ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

ECYS | DTT
DESARROLLO DE TRANSFERENCIA TECNOLÓGICA

Día, Fecha:	Viernes, 05/04/2024
Hora de inicio:	15:40 - 17:20

Sistemas Operativos 1 “A”

Jhonathan Daniel Tocay Cotzojay

Primer Semestre 2024

Sistemas Operativos 1

Jhonathan Daniel Tocay Cotzojay

Datos del Auxiliar

- Jhonathan Tocay
- Correo Electrónico: 2878571900109@ingenieria.usac.edu.gt
- Asunto: [SO1]Asunto
- Usuario GitHub: JhonathanTocay2020
- Repositorio: https://github.com/JhonathanTocay2020/Laboratorio_SO1_1S2024.git

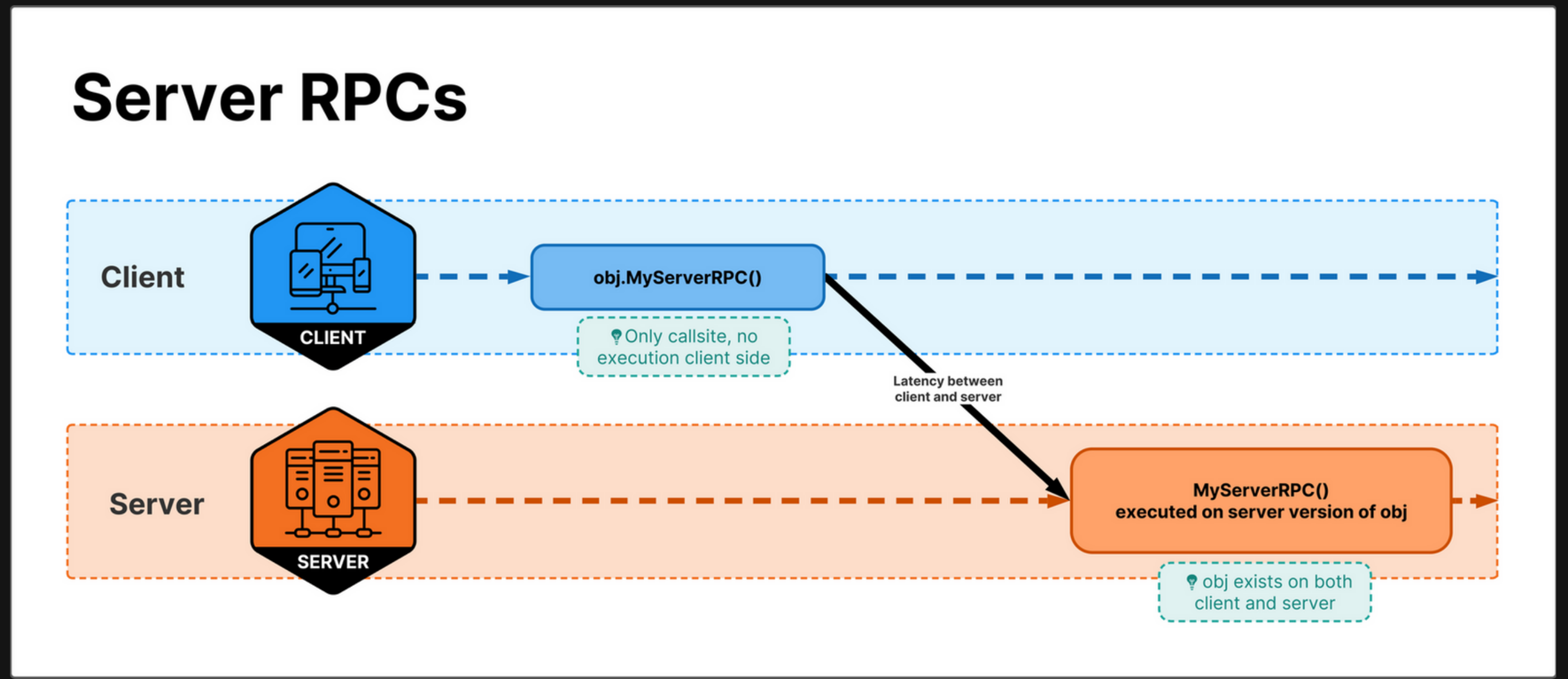
Agenda

- AVISOS
- Lectura del Proyecto 2
- Clase GRPC
- Tarea 4
- Foro 10
- Formulario de Asistencia

Llamadas a Procedimientos Remotos (RPC)

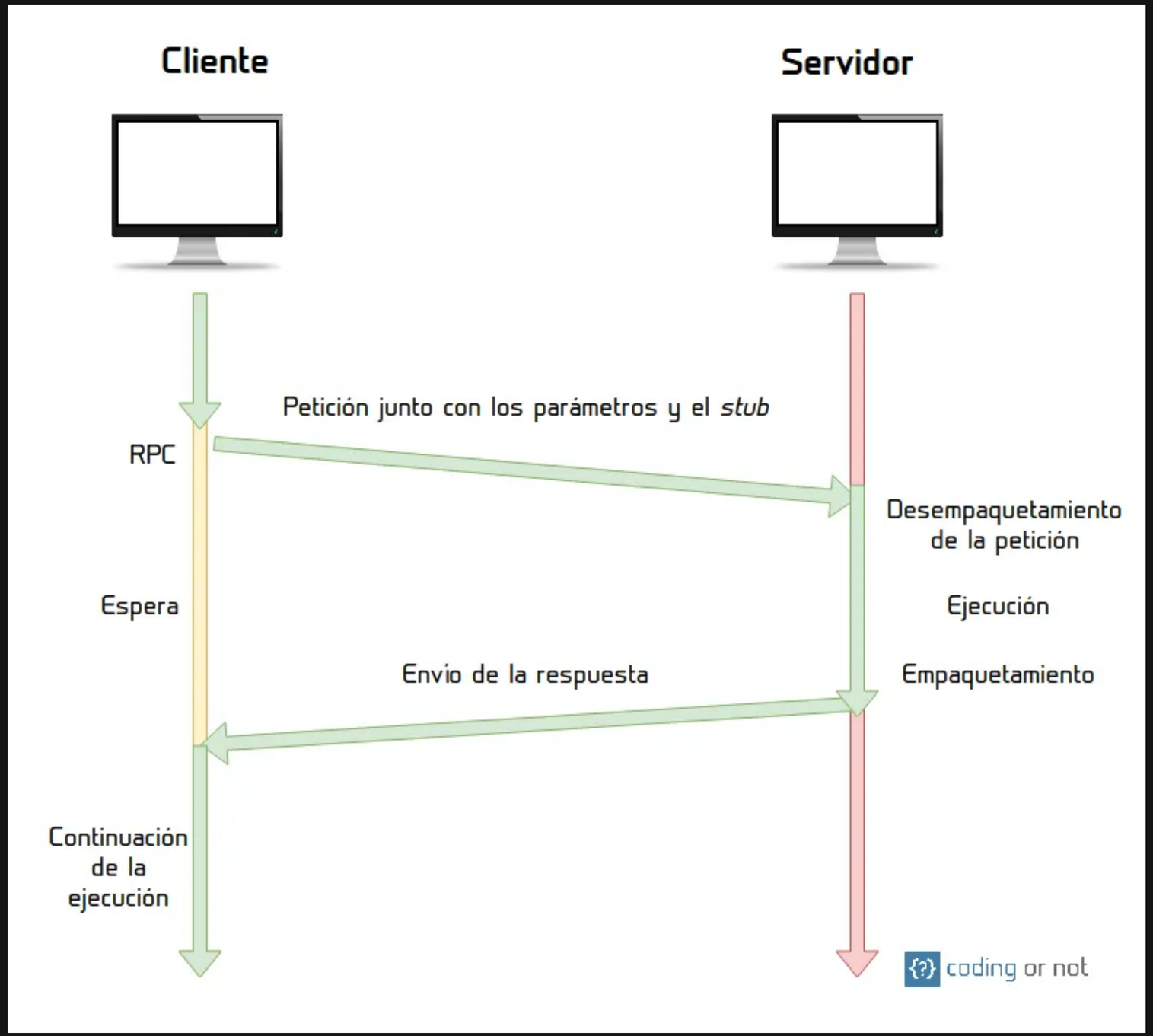
¿Qué es?

Es una técnica que utiliza el modelo cliente-servidor para ejecutar tareas en un proceso diferente como podría ser en una computadora remota. A veces solamente se le llama como llamada a una función o subrutina remota.



¿Cómo funciona?

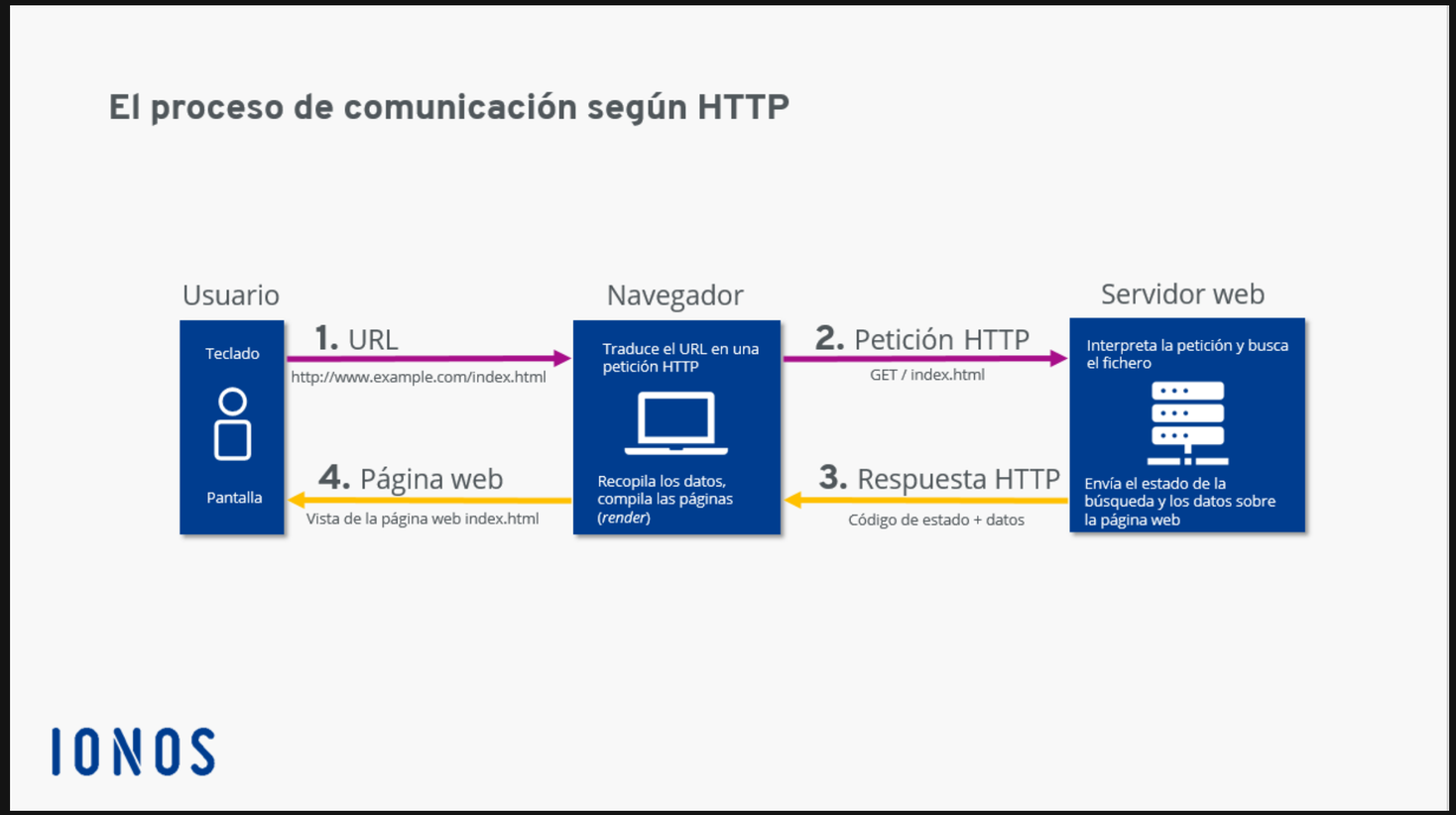
1. **Cliente hace la llamada al procedimiento remoto por un mensaje a través de la red.**
2. **El servidor recibe la petición y desempaqueta el mensaje para extraer información necesaria para realizar la tarea.**
3. **El servidor ejecuta la tarea.**
4. **El servidor crea un mensaje de respuesta para el cliente.**
5. **El cliente recibe y desempaqueta el mensaje de respuesta.**


Cliente
Servidor
Petición junto con los parámetros y el *stub*
RPC
Desempaquetamiento de la petición
Espera
Ejecución
Envio de la respuesta
Empaquetamiento
Continuación de la ejecución
{?} coding or not

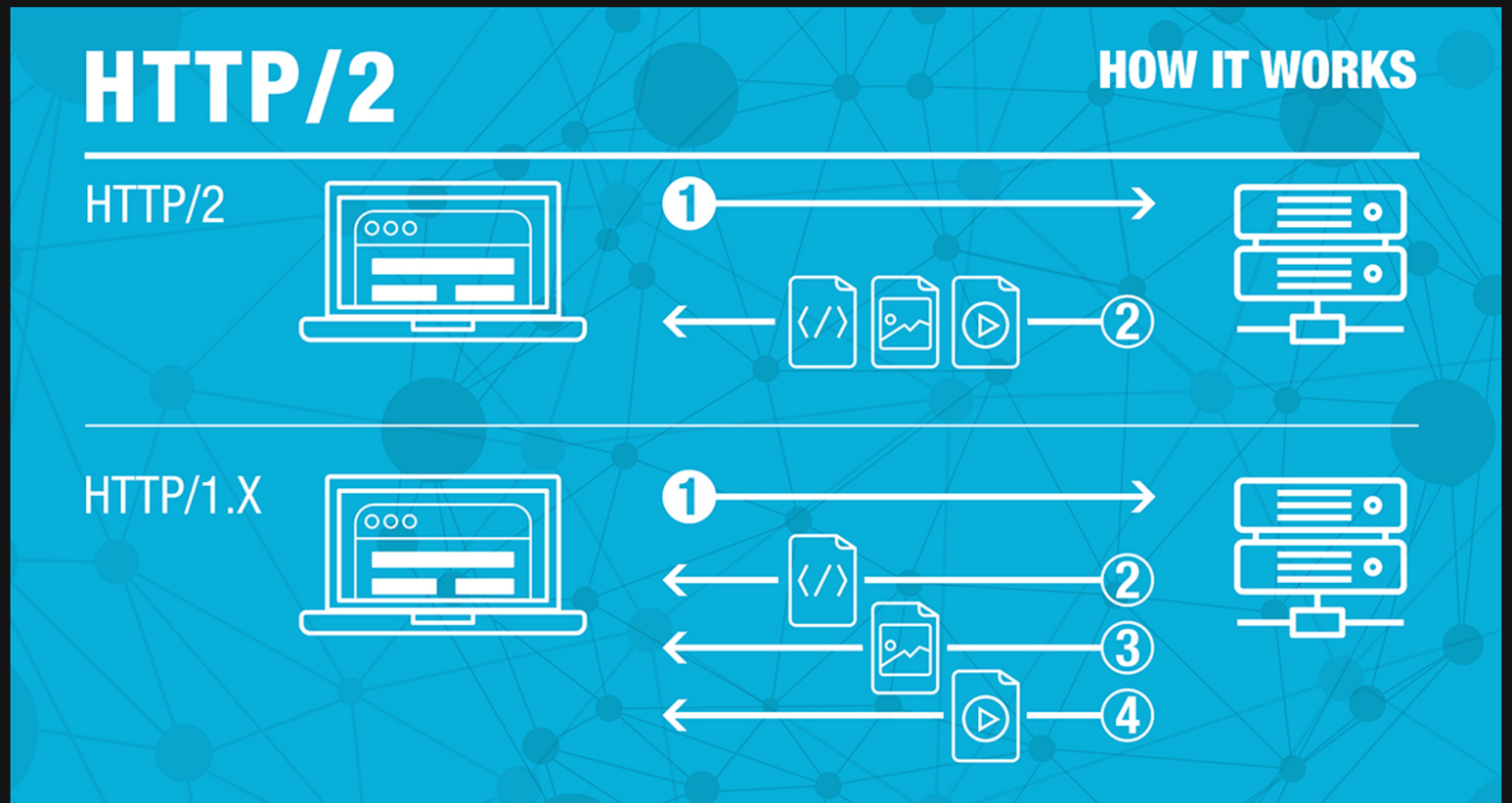
HTTP/2

HTTP

HTTP es un protocolo de transferencia de hipertexto y vive en la capa de aplicación diseñado para transferir información entre los dispositivos conectados de la red.

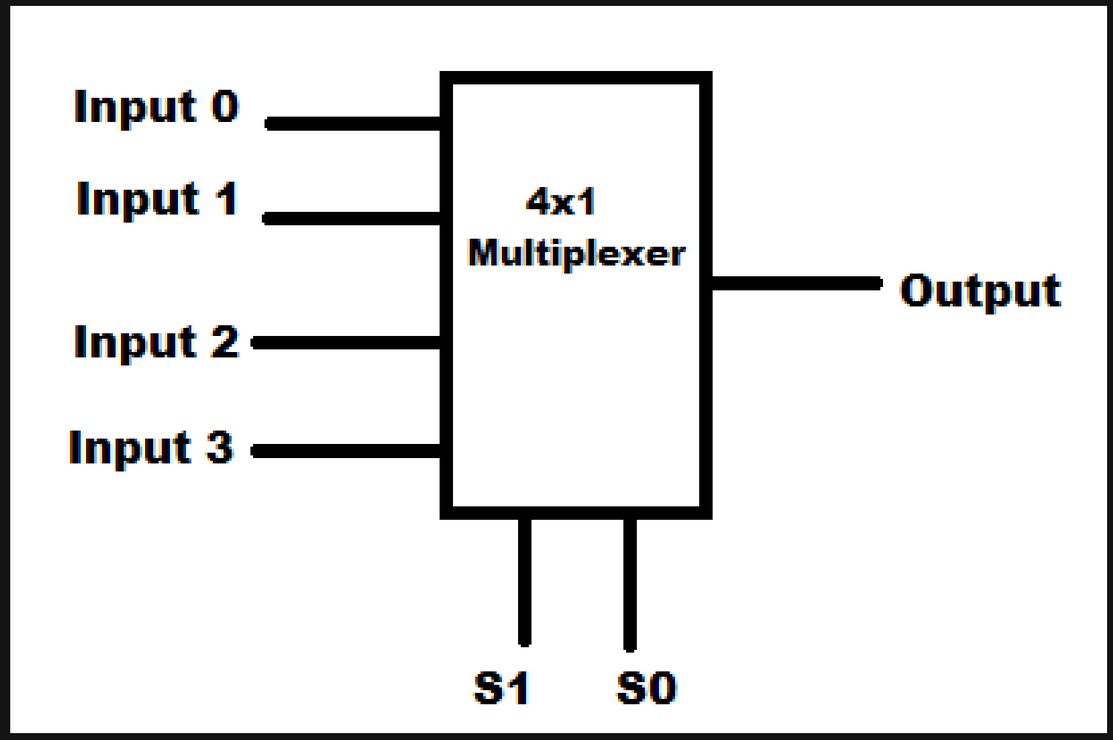


Nace con la intención de resolver los problemas que surgen de HTTP que tiene cada vez más complicado responder a mensajes o contenidos cada vez más grandes y complejos. Por lo que, entre otras muchas mejoras, podemos destacar la de una mayor velocidad en la transferencia de información entre servidor y navegador.

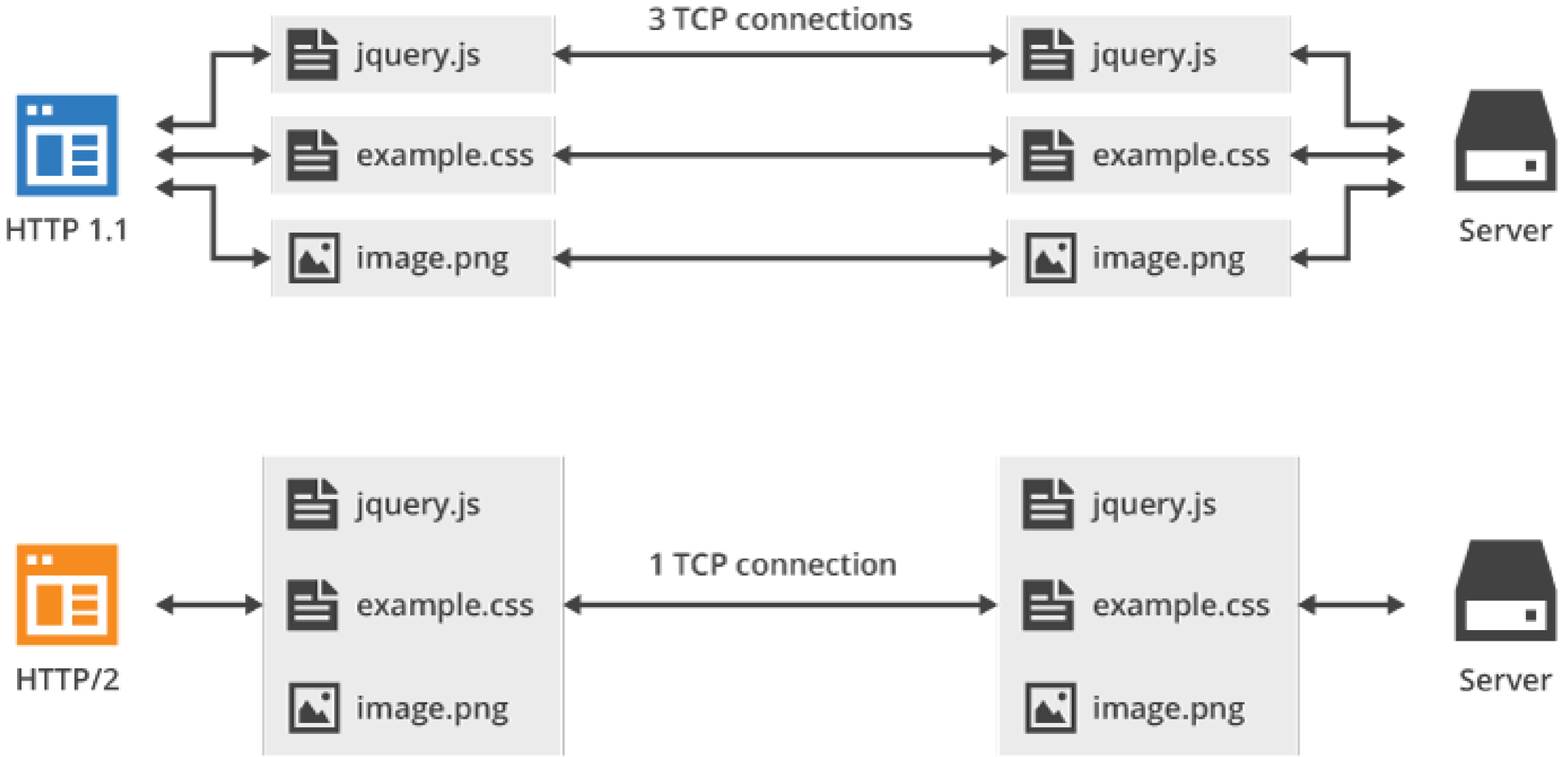


Características

- HTTP2 es binario y no textual, por lo que los mensajes se procesan de manera más rápida.
- Es multiplexado y paralelo, por lo que permite procesar muchas peticiones al mismo tiempo.



Multiplexing



Beneficios

- Mejor rendimiento web
- Mejor rendimiento en móviles
- Descenso en el coste de internet
- Ayuda a expandir el internet
- Navegación más segura
- Mejorar experiencia de usuario

gRPC

Es una implementación de RPC diseñado originalmente por Google

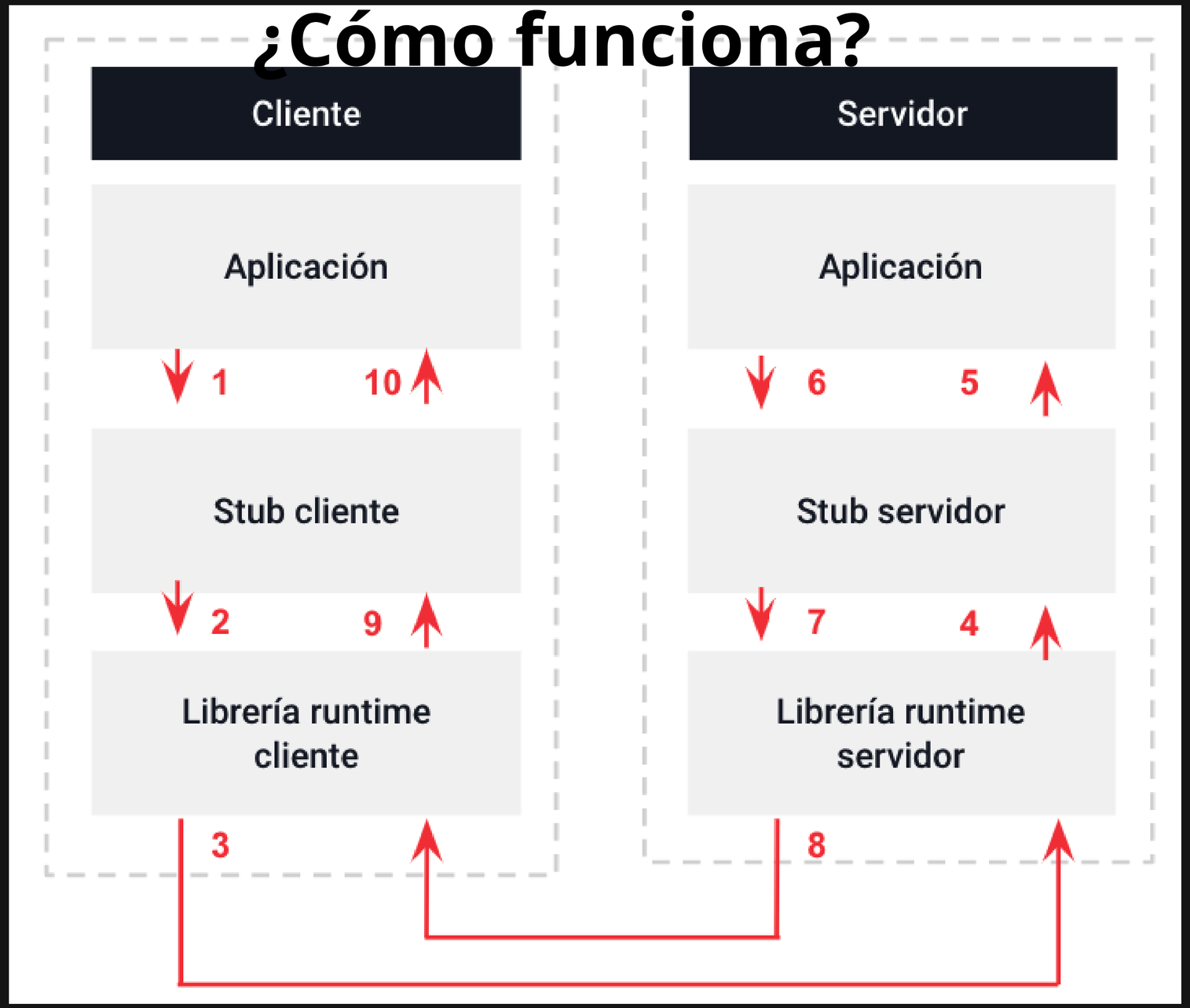
Se emplea en comunicaciones cliente-servidor distribuidas por su eficiencia gracias a la ingeniería de procesos basada en RPC. Este mecanismo permite la comunicación de una forma tan sencilla como si se tratara de una comunicación local entre procesos de una misma máquina.

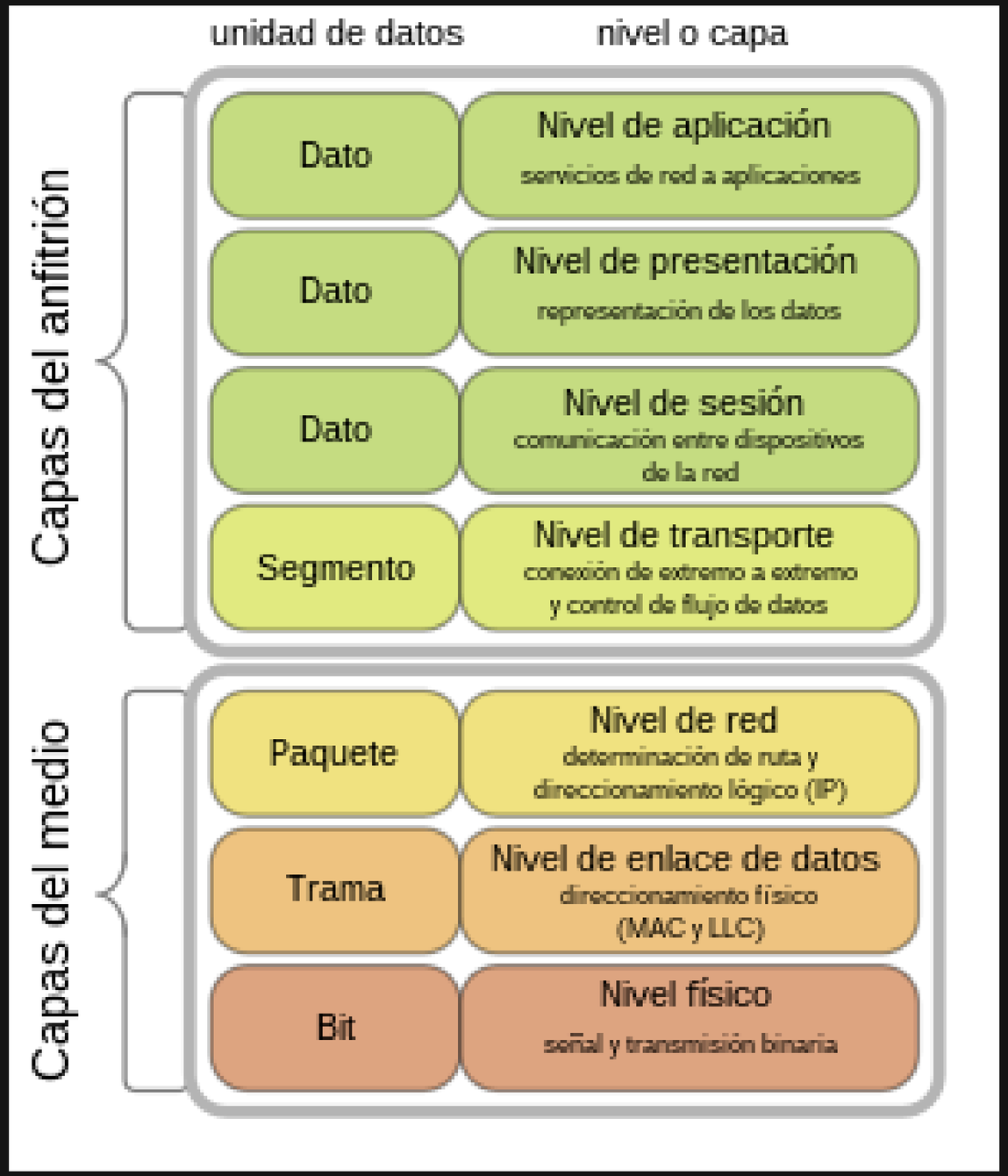
Características

- **Sencillez: Son llamadas a funciones y no requiere mucho código.**
- **Flexibilidad: Permite trabajar con muchos lenguajes de programación.**
- **Rendimiento: Es una alternativa atractiva para llamadas entre servicios.**

Postman ha incluido gRPC para su funcionamiento

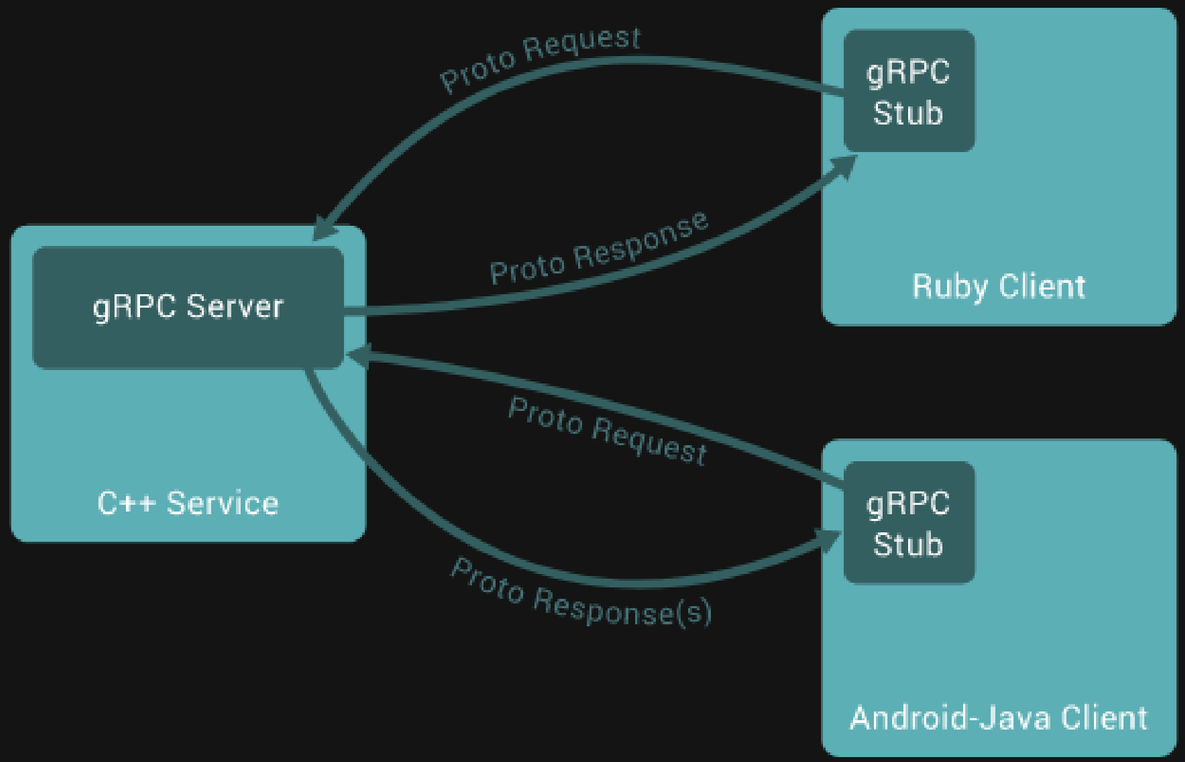
¿Cómo funciona?




unidad de datos
nivel o capa
Capas del anfitrión
Dato
Nivel de aplicación
servicios de red a aplicaciones
Dato
Nivel de presentación
representación de los datos
Dato
Nivel de sesión
comunicación entre dispositivos de la red
Segmento
Nivel de transporte
conexión de extremo a extremo y control de flujo de datos
Capas del medio
Paquete
Nivel de red
determinación de ruta y direccionamiento lógico (IP)
Trama
Nivel de enlace de datos
direccionamiento físico (MAC y LLC)
Bit
Nivel físico
señal y transmisión binaria

Diferencias con RPC

- **Uso de HTTP/2 para envío de datos por la capa de transporte.**
- **Utilización de Protocol Buffers para gestión de estructura y distribución de datos.**



Protobuf

Son un formato binario que facilita el almacenamiento e intercambio de datos en aplicaciones.

- **Serializa de las estructuras de datos, es decir, es el formato subyacente para el intercambio de mensajes.**
- **Describe la interfaz de comunicación, es decir, actúa como el contrato intermediario que permite que el cliente y el servidor se entiendan.**

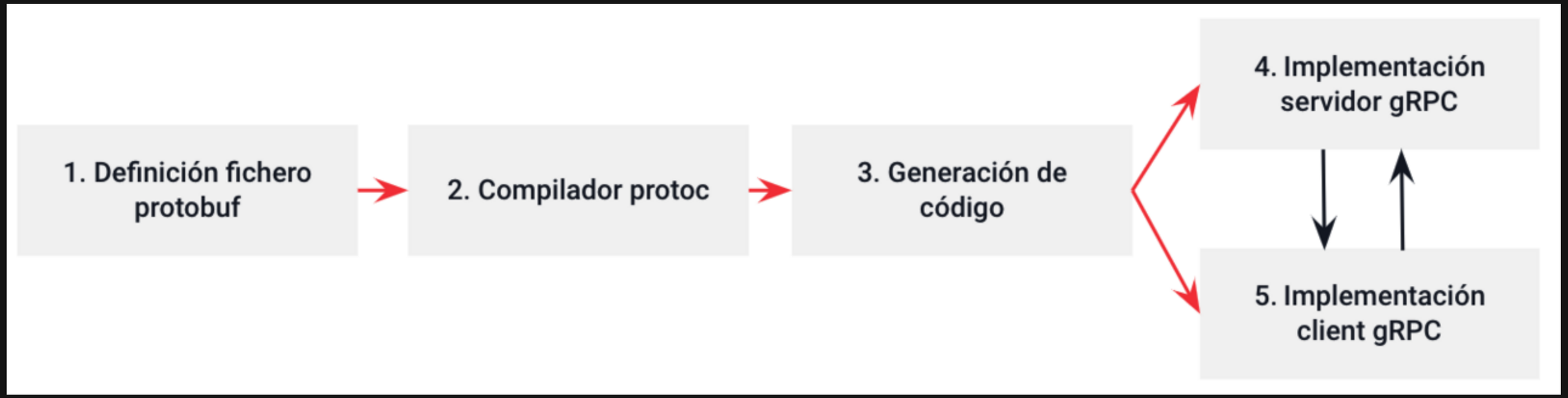
```
syntax = "proto3";
package com.book;

message Book {
    int64 isbn = 1;
    string title = 2;
    string author = 3;
}
```

```
message GetBookRequest {
    int64 isbn = 1;
}
```

Pasos para trabajar con gRPC y Protobuf

1. **Definición de servicio en archivo .proto**
2. **Compilación del archivo con protoc**
3. **Generación de código a lenguaje de programación**
4. **Implementar el stub de servidor**
5. **Implementación del stub cliente**


1. Definición fichero protobuf
2. Compilador protoc
3. Generación de código
4. Implementación servidor gRPC
5. Implementación client gRPC

gRPC vs REST

	gRPC	REST
Protocolo	HTTP/2 -> rápido	HTTP/1.1 -> lento
Payload	Protobuf -> binario, no legible + rápido + pequeño	JSON -> texto plano, legible + lento + grande
Contrato API	Estricto y requerido -> .proto	Flexible y opcional -> OpenAPI
Diseño API	Orientado al qué (operaciones) -> diseño libre	Orientado a CRUD -> diseño complejo
Generación código	Protocols Buffers -> integrado para cualquier lenguaje	OpenAPI -> OpenAPI -> terceros
Streaming	Bidireccional y soporta asincronía	Cliente-servidor y solo síncrono

¿Dónde usar gRPC?

1. **Comunicación punto a punto en tiempo real.**
2. **Comunicación eficiente**
3. **Entorno multilingües**
4. **Soluciones de integración**

Tráfico Web

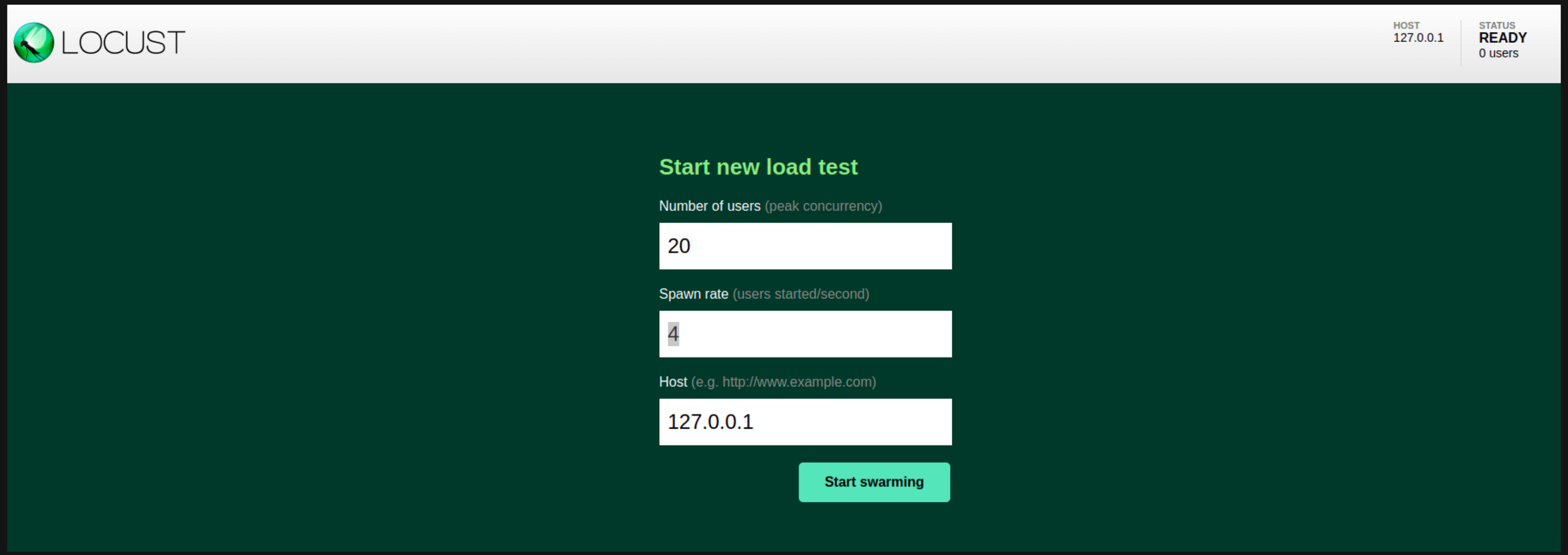
¿Qué es?

Es la cantidad de datos enviados y recibidos por los visitantes de una web.
Este dato se determina por el número de visitantes y de páginas que visitan.

Locust Python

Una herramienta de prueba de rendimiento fácil de usar, programable y escalable. Se define el comportamiento de sus usuarios en el código Python normal, en lugar de estar atrapado en una interfaz de usuario o en un lenguaje específico de dominio restrictivo. Esto hace que Locust sea infinitamente expandible y muy amigable para los desarrolladores

Interfaz Gráfica de Locust



Parámetros de Locust

1. Número de usuarios.

2. Tasa de generación: usuarios iniciados por segundo.

3. Host: si especificamos el host en el código de Python, automáticamente se llenará aquí; de lo contrario, también podemos agregar desde la interfaz de usuario.

Resultados

Type	Name	# Requests	# Fails	Median (ms)	90%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
Odoo JsonRPC	common : login	20	0	160	230	260	147	76	257	28	2.22	0
Odoo JsonRPC	product.product : read	13	0	1600	3100	3200	1788	654	3213	336	1.11	0
Odoo JsonRPC	product.product : search	13	0	48	93	170	59	17	166	336	1.22	0
Odoo JsonRPC	res.partner : read	21	0	3100	5200	5500	3142	957	5507	760	1.33	0
Odoo JsonRPC	res.partner : search	22	0	47	120	130	66	11	132	760	1.89	0
	Aggregated	**89**	**0**	**160**	**4800**	**5500**	**1060**	**11**	**5507**	**472**	**7.78**	**0**