



# Chamada de Procedimento Remoto (RPC)

Por Victor Guedes



# Definição e forma de funcionamento

- RPC é um protocolo que permite a comunicação entre dois ou mais sistemas, ou partes de software, que estejam localizados em diferentes dispositivos. Assim permitindo que um sistema “chame” uma função ou procedimento localizado em outro sistema.
- O processo envolve 2 componentes: Cliente RPC e servidor RPC. O servidor RPC é onde está localizado o serviço que o cliente RPC está solicitando, o servidor RPC atende a solicitação, processa o pedido e devolve o resultado.



# Importância do RPC em Sistemas Distribuídos

- A importância do protocolo RPC está na sua capacidade de aumentar escalabilidade e desacoplamento entre sistemas, implementando chamadas remotas de forma “simples”, como se fosse chamadas de métodos locais.
- Desta forma, o sistema consegue ser desenvolvido de forma mais eficiente e menos complexa, visto que, com o RPC, exclui-se a necessidade de gerenciar sockets de forma manual, ou mesmo se preocupar com serialização.



# Vantagens e limitações do RPC

- Simplicidade - Tanto para o servidor, quanto para o cliente, implementar o RPC acaba sendo simples, sendo muito parecido com chamadas de métodos locais
- Transparência - Essa simplicidade faz com que os desenvolvedores não precisam gerenciar manualmente sockets ou mecanismos de serialização.
- Escalabilidade - Usar o RPC aumenta a escalabilidade de um sistema, já que permite que ele se desenvolva de forma independente dos outros.



# Limitações do RPC

- Latência na rede - Como o RPC é um protocolo de comunicação na rede, ele pode estar sujeito à travamentos ou demora na resposta.
- Segurança - As chamadas RPC podem ser vulneráveis a ataques na rede, como interceptação de mensagens. O que exige a necessidade de medidas de segurança, como autenticação e criptografia.
- Tratamento de erros - Por conta dos serviços ou métodos chamados não serem locais, pode ser complexo tratar os erros em um sistema RPC. Por exemplo, a conexão do servidor RPC cair no meio de uma execução do cliente, ou mesmo um problema na rede causar uma execução repetida de algum serviço.



# Bibliotecas existentes que implementam o RPC

- xmlrpc - Uma biblioteca padrão do Python que implementa RPC, tem como diferença que usa o formato XML com o protocolo XML-RPC para codificar e decodificar suas chamadas e respostas.
- jsonrpclib - É uma biblioteca externa que implementa o RPC, usando o formato JSON com o protocolo JSON-RPC para codificar e decodificar suas chamadas e respostas.
- RPyC - É uma biblioteca que também implementa o RPC em python, permitindo a execução um método ou serviço remoto de outro programa também em python.



# Boas práticas

- Tratamento de erros adequado - Como dito anteriormente, o uso do RPC pode gerar erros imprevistos, sendo necessário a implementação de tratamentos adequados como timeouts, retentativas, entre outros.
- Garantir segurança durante a comunicação - As chamadas RPC podem ser vulneráveis à ataques, como interceptação da resposta do servidor, sendo necessário a implementação de uma criptografia ou autenticação adequada.
- Otimizar o desempenho - Adotando medidas como reduzir o número de chamadas RPC (Evitar chamadas desnecessárias), ou caching (Armazenar uma resposta que não muda com frequência).