

experts
coding

¿Cómo, cuándo y por qué
usar gRPC?

Quien soy



Manu Vilachán

Más de 15 vueltas al sol diseñando cohetes

Arquitecto de Software, muy experto en tecnologías Microsoft y diseño de soluciones técnicas.

Desarrollo software complicado a veces, simple otras y si me queda un rato libre hago pan.

manuel.vilachan@expertsencoding.es
@manuvilachan





gRPC

- ¿Qué es y por qué se creó gRPC?
- ¿Cómo creamos y consumimos servicios gRPC en .net?
- ¿Cuándo usar gRPC?

¿Qué es y por qué se creó gRPC?



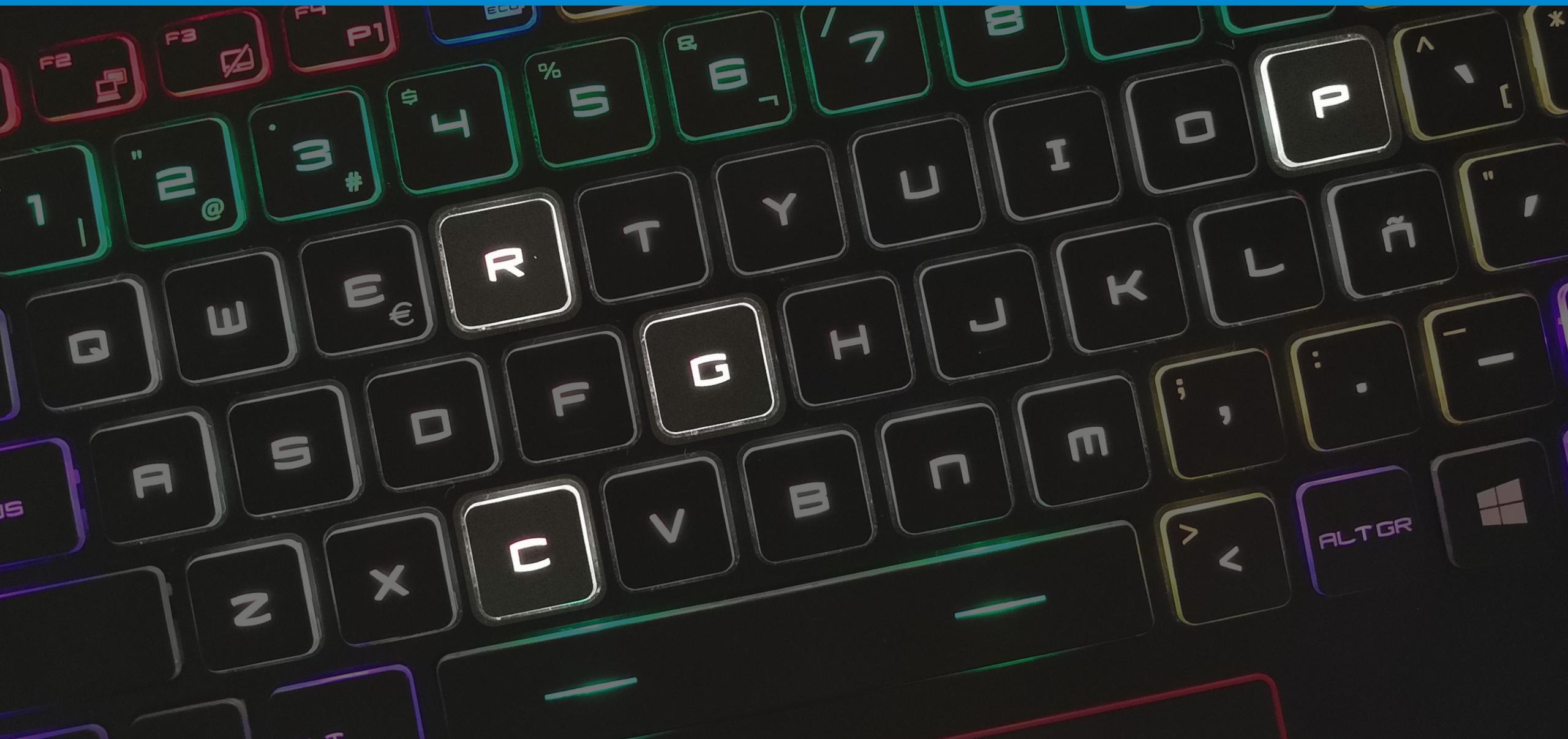
Proyecto de Google para sustituir
la infraestructura actual: Stubby

- El RPC de Google
- Objetivos claros: microservicios (y unos cuantos más)
- Construido sobre Protocol Buffers (protobuf)

<https://grpc.io/blog/principles/>



¿Cómo usamos gRPC en .net core?



¿Cómo usamos gRPC en .net core?



Para poder empezar con la demo, primero debemos saber.....

Especificación de Protobuf

<https://developers.google.com/protocol-buffers/docs/proto3>

A tener en cuenta:

- Ojo con los tipos: haz caso a las recomendaciones.
- La definición de los métodos siempre tienen que tener un mensaje de entrada y retorno, y no valen los tipos escalares.
- Acuérdate de usar imports y namespaces para organizar tu código.

¿Cómo usamos gRPC en .net core?



Demo



¿Cuándo usar gRPC?



¿Cuándo usar gRPC?



Tenemos una orientación y arquitectura clara de microservicios

Entornos de alta carga de trabajo con un plus de rendimiento

Comunicación entre Back2Back, Tool2Back, Apps móviles

Entornos desarrollo heterogéneos

Realtime, IoT, Streaming (Uni & bidireccional)

¿Cuándo no usarlo?



Formato binario (nativo en gRPC) no me conviene

Lenguajes de programación sin soporte en el tooling

Cuando el entorno de ejecución no favorece la introducción de una tecnología nueva

*Necesito soporte en navegadores



¿Qué más puedo hacer?



Optimización con MessagePack, bson incluso protobuf

Mitigar lenguajes sin soporte: OpenAPI

Rendimiento de Web.API:

- HTTP/2
- Modo InProc (disponible desde ASP.NET Core 2.2)
- Balanceadores

Referencias



gRPC en General

Sitio principal gRPC: <https://grpc.io>

Especificación protobuf: <https://developers.google.com/protocol-buffers/docs/proto3>

Mensajes built-in protobuf: <https://github.com/protocolbuffers/protobuf/tree/master/src/google/protobuf>

Web tool para probar servicios gRPC: <http://ui.grpcs.top/>

Ecosistema de proyectos: <https://github.com/grpc-ecosystem/awesome-grpc>

gRPC + JSON: <https://grpc.io/blog/grpc-with-json/>

.Net Core

gRPC en ASP.NET Core 3.0: <https://docs.microsoft.com/en-us/aspnet/core/grpc/?view=aspnetcore-3.0>

Consideraciones de seguridad: <https://docs.microsoft.com/en-us/aspnet/core/grpc/security?view=aspnetcore-3.0>

ASP.NET Core gRPC for WCF Developers: <https://github.com/dotnet/docs/blob/master/docs/architecture/grpc-for-wcf-developers>

Repositorio gRPC-dotnet: <https://github.com/grpc/grpc-dotnet>

Otras referencias:

Sitio principal HTTP/2: <https://http2.github.io/>

SPDY en Wikipedia: <https://es.wikipedia.org/wiki/SPDY>

QUIC en Wikipedia: <https://en.wikipedia.org/wiki/QUIC>

ASP.NET Core InProc IIS: <https://weblog.west-wind.com/posts/2019/Mar/16/ASPNET-Core-Hosting-on-IIS-with-ASPNET-Core-22>

ASP.NET Core HTTP/2 in Kestrel: <https://devblogs.microsoft.com/aspnet/asp-net-core-2-2-0-preview1-http-2-in-kestrel/>

ASP.NET Core HTTP comparison: <https://github.com/danroth27/Http2Test>

Benefits of REST APIs With HTTP/2: <https://dzone.com/articles/benefits-of-rest-apis-with-http2>

JWT Builder: <http://jwtbuilder.jamiekurtz.com/>



github sources



**Gracias por tu
atención**