As 6 regras de uma API REST são:

Arquitetura cliente-servidor: A arquitetura cliente-servidor é a base da web e da arquitetura REST. Isso significa que o cliente e o servidor são sistemas independentes que se comunicam entre si através de uma interface definida por uma API.

Stateless: Cada solicitação do cliente para o servidor deve conter todas as informações necessárias para que o servidor possa entender e responder à solicitação. Isso significa que o servidor não deve manter nenhum estado entre as solicitações do cliente.

Cache: As respostas do servidor devem ser marcadas como cacheáveis ou não cacheáveis. As respostas cacheáveis devem ser armazenadas em cache pelo cliente para que possam ser reutilizadas posteriormente.

Interface uniforme: A interface uniforme é o coração da arquitetura REST. Isso significa que a API deve ser projetada em torno de recursos, que são identificadores exclusivos para as entidades que estão sendo manipuladas pela API. Cada recurso deve ter uma representação, que pode ser um documento HTML, JSON, XML ou outro formato.

Camadas: A arquitetura REST deve ser projetada em camadas. Isso significa que o cliente não precisa saber sobre a estrutura interna do servidor e o servidor não precisa saber como o cliente está implementado. Cada camada é responsável por uma tarefa específica e não pode acessar diretamente as camadas acima ou abaixo dela.

Sistema de código sob demanda (opcional): A arquitetura REST permite que o servidor envie código executável para o cliente, como JavaScript ou applets Java. Isso permite que o servidor estenda a funcionalidade do cliente e melhore a experiência do usuário, mas é uma opção que nem sempre é usada.

Em resumo, as 6 regras de uma API REST são: arquitetura cliente-servidor, stateless, cache, interface uniforme, camadas e sistema de código sob demanda (opcional). Essas regras ajudam a garantir que a API seja bem projetada, escalável, segura e fácil de usar.