



# MIT Arquitetura de Sistemas, turma 2020.1

Professor: Gustavo de Miranda

Aluno: Rodrigo Germano



INSTITUTO  
INFNET

# RPC

A idéia de RPC data de 1976, quando foi descrito no RFC 707. Um dos primeiros usos comerciais da tecnologia foi feita pela Xerox no "Courier", de 1981. A primeira implementação popular para Unix foi o Sun RPC (atualmente chamado ONC RPC), usado como base do Network File System e que ainda é usada em diversas plataformas.

Outra implementação pioneira em Unix foi o Network Computing System (NCS) da Apollo Computer, que posteriormente foi usada como fundação do DCE/RPC no Distributed Computing Environment (DCE). Uma década depois a Microsoft adotou o DCE/RPC como base para a sua própria implementação de RPC, MSRPC, a DCOM foi implementada com base nesse sistema. Ainda no mesmo período da década de 1990, o ILU da Xerox PARC e o CORBA ofereciam outro paradigma de RPC baseado em objetos distribuídos, com mecanismos de herança.

De forma análoga, atualmente utiliza-se XML e JSON como linguagem de descrição de interface e HTTP como protocolo de rede para formar serviços web, cujas implementações incluem SOAP, REST e XML-RPC.

Disponível em: [https://pt.wikipedia.org/wiki/Chamada\\_de\\_procedimento\\_remoto](https://pt.wikipedia.org/wiki/Chamada_de_procedimento_remoto) Acesso em: 11/05/2020.



# gRPC

O gRPC é uma biblioteca RPC universal desenvolvida pelo Google, com código aberto e alto desempenho. No gRPC, um aplicativo cliente pode chamar métodos diretamente em um aplicativo servidor de uma outra máquina, como se ela fosse um objeto local, facilitando a criação de aplicativos e serviços distribuídos.

Um dos principais benefícios de usar gRPC refere-se à documentação. É possível usar sua configuração do serviço e arquivos de definição da interface da API para gerar documentação de referência para sua API. Consulte Visão geral do portal do desenvolvedor para saber mais informações.

Disponível em: <https://cloud.google.com/endpoints/docs/grpc/about-grpc?hl=pt-br> Acesso em: 11/05/2020.



# Vantagens

- Open source.
- Serialização Eficiente (Binária).
- Multiplataforma.
- Streaming cliente, servidor e bidirecional.
- HTTP<sub>2</sub>.

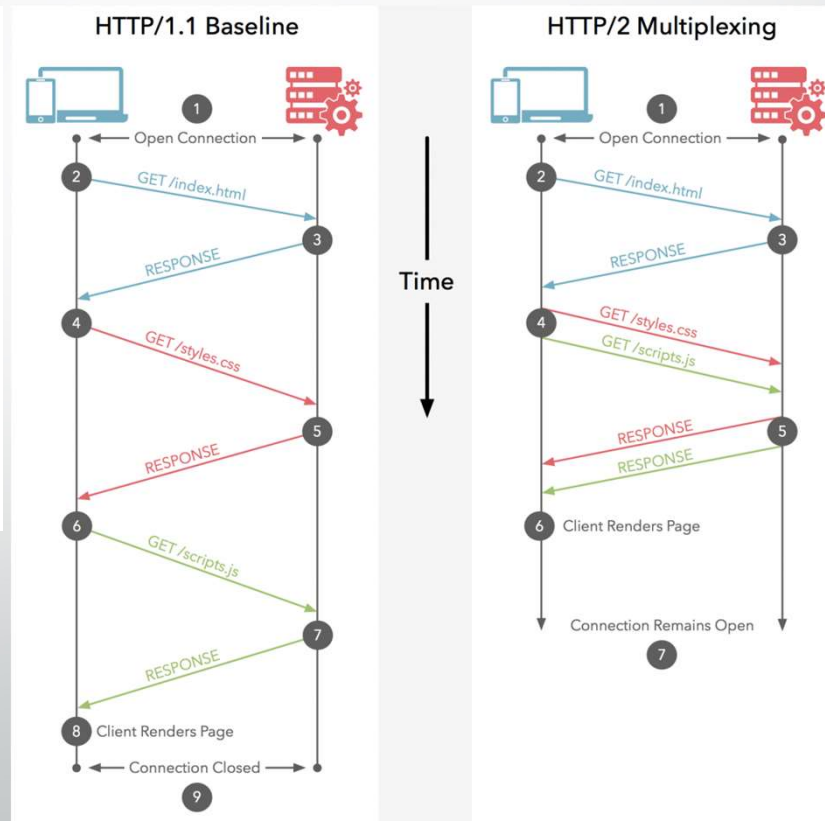
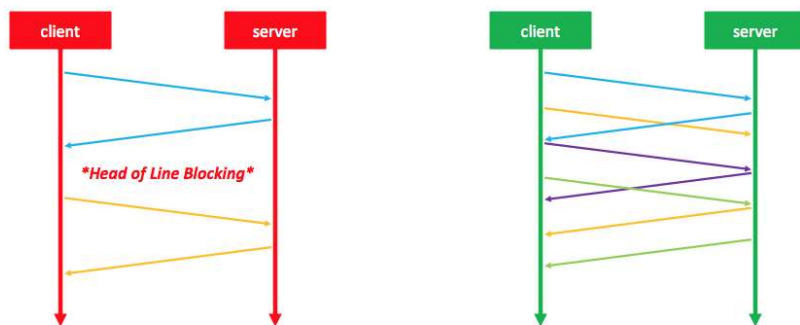
Disponível em: <https://gago.io/blog/grpc-no-asp-net-core-guia-introdutorio/> Acesso em: 11/05/2020.

Disponível em: <https://docs.microsoft.com/pt-br/aspnet/core/grpc/comparison?view=aspnetcore-3.1> Acesso em: 11/05/2020.



# HTTP<sub>2</sub>

## HTTP/1.1 Pipelining vs HTTP/2 Multiplexing



Disponível em: <https://gago.io/blog/grpc-no-asp-net-core-guia-introdutorio/> Acesso em: 11/05/2020.

# Desvantagens

- O HTTP2 não é suportado em todos os webserver, proxy reversos, ou load balancers.
- O browser suporta HTTP2, mas existem pendências (motores de javascript e libraries).
- Não é legível por humanos. (Não tem swagger 😞)

Disponível em: <https://gago.io/blog/grpc-no-asp-net-core-guia-introdutorio/> Acesso em: 11/05/2020.

Disponível em: <https://docs.microsoft.com/pt-br/aspnet/core/grpc/comparison?view=aspnetcore-3.1> Acesso em: 11/05/2020.



# gRPC x APIs HTTP com JSON

Recurso	gRPC	APIs HTTP com JSON
Contrato	Necessário (.proto)	Opcional (OpenAPI)
Protocolo	HTTP/2	HTTP
Carga útil	Protobuf (pequeno, binário)	JSON (grande, legível por humanos)
Prescriptiveness	Especificação estrita	Cancelar. Qualquer HTTP é válido.
Streaming	Cliente, servidor, bidirecional	Cliente, servidor
Suporte ao navegador	Não (requer grpc-Web)	Sim
Segurança	Transporte (TLS)	Transporte (TLS)
Geração de código de cliente	Sim	OpenAPI + ferramentas de terceiros

Disponível em: <https://docs.microsoft.com/pt-br/aspnet/core/grpc/comparison?view=aspnetcore-3.1> Acesso em: 11/05/2020.



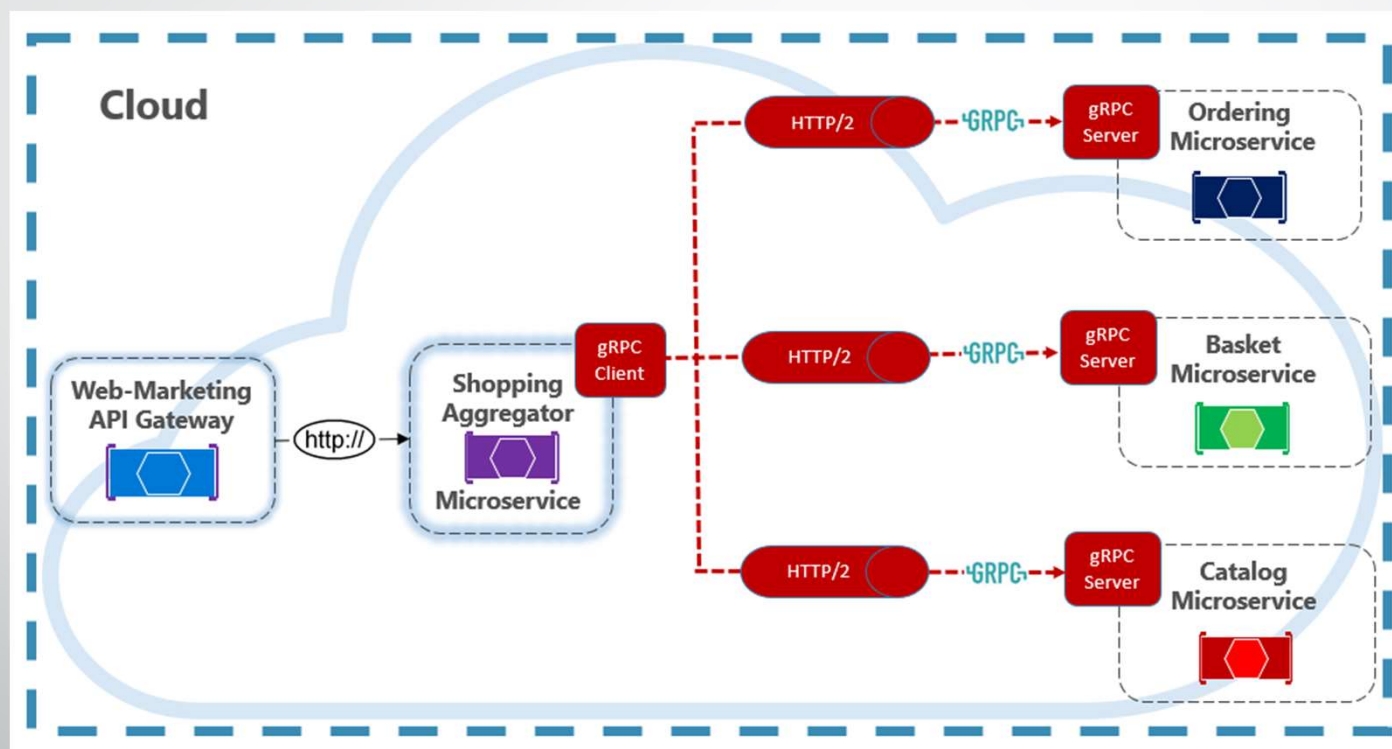
Demo



INSTITUTO  
INFNET



# Minha opinião



Disponível em: <https://docs.microsoft.com/pt-br/dotnet/architecture/cloud-native/grpc> Acesso em: 11/05/2020.

# Referências

- <https://grpc.io/docs/>
- <https://developers.google.com/protocol-buffers>
- <https://docs.microsoft.com/pt-br/aspnet/core/grpc/?view=aspnetcore-3.1>
- <https://gago.io/blog/grpc-no-asp-net-core-guia-introdutorio/>
- [https://pt.wikipedia.org/wiki/Chamada\\_de\\_procedimento\\_remoto](https://pt.wikipedia.org/wiki/Chamada_de_procedimento_remoto)
- <https://docs.microsoft.com/pt-br/dotnet/architecture/cloud-native/grpc>



# Obrigado!



Link do material

<https://github.com/rodrigogermano/PGPARQo1C1.2N.P1.DOTNET>



INSTITUTO  
INFNET