



Tutorial PyNE 2019

Construindo microserviços de alta performance com gRPC

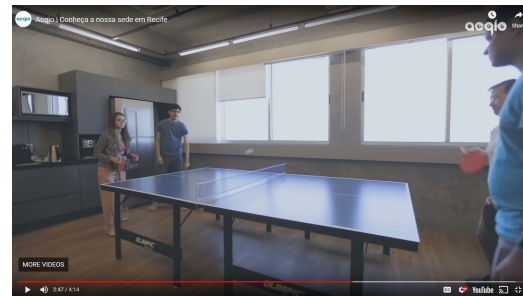
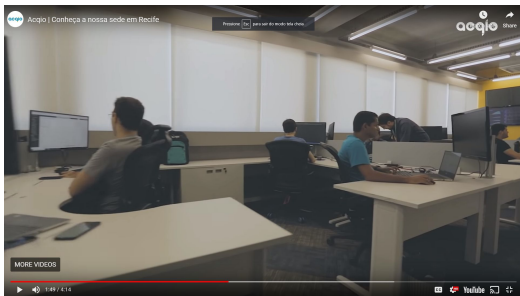
Paulo Barros

Sobre mim

- Mestre e bacharel em CC pela UFPE.
- SPARTA e Checker Framework - University of Washington.
- 3 anos como dev na Acqio.
- Tech-lead.
- “Nem sempre o que funciona está correto”.



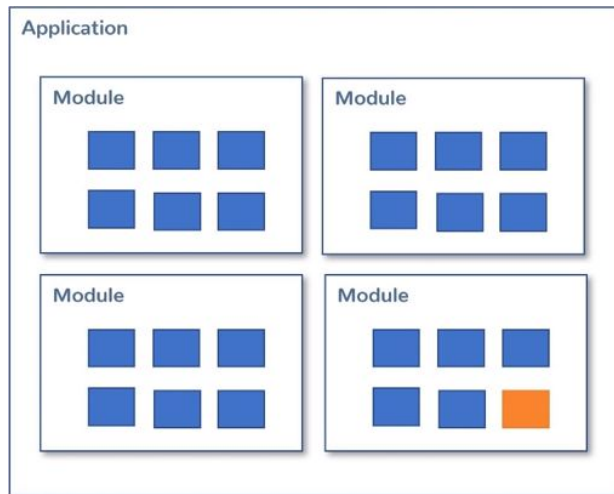
A Acqio está contratando!



<https://acqio.com.br/jobs/>

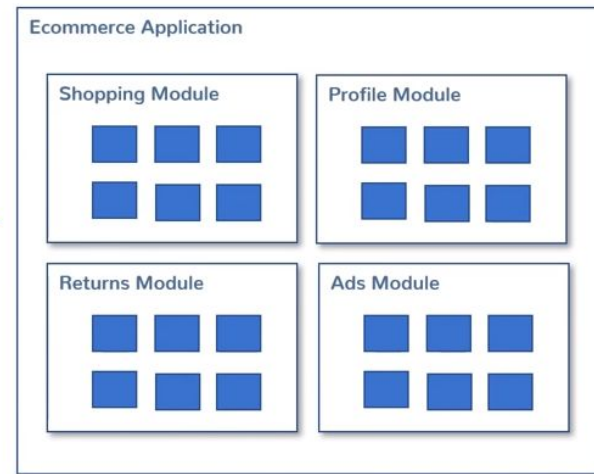


Aplicações monolíticas

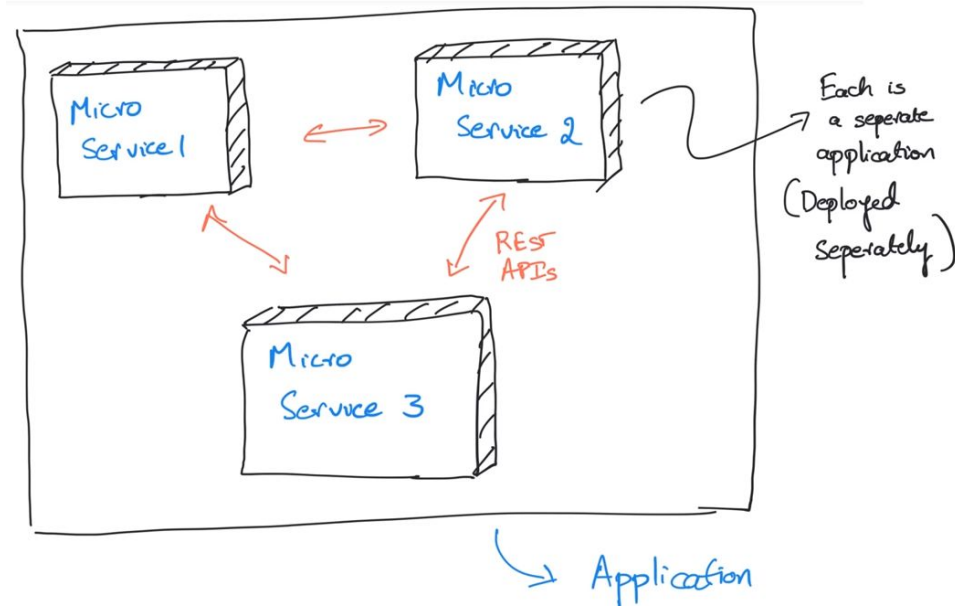


Aplicações monolíticas

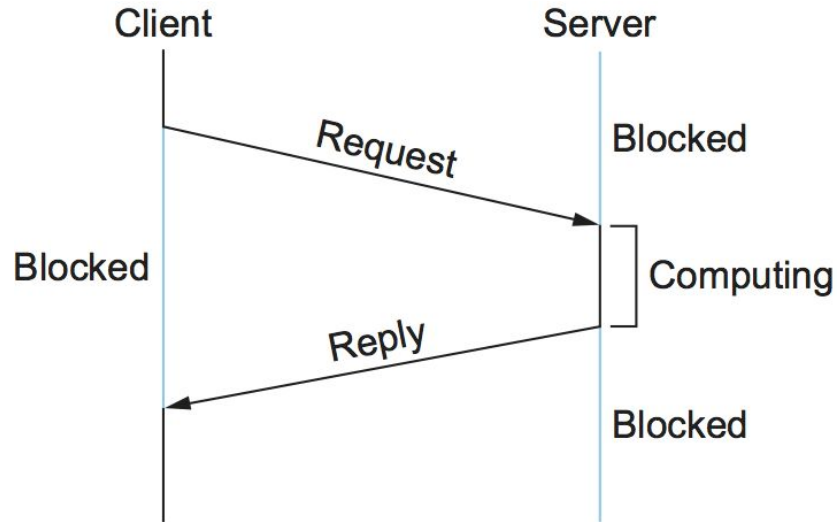
- Difícil de deployar
- Difícil de escalar



Microserviços



O que é RPC?



E gRPC?

- Framework RPC universal, gratuito e open-source.
- Criado pelo Google.
- Faz parte da CNCF, que inclui Docker e Kubernetes.
- Gera código RPC para cliente e servidor.
- Alta performance (Protocol Buffer + HTTP2).
- Fácil de usar.
- Suporte plugável para:
 - Streaming
 - Autenticação
 - Monitoramento
 - Balanceamento de carga.
 - ...
- ...

gRPC

C++	Java	Python
Go	Ruby	C#
Node.js	Android Java	Objective-C
	PHP	



Protocol Buffers

- IDL para representar dados estruturados.
- Definição de mensagens e serviços.
- Gerador de código RPC para cliente e servidor.
- Formato binário.
- De 3 a 4 vezes mais compacta que JSON.
- Deserialização consome menos CPU que parsear JSON, economizando bateria de dispositivos mobile.



Protocol Buffers

```
protos ▸ pyne ▸ greeter.proto
1  syntax = "proto3";
2
3  service Greeter {
4    rpc SayHello (HelloRequest) returns (HelloReply) {}
5  }
6
7  message HelloRequest {
8    string name = 1;
9    int32 age = 2;
10 }
11
12 message HelloReply {
13   string greetings = 1;
14 }
15
```



Protocol Buffers

<http://bit.ly/protovsjson>



HTTP 1.1 X HTTP 2

<http://bit.ly/http2vshttp1>



HTTP 1.1 X HTTP 2

HTTP1.1

HTTP2

Lançado em 1997.	Lançado em 2015.
Abre nova conexão a cada requisição.	Multiplexação.
Request e Response unária.	Suporte a server push e streams.
Não comprime o Header. Header em PLAINTEXT.	Comprime o Header. Header binário.
O padrão é HTTP	O padrão é HTTPS



E APIs REST?

- JSON
- HTTP 1.1
- CRUD-oriented: GET, POST, ...
- Request e Response
- Client -> Server

```
POST /member/109087/cart HTTP/1.1
Host: api.example.com
Authorization: Basic username:password
Content-type: application/json
Accept: application/hal+json

{
  "inventory_id": 12345,
  "quantity": 1
}

HTTP/1.1 201 Created
Date: Mon, 20 Jun 2011 21:15:00 GMT
Content-Type: application/hal+json
Location: /member/109087/cart/14418796
```



gRPC x API REST

gRPC

API REST

Protocol buffers	JSON
HTTP2	HTTP1.1
Bidirecional e async	Client -> Server apenas
Suporte a streaming	Request e Response apenas
Baseado em RPC	Baseado em verbos HTTP (CRUD)

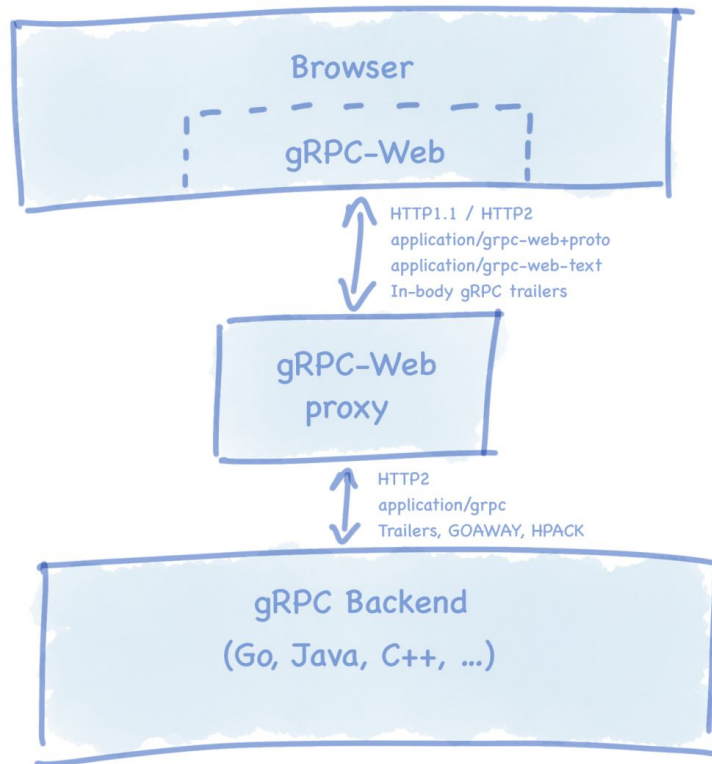


gRPC x API REST

Experimentos comprovando que gRPC é até 25 vezes mais eficiente que REST:

<http://bit.ly/restvsgrpc>

gRPC-WEB





Que tal ver como funciona na prática?

Por favor acessem e clonem o repositório em:

<http://bit.ly/pyne-grpc>