

Microservicios Ejemplo de Flujo



Adrian Pedro Zelada Torrez

[Follow](#)

May 16, 2018 · 3 min read

En el anterior post ([Microservicios Conceptos](#)) hablamos un poco de que son los microservicios , también de la reglas que se debe de cumplir para realizar un microservicio. En este Post desarrollaremos un pequeño ejemplo que sera una pequeña aplicación para poder ver información de cursos o libros.

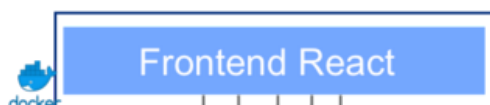
Ejemplo

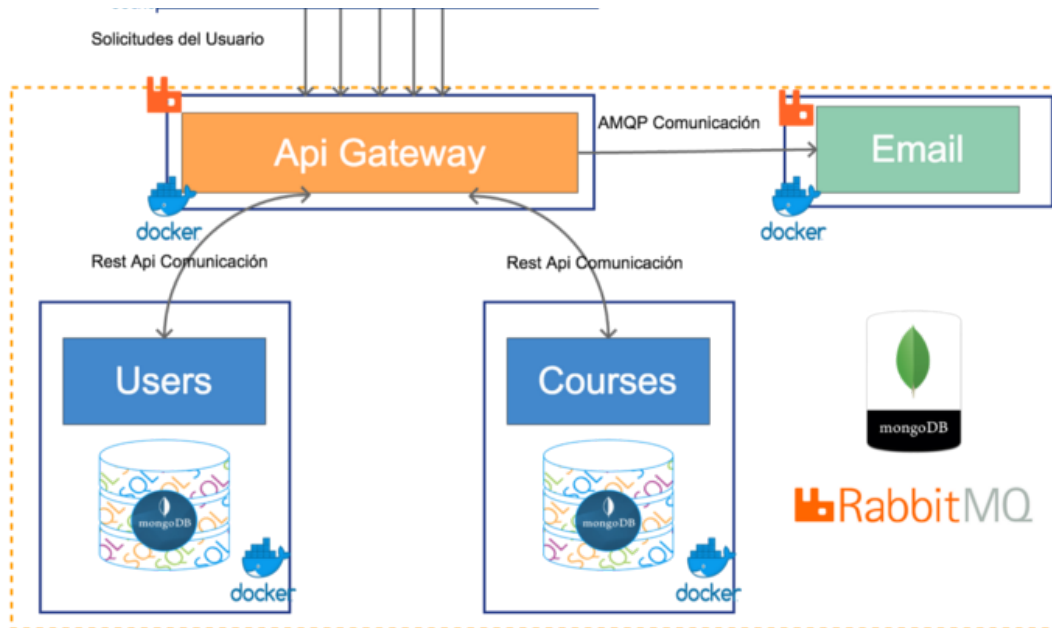
Se realizará una Tienda de Libros o Cursos donde se mostraran los artículos disponibles, el cliente podrá obtener información de cualquier artículo ingresando su email. Un cliente podrá ver los artículos que obtuvo solo proporcionando su correo.

Diagrama

Para este pequeño ejemplo utilizaremos cuatro microservicios :

- **Api Gateway.-** Tiene la tarea de comunicarse con el frontend y según las peticiones de el usuario se comunica con los otros microservicios.
- **Users y Courses.-** Dos microservicios que estan en un nivel atómico estos solo se encargan de ellos. Notemos que cada microservicio tienes su propia base de datos.
- **Email.-** Microservicio encargado solo de enviar correos a los usuarios, notemos que no tiene una comunicación REST API como en los anteriores microservicios, para este utilizaremos el protocolo AMQP ayudandonos con RabbitMQ, esto nos dará la capacidad de poner en cola las peticiones que le estén llegando.





Flujo

Se mostrará una variedad de libros o cursos de los cuales el usuario podrá seleccionar alguno, si desea obtener información de algún artículo entonces procedemos a pedirle su Correo para mandarle la información del artículo que selecciono ,si el Usuario es nuevo entonces se le enviara un correo de bienvenida a la aplicación, posteriormente se le enviaremos la información del artículo que selecciono.

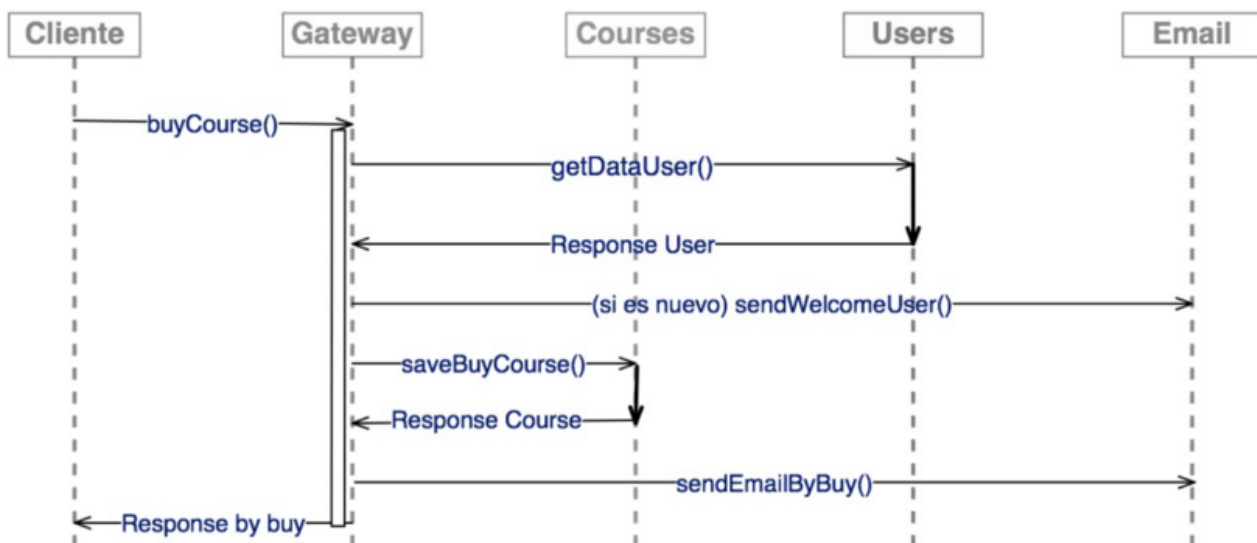


Diagrama de Secuencia para Solicitar información de un artículo

Descripción

- **BuyCourse:** Desde el cliente se hace la solicitud de información de un artículo a el servicio Gateway.
- **GetDataUser:** El servicios Gateway solicita los datos del Usuario al servicio Users enviando el email del cliente ,este servicio verifica si el usuario existe ,si no existe el usuario crea uno nuevo con el email proporcionado , teniendo el usuario respondemos a el servicio Gateway enviando el usuario.
- **SendWelcomeUser:** El servicio Gateway verifica si el usuario obtenido a sido creado recientemente si es así se envía un correo de Bienvenida a la Tienda.
- **SaveBuyCourse:** El servicio Gateway se comunica con el servicio Courses ,este guarda los datos de la solicitud de información de el articulo por usuario, al finalizar el guardado del curso se manda una respuesta de exitosa a Gateway.
- **SendEmailByBuy:** Una vez realizada la solicitud se realiza el envio de un email al cliente con la información de el artículo.

Este solo es un flujo de una funcionalidad que se puede realizar con la aplicación. Seguramente pensaron pero si esta aplicación se podría hacer de una forma mas sencilla con una arquitectura monolítica y yo les diría “ Pues si ustedes tienen razón” pero recordemos que este es un ejemplo, generalmente la arquitectura de microservicios se aplica para aplicaciones de gran crecimiento esto nos ayudara a modularizar cada funcionalidad que necesitemos ,y cada modulo sera independiente a cualquier otro modulo.

Desarrollo

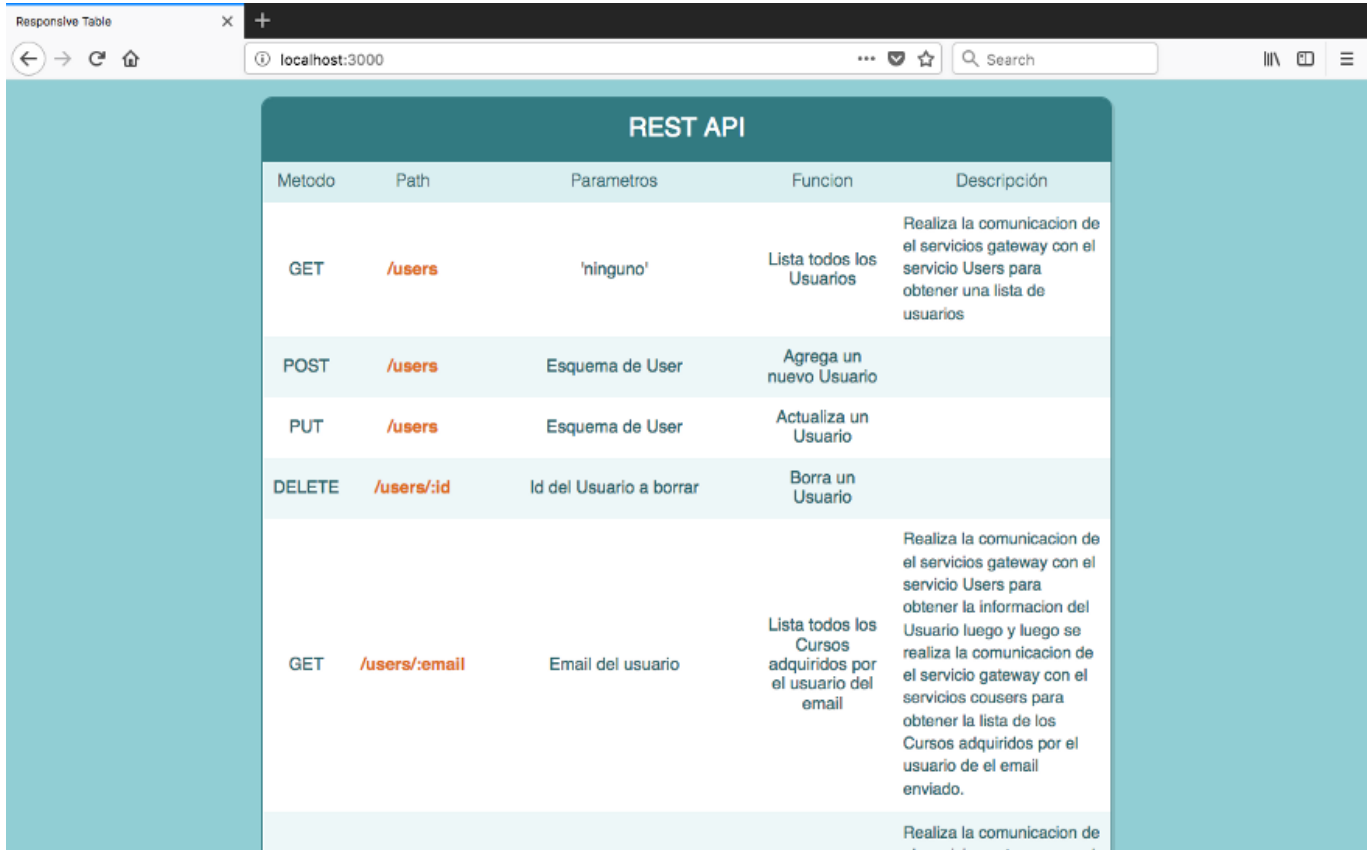
Para realizar este objetivo utilizaremos las tecnologias :

- Docker
- Nodejs
- MongoDB
- RabbitMQ

Y para la realización del frontend:

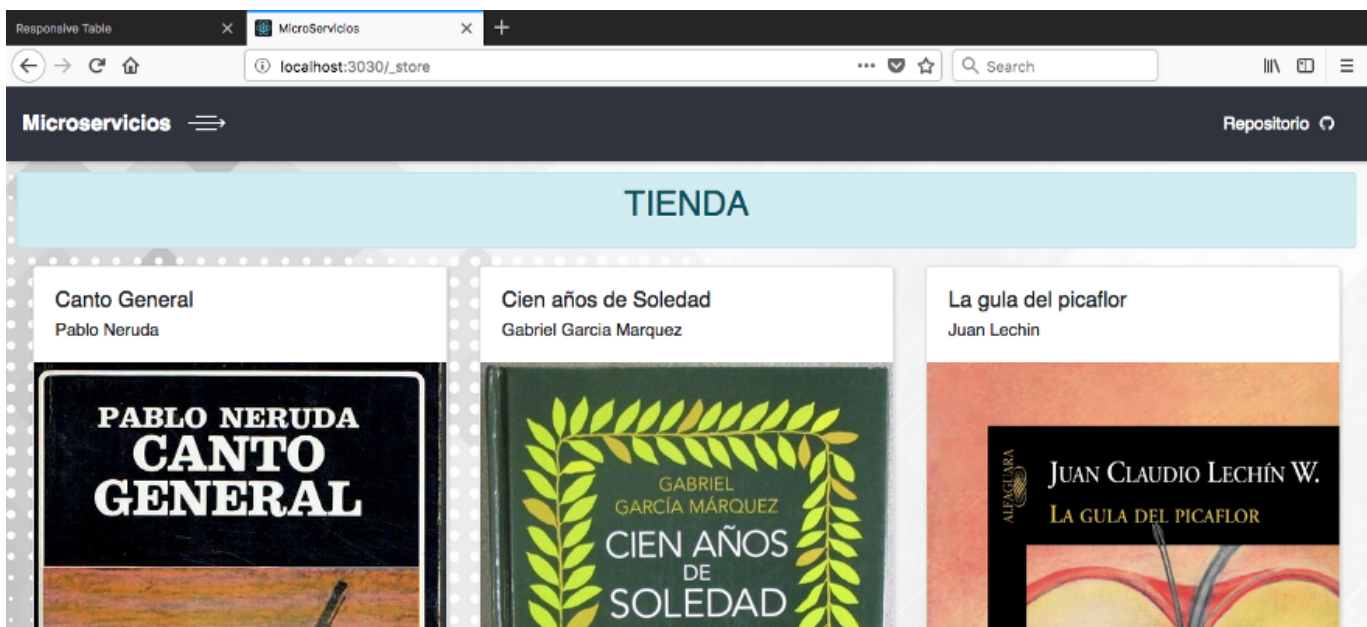
- React

La lista de End Points del servidor node están <http://localhost:3000>



REST API				
Metodo	Path	Parametros	Funcion	Descripción
GET	/users	'ninguno'	Lista todos los Usuarios	Realiza la comunicacion de el servicios gateway con el servicio Users para obtener una lista de usuarios
POST	/users	Esquema de User	Agrega un nuevo Usuario	
PUT	/users	Esquema de User	Actualiza un Usuario	
DELETE	/users/:id	Id del Usuario a borrar	Borra un Usuario	
GET	/users/:email	Email del usuario	Lista todos los Cursos adquiridos por el usuario del email	Realiza la comunicacion de el servicios gateway con el servicio Users para obtener la informacion del Usuario luego y luego se realiza la comunicacion de el servicio gateway con el servicios cousers para obtener la lista de los Cursos adquiridos por el usuario de el email enviado.
				Realiza la comunicacion de el servicios gateway con el

Y la aplicación se despliega en el http://localhost:3030/_store





Puedes ver la aplicación [GITHUB](#)

Y como pudimos ver la arquitectura de microservicios es mas compleja, pero nos da mas facilidad para poder escalar la aplicación al nivel que deseemos.

[Microservices](#) [Rabbitmq](#) [Mongodb](#) [Docker Compose](#) [Docker](#)

[About](#) [Help](#) [Legal](#)

Get the Medium app

