Integrantes

Sergio Corral Cristo - UO265363

Pedro José Fernández Franco – UO224775

Samuel Moreno Vincent -

UO266321

¿Qué es gRPC?

Principales características:

- Es un framework de RPCs
- Independiente del entorno de desarrollo.
- Es un proyecto open source
- Permite conectar microservicios de balanceo de carga, health checking, tracing y autentificación de forma facil y eficiente
- Usa protocol buffers
- Transmisión bidireccional mediante httpd/2

¿Cómo surge gRPC?

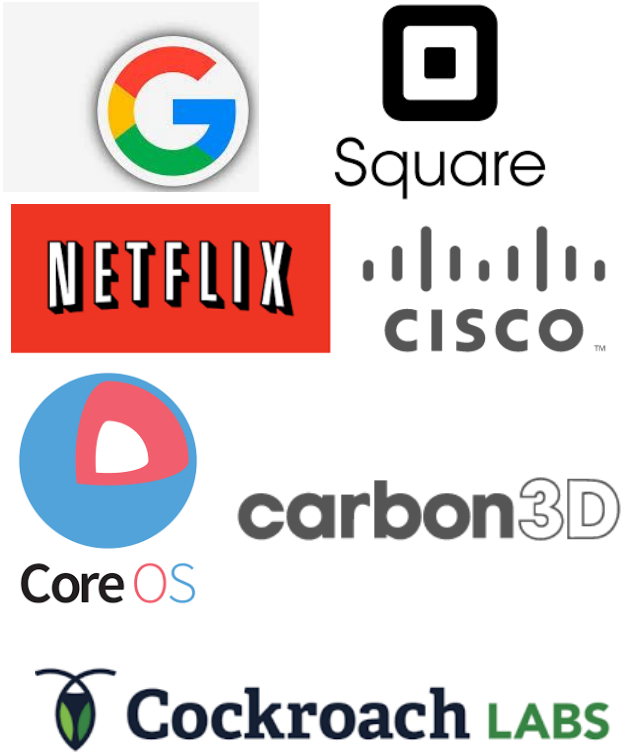
gRPC llega despues de la aparición de HTTP/2 y QUIC

gRPC surge como una necesidad para google de reelaborar su infraestructura de microservicios

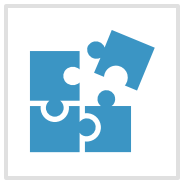
¿Dónde se usa mayoritariamente?

- Para conectar servicios poliglotos en una arquitectura de microservicios
- Para conectar backend con dispositivos móviles y navegadores
- Para generar librerías con gRPC

Stakeholders



Principales Atributos de Calidad



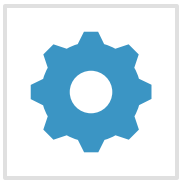
Compatibilidad

Soporta una gran cantidad de plataformas y lenguajes de programación.

Rendimiento

Diseñado para la comunicación de baja latencia y alto rendimiento.

- Http/2
- Uso de Protocol Buffers



Mantenibilidad

Proyecto de Código Abierto

Diversos Canales de Comunicación

Compartimentado

Usabilidad

Bastantes tutoriales y ejemplos

Restricciones

- Protocolo Binario
- Especificación Estricta
- Soporte Web Limitado

Gestión del proyecto en Github

- Open Source
- Sistema de Colaboracion Abierta
- Cuenta con un Código de Conducta y Reglas de Contribución
- Apartado de First Issues para principiantes
- Comunicación mediante foro de mensajes y chat

Arquitectura

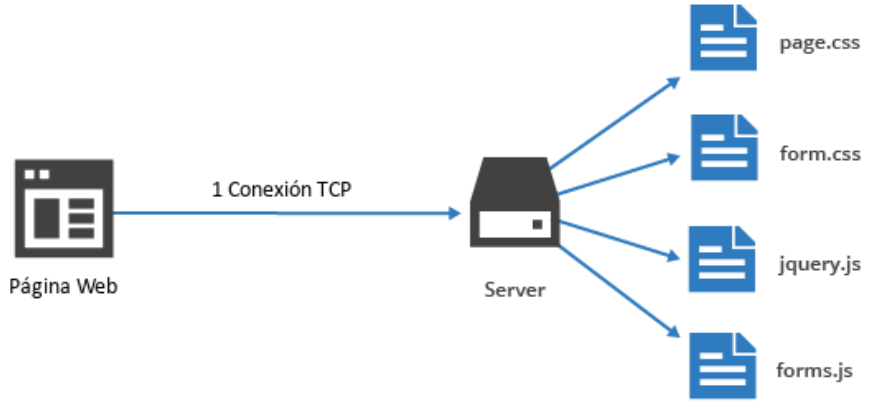
Python	Ruby	Java	C++	...
Python	Ruby	Java	C++	...
API genérico de bajo nivel en C				
Núcleo de gRPC en C				
Http 2.0				
SSL (Capa de seguridad)				

Lenguajes:

C++	NodeJS
C#	Objective-C
Dart	PHP
Go	Python
Java	Ruby
Kotlin	WebJS

	Capa de aplicación
	Capa de framework
	Capa de transporte

Protocolo HTTP2



- Una única conexión
- Multiplexación
- Protocolo binario
- Servicio 'server push'
- Compresión de cabeceras

Protocolo Buffers

```
//polyline.proto
syntax = "proto2";

message Point {
  required int32 x = 1;
  required int32 y = 2;
  optional string label = 3;
}

message Line {
  required Point start = 1;
  required Point end = 2;
  optional string label = 3;
}

message Polyline {
  repeated Point point = 1;
  optional string label = 2;
}
```

- Método de serialización de datos estructurados.
- Desarrollado por Google

- Mas simple que XML
- Mas fácil de programar que JSON