

---

---

# GRPC & PROTOBUF

Thiago Luiz de Souza Gomes 180028324

---

# gRPC

O gRPC é uma estrutura de código aberto e alto desempenho criada pelo Google. O gRPC segue amplamente a semântica HTTP sobre HTTP/2 e, assim, permite que você use o streaming full-duplex, possibilitando a comunicação entre diferentes sistemas via conexão de rede. Com isso, um ponto de grande impacto é o ganho de performance ao trafegar um payload extensivo, visto que, no gRPC a comunicação é feita em binário, o que traz uma melhora enorme na performance, que pode ser até 7 vezes mais rápida na resposta de dados e 10 vezes no envio. Isso se deve principalmente ao empacotamento compacto dos proto buffers e ao uso de HTTP/2 pelo gRPC.

# ProtoBuf

Protobuf (sigla de Protocol buffers) é um mecanismo criado e usado pelo Google para serializar dados estruturados. É independente de linguagem ou plataforma.

# ProtoBuf

-Primeiro é definido como deseja que os dados sejam estruturados — em um arquivo de extensão .proto.-

-Em seguida, esta definição é compilada e o resultado é um código-fonte automaticamente gerado na linguagem desejada — no momento que escrevo este post, as linguagens compatíveis são C++, C#, Go, Java e Python.

-Finalmente, código-fonte gerado é utilizado para gravar e ler os dados estruturados.

-Sempre que houver mudança na estrutura dos dados, o ciclo se repetirá.

# ProtoBuf

```
syntax = "proto3";

service ElementService {
  rpc List (Empty) returns (ElementList) {}
}

message Empty {}

message Element {
  int32 id = 1;
  double number = 2;
}

message ElementList {
  repeated Element elements = 1;
}
```

# HTTP/2

HTTP2 é uma versão nova do protocolo HTTP/1.1 que é usado pela internet amplamente para servidores e sites. Ele permite a conversa padronizada entre cliente, em geral os navegadores, e servidor, que é onde fica armazenado o conteúdo enviado quando um site é acessado.

# HTTP1.1

- cliente requisita site.com.br/index.html
- servidor envia o arquivo index.html para o cliente
- cliente processa index.html
- cliente então requisita site.com.br/estilo.css
- cliente então requisita site.com.br/menu.js
- cliente então requisita site.com.br/header.png
- servidor envia o arquivo estilo.css para o cliente
- servidor envia o arquivo menu.js para o cliente
- servidor envia o arquivo header.png para o cliente
- cliente processa a página completa  
(index.htm+menu.js+estilo.css+header.png)

# HTTP/2

- cliente requisita site.com.br/index.html
- servidor envia o arquivo index.html para o cliente
- servidor sabe que já precisa enviar estilo.css
- servidor sabe que já precisa enviar menu.js
- cliente processa index.html
- cliente então requisita site.com.br/header.png
- servidor envia o arquivo header.png para o cliente
- cliente processa a página completa  
(index.htm+menu.js+estilo.css+header.png)





# Bibliografia

<https://king.host/blog/2017/04/o-que-e-o-http2-e-quais-os-seus-beneficios/>

[https://www.hostinger.com.br/tutoriais/installar-node-js-ubuntu?ppc\\_campaign=google\\_performance\\_max&gclid=CjwKCAjw7SWBhAnEiwAx8ZLajiwxWKmqoMDCD4gEZHxrgpGzCnxY5PXT2EauiUWFx3UGcyHFnmgxCGmlQAvD\\_BwE](https://www.hostinger.com.br/tutoriais/installar-node-js-ubuntu?ppc_campaign=google_performance_max&gclid=CjwKCAjw7SWBhAnEiwAx8ZLajiwxWKmqoMDCD4gEZHxrgpGzCnxY5PXT2EauiUWFx3UGcyHFnmgxCGmlQAvD_BwE)

<https://vertigo.com.br/grpc-ou-rest-qual-utilizar/>

<https://medium.com/trainingcenter/protobuf-uma-alternativa-ao-json-e-xml-a35c66edab4d>