

# Acelerando 'Microservicios' {

[09.03.2022]

< Python + GRPC />

# Ramón Molina Milla

# 🙌 Engineering Manager @Nextail

}

# Contenido {

01

¿Qué es GRPC?

Introducción sobre el framework

02

Explicando la arquitectura

Dibujemos algunas cajas

03

Ejercicio práctico

dev01:~\$ code .

}

# Qué es 'GRPC';

Es un protocolo `Remote Procedure Call`, bajo licencia `open-source` creado por `Google` y que usa para comunicar sus microservicios.

- Funciona sobre `HTTP/2` y datos binarios
- Enfocado a conseguir rendimiento, bajo consumo de CPU, ancho de banda y latencia.
- Bidireccionalidad
- Serialización `Proto3`
- Autenticación incorporada

# Contenido {

01

¿Qué es GRPC?

Introducción sobre el framework

02

Explicando la arquitectura

Dibujemos algunas cajas

03

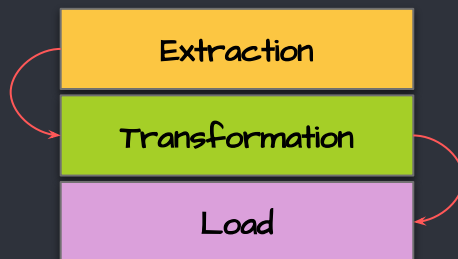
Ejercicio práctico

dev01:~\$ code .

}

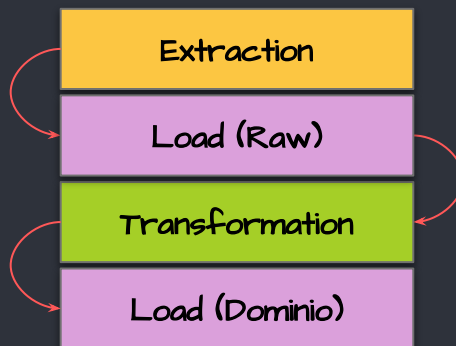
# Ejemplo 'Arquitectura';

Comencemos con una aplicación **ETL** estándar que tiene que extraer una información de un fichero CSV, aplicarle una lógica de negocio y posteriormente almacenarla en en una base de datos.



# Ejemplo 'Arquitectura';

Pero es preferible usar un modelo **ELTL** que nos permite almacenar la información RAW tal y como la recibe nuestro sistema para almacenarla posteriormente transformada y estandarizada al dominio de nuestro producto.



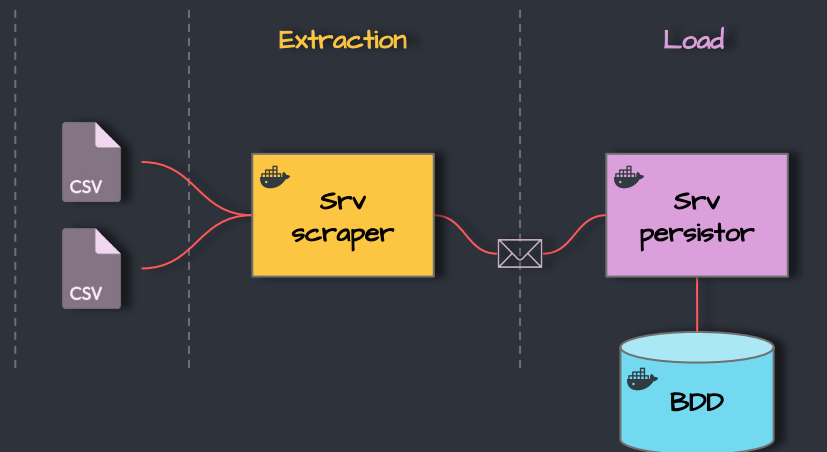
# Ejemplo 'Arquitectura';

Simplificación Extracción y Carga para desarrollar durante el ejercicio.



# Ejemplo 'Arquitectura';

Simplificación Extracción y Carga para desarrollar durante el ejercicio





# Contenido {

01

¿Qué es GRPC?

Introducción sobre el framework

02

Explicando la arquitectura

Dibujemos algunas cajas

03

Ejercicio práctico

dev01:~\$ code .

}

# Ejercicio 'Práctico' {

01

Envío básico

Casi todo puede ser un  
string

02

Unary connection

Enviando un solo objeto por  
conexión

03

Stream connection

Reutilizando un flujo HTTP2  
para enviar varios objetos

}

# Bad 'Practices' {



Modificar MAX\_GRPC

Ajustar el tamaño  
mínimo y máximo de  
cada mensaje.



unsecure\_channel

Securizar una parte  
no expuesta de  
nuestra aplicación

}