

## **Resumo: Diferenças entre RPC e REST**

RPC (Chamada de Procedimento Remoto) e REST (Representational State Transfer) são dois estilos arquitetônicos usados para construir APIs. Ambas permitem que diferentes componentes de software se comuniquem, promovendo integração entre sistemas. No entanto, apresentam abordagens distintas para lidar com essa comunicação.

As APIs RPC simulam chamadas (podendo serem conhecidas como procedimentos) de função locais, podendo passar valores de dados aos servidores remotos. Essa abordagem é útil para ações específicas, como adicionar funcionalidades externas (ex: chat), chamando métodos diretamente.

A RPC se baseia em funções ou ações, e seus princípios não são padronizados (Podem utilizar o método POST para alterar ou recuperar valores, por exemplo) e podem guardar estado, enquanto a REST se concentra em recursos ou objetos, não permite que o servidor guarde estados do cliente a cada requisição, e exerce operações como criar, ler, atualizar ou deletar dados por meio de URLs e verbos HTTP, tendo o padrão:

- GET - Recupera valores;
- POST - Adiciona valores;
- PUT - Atualiza valores;
- DELETE - Exclui valores;

Apesar das diferenças, REST e RPC compartilham semelhanças: ambas abstraem os detalhes técnicos de rede, usam protocolos como HTTP e suportam formatos como JSON e XML. Essa abstração facilita o trabalho dos desenvolvedores, permitindo foco nas funcionalidades. Além disso, ambas são compatíveis com múltiplas linguagens de programação, o que garante flexibilidade na sua implementação.

REST é geralmente preferido por sua escalabilidade, padronização e facilidade de manutenção, sendo mais comum na web. Já RPC pode ser mais direto e eficiente em chamadas simples ou dentro de sistemas internos com baixa latência.

A escolha entre REST e RPC depende do contexto: REST se destaca em ambientes distribuídos, com foco em interoperabilidade e legibilidade. RPC pode ser ideal para ações específicas com menos sobrecarga.

Por fim, a popularidade crescente do REST é atribuída à sua simplicidade e ao uso natural com a estrutura da web. No entanto, ambos os estilos continuam relevantes, sendo amplamente suportados por serviços modernos como os da AWS.