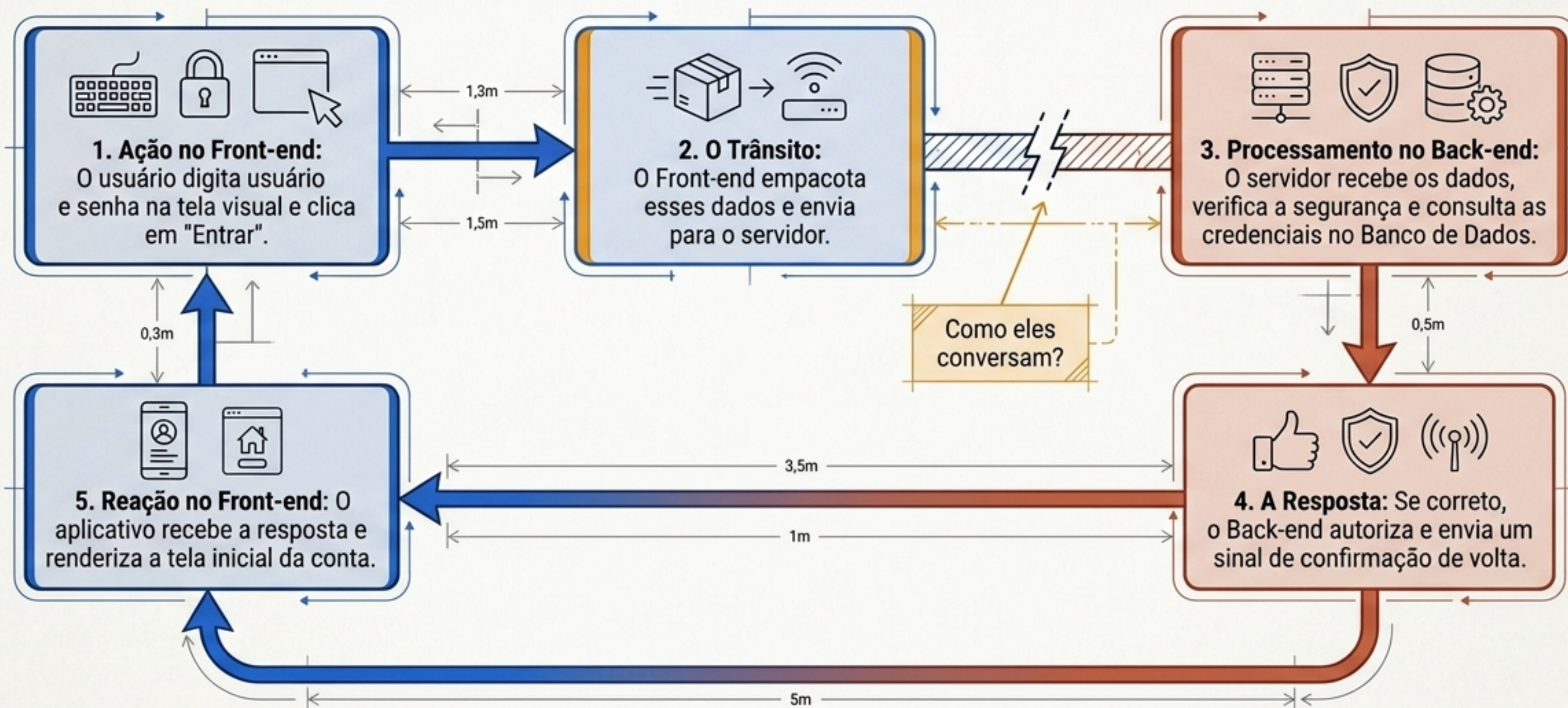


A Divisão do Sistema: Cliente e Servidor

	Front-end (O Cliente)	Back-end (O Servidor)
A Metáfora	 O Salão do Restaurante (Onde o cliente senta, lê o menu e interage)	 A Cozinha (Onde o pedido é processado e preparado)
Responsabilidade Principal	Interface visual, experiência do usuário e captura de ações. → Processamento de dados, regras de negócio, segurança e autenticação.	
Ambiente de Execução	 Navegadores Web, Aplicativos iOS/Android. 	 Servidores e Bancos de Dados. 
Tecnologias Chave	HTML, CSS, JavaScript, React, Angular, Vue.	Java, Node.js, Python, PHP, C#, Go.

O Ciclo de Vida de uma Ação Cotidiana

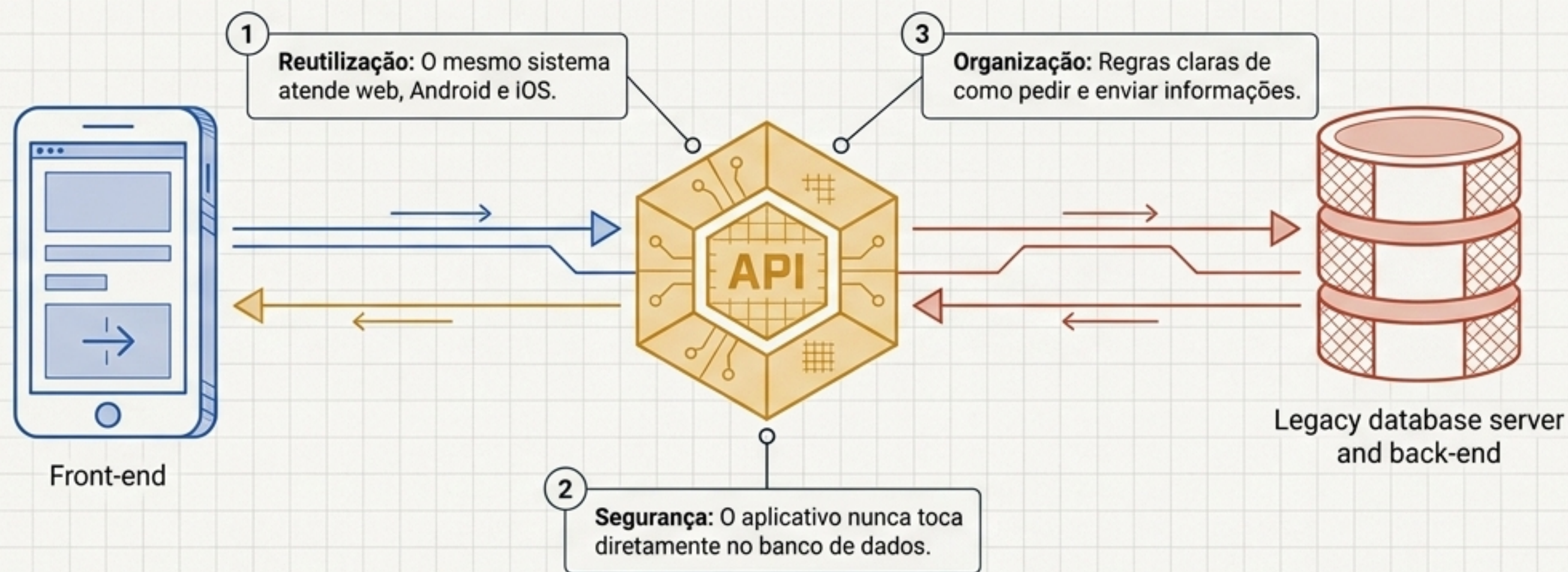
O que acontece exatamente quando você faz login em um aplicativo de banco?



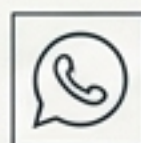
A API: O Centro de Distribuição Digital

Application Programming Interface

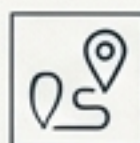
Uma API é uma interface estruturada que permite que dois sistemas completamente diferentes conversem entre si. Ela recebe uma solicitação do aplicativo, processa e devolve a resposta exata do banco de dados.



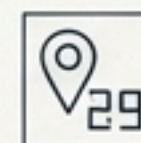
APIs que movem a internet:



API do WhatsApp



API do Google Maps



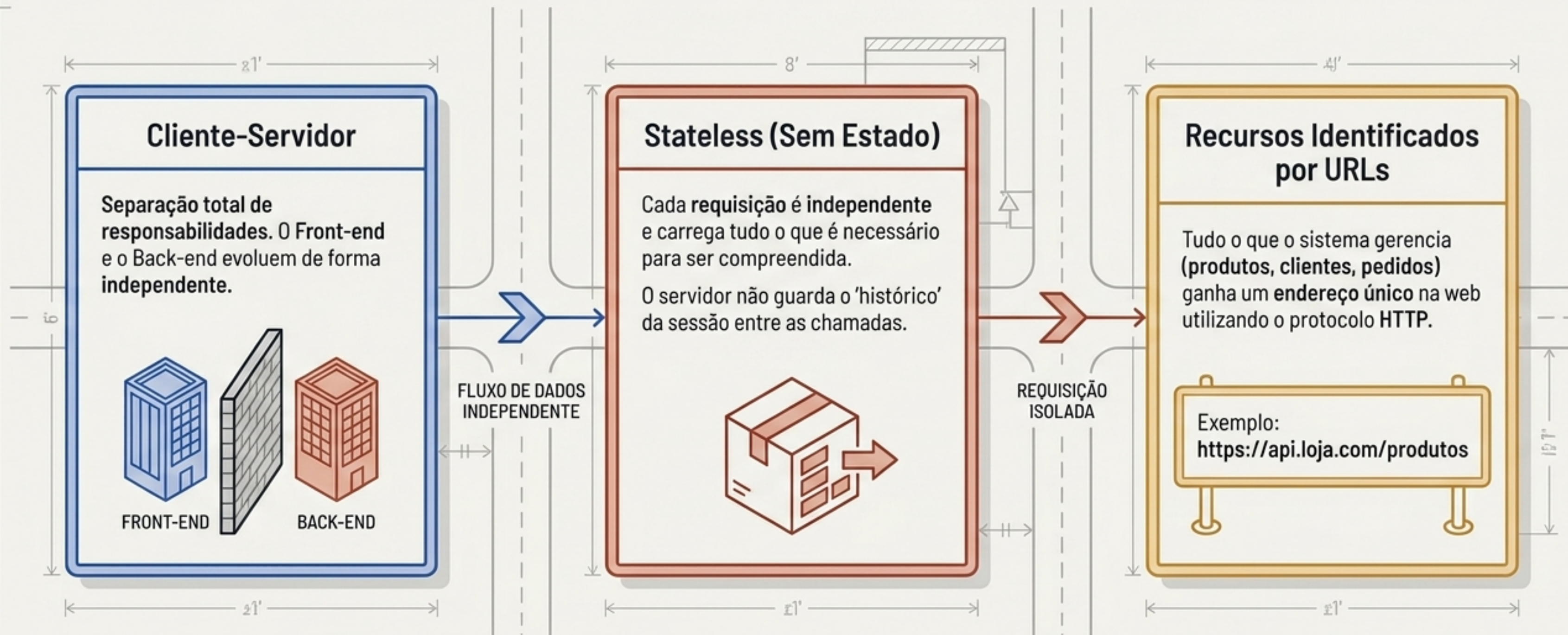
API de CEP



API do GitHub

O Padrão REST: As Leis de Trânsito

Representational State Transfer é o conjunto de **boas práticas** que padroniza a criação de APIs modernas.



A Matriz de Comando: Verbos e Endpoints

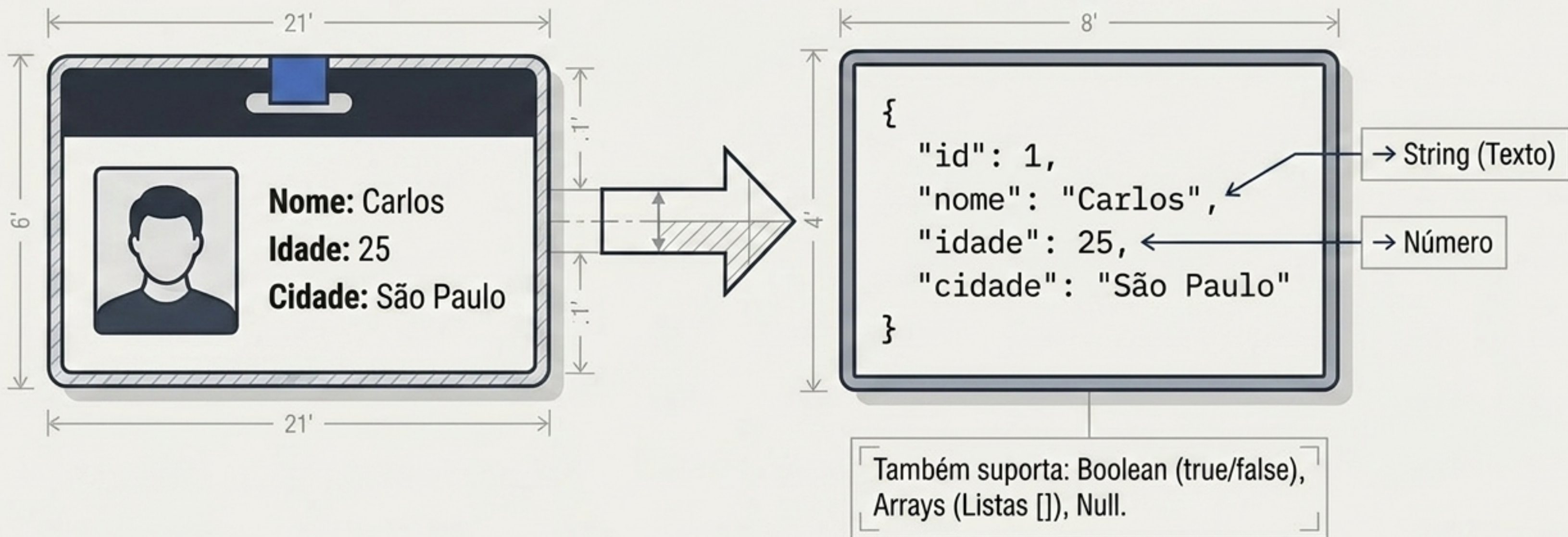
O Endpoint (/recurso) é o destino. O Verbo HTTP é a ação que queremos realizar ao chegar lá.

GET (Consultar) + /produtos		Lista todos os produtos disponíveis.
POST (Inserir) + /produtos		Cadastra um novo produto no sistema.
PUT / PATCH (Atualizar) + /produtos/10		Atualiza (total ou parcialmente) o produto de ID número 10.
DELETE (Excluir) + /produtos/10		Remove o produto de ID número 10 do banco de dados.

JSON: O Contêiner Padronizado

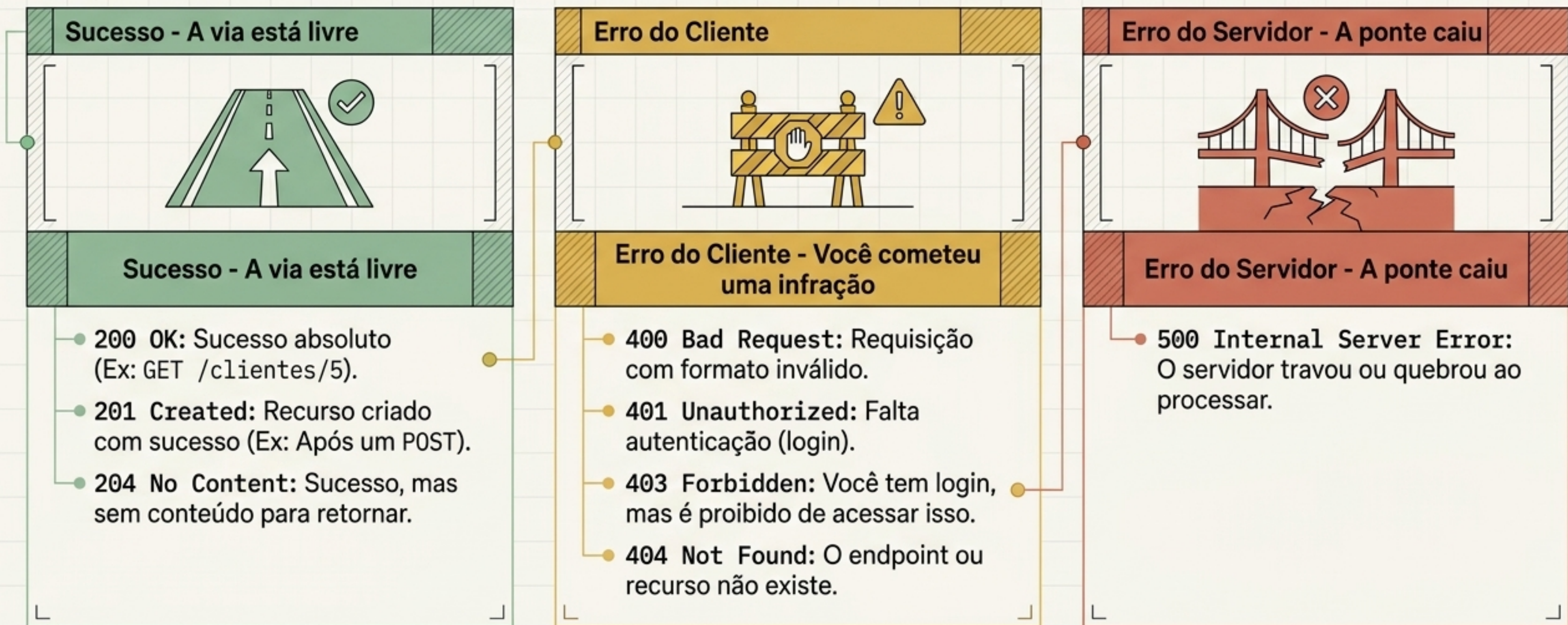
JavaScript Object Notation é o formato universal de texto para troca de dados entre sistemas.

Code-to-Visual Payload



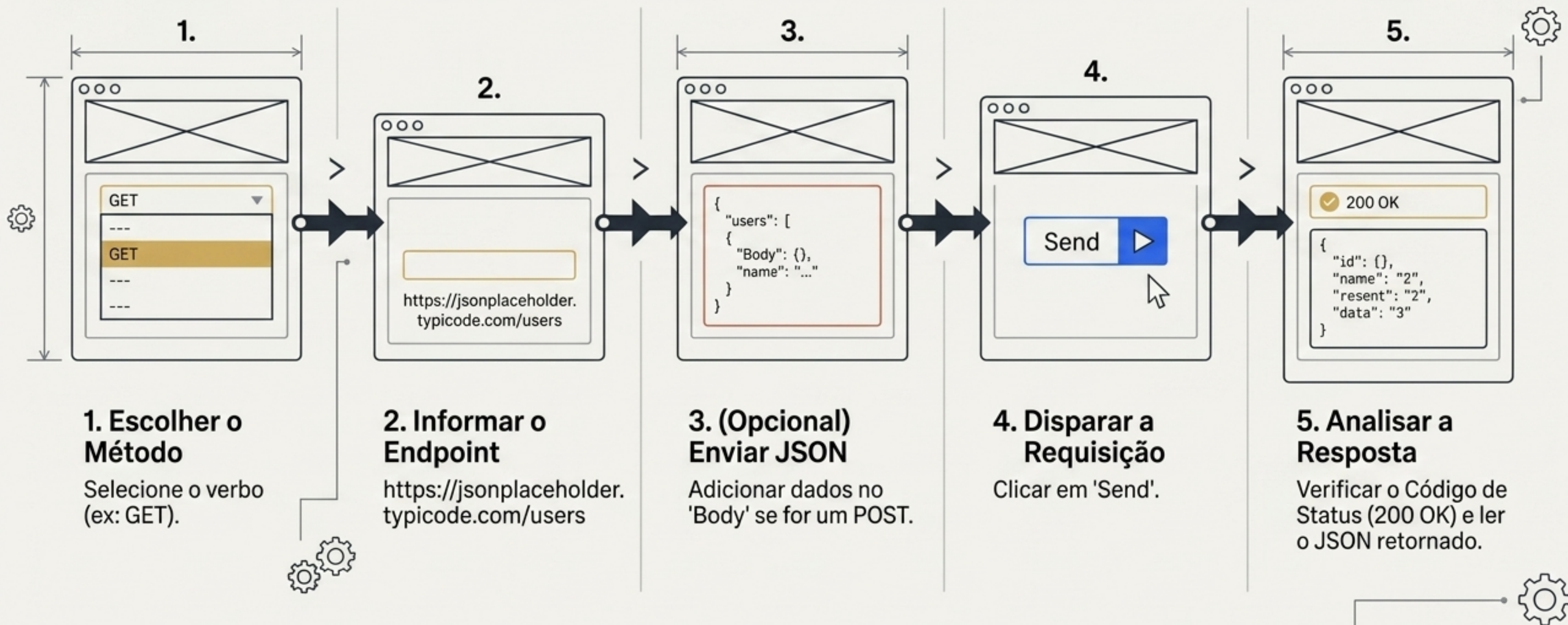
Códigos de Status HTTP: O Painel de Tráfego

Como o servidor sinaliza o resultado da sua requisição.



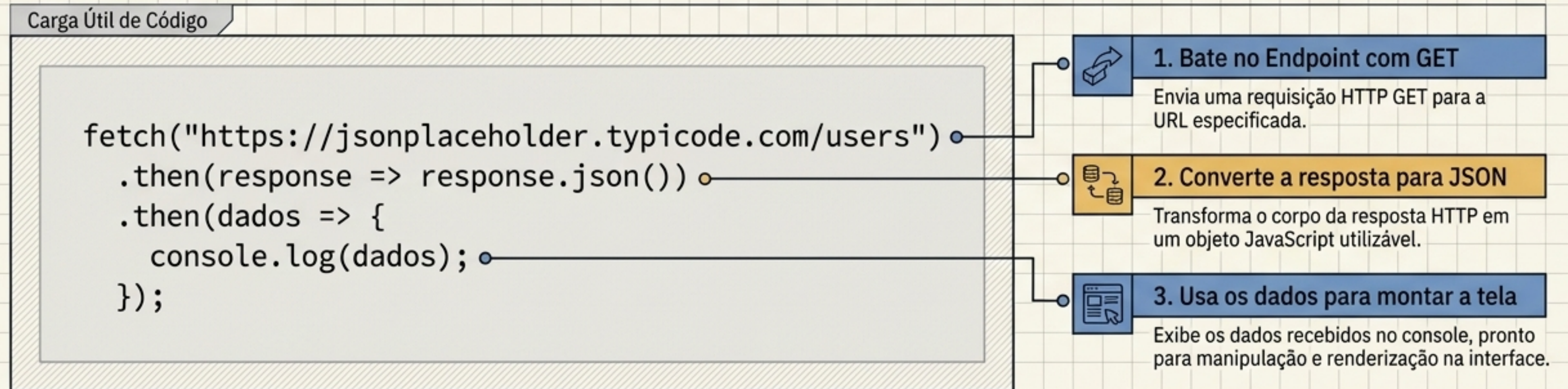
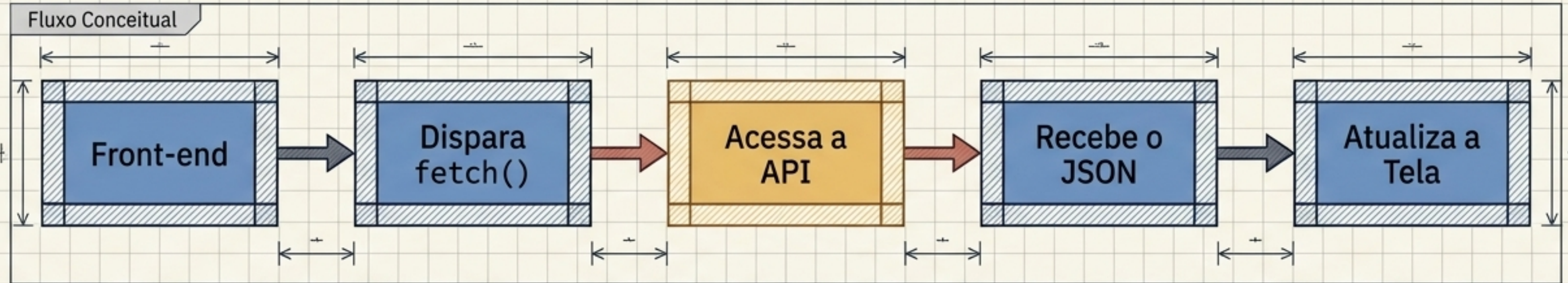
Postman: O Test-Drive da Rota

Antes de integrar com o Front-end, testamos a API isoladamente.



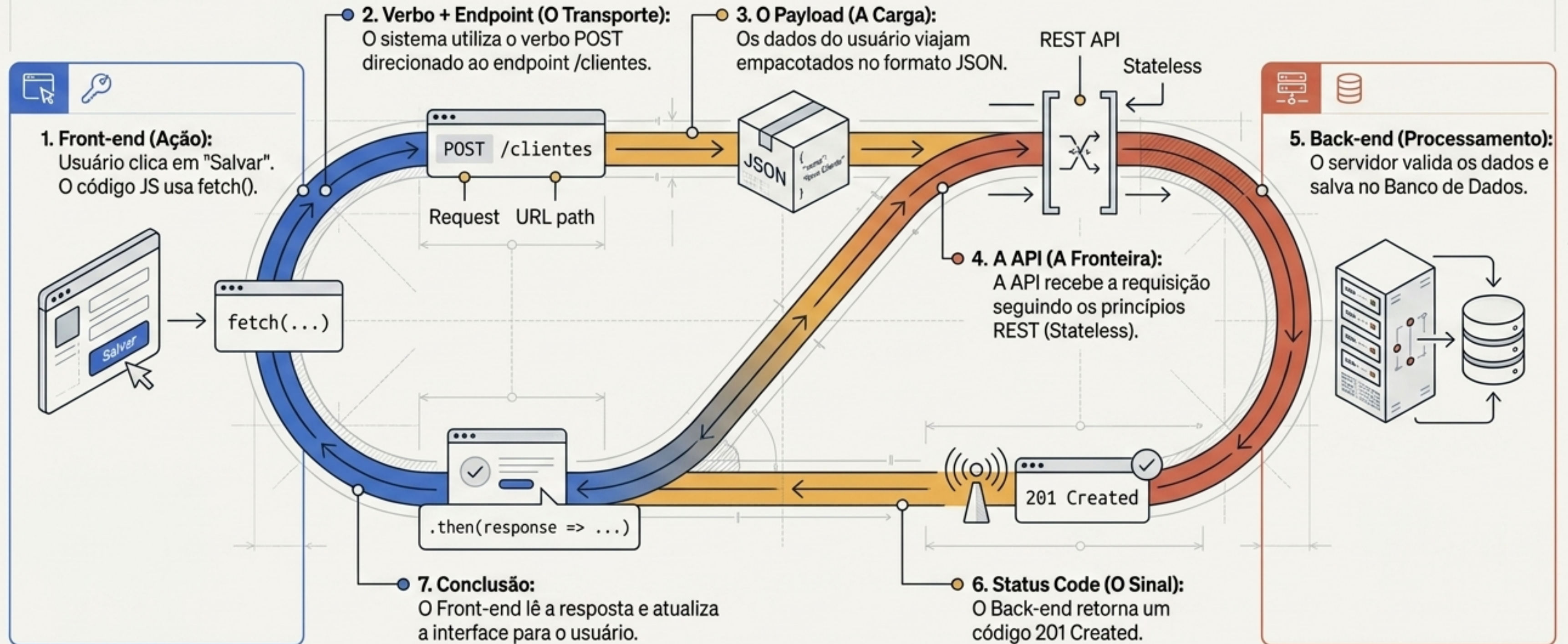
JavaScript fetch(): Pavimentando a Estrada

Como o Front-end consome os dados da API de forma programática.



A Anatomia de uma Requisição Completa

Este é o grande esquema de síntese, um plano mestre de 7 passos.



Projeto Final: Assumindo o Controle

Chegou a hora de construir o seu próprio sistema.

- [] **Planejar a Rota:** Escolha uma API REST pública.
- [] **Testar no Laboratório:** Utilize o Postman para testar os endpoints antes de codificar.
- [] **Consultar Dados:** Utilize o método GET via JavaScript (`fetch()`) para buscar informações.
- [] **Inserir Dados:** Crie uma função que envie novos dados para a API utilizando POST e um payload em JSON.
- [] **Gerenciar o Tráfego:** Capture e interprete corretamente os Códigos de Status (Exiba sucesso para 200/201 e mensagens de erro para 400/500).
- [] **Renderizar:** Exiba os resultados em uma interface HTML organizada.

Ao concluir este desafio, você dominará a infraestrutura invisível que conecta a web moderna.