As seis regras de uma API REST, também conhecidas como os princípios REST, são:

1. Arquitetura cliente-servidor: a API REST deve ser baseada em uma arquitetura cliente-servidor, na qual o cliente faz requisições ao servidor e o servidor processa essas requisições e retorna uma resposta ao cliente.
2. Comunicação sem estado: cada requisição feita pelo cliente ao servidor deve conter todas as informações necessárias para que o servidor possa processar a requisição. O servidor não deve manter nenhum tipo de estado entre as requisições.
3. Cacheável: as respostas enviadas pelo servidor devem ser explicitamente marcadas como cacheáveis ou não-cacheáveis. Isso permite que os clientes possam armazenar em cache as respostas recebidas, melhorando a eficiência da comunicação.
4. Interface uniforme: a API REST deve ser definida através de uma interface uniforme, que define um conjunto de operações padrão (como GET, POST, PUT e DELETE) e recursos, permitindo que o cliente possa interagir com a API de forma padronizada.
5. Sistema em camadas: a arquitetura da API REST deve ser composta por camadas, onde cada camada é responsável por realizar uma determinada tarefa (como autenticação, roteamento, armazenamento em cache, etc.). Isso permite que a API seja escalável e flexível.
6. Código sob demanda (opcional): a API REST pode permitir que o servidor envie código executável para o cliente, permitindo que o cliente possa estender a funcionalidade da API. Esta regra é opcional e não é utilizada com frequência.

Essas seis regras foram definidas por Roy Fielding em sua tese de doutorado e são amplamente utilizadas como guias para o desenvolvimento de APIs RESTful. Ao seguir essas regras, é possível criar APIs que sejam flexíveis, escaláveis e de fácil integração com outras aplicações.