

1- Instanciación del sistema

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4$ mkdir -p socks-demo
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4$ cd socks-demo/
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ git clone https://github.com/microservices-demo/microservices-demo.git
Cloning into 'microservices-demo'...
remote: Enumerating objects: 10197, done.
remote: Total 10197 (delta 0), reused 0 (delta 0), pack-reused 10197
Receiving objects: 100% (10197/10197), 52.95 MiB | 1.08 MiB/s, done.
Resolving deltas: 100% (6208/6208), done.
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ cd microservices-demo/
```

```
709515475419: Already exists
b49495cd3b4e: Pull complete
0a724dd10a63: Pull complete
a9f88cb33c46: Pull complete
Digest: sha256:6740c0a9afd5f4f643504728104a74501c4d78e6f4e7ded2a7caa351b417df99
Status: Downloaded newer image for weaveworksdemos/user:0.4.4
Pulling user-db (weaveworksdemos/user-db:0.4.0)...
0.4.0: Pulling from weaveworksdemos/user-db
75a822cd7888: Already exists
3bf369f658b6: Pull complete
7d7cb343d20e: Pull complete
73a933a908f7: Pull complete
658569c28c55: Pull complete
124a8bf940da: Pull complete
7c19551df503: Pull complete
a18347fe18d9: Pull complete
53e710c6ec29: Pull complete
44a8bd3d5a66: Pull complete
6f31a4ac728b: Pull complete
d883cd3a4889: Pull complete
Digest: sha256:b43f0f8a76e0c908805fcec74d1ad7f4af4d93c4612632bd6dc20a87508e0b68
Status: Downloaded newer image for weaveworksdemos/user-db:0.4.0
Pulling user-sim (weaveworksdemos/load-test:0.1.1)...
0.1.1: Pulling from weaveworksdemos/load-test
af65d799fdb4: Pull complete
9fcd4fbb580: Pull complete
cb94c92135ba: Pull complete
02e3537a5fe2: Downloading [=====] 88.11MB/96.94MB
02e3537a5fe2: Pull complete
fb840ff3e7ea: Pull complete
14c7209d4eea: Pull complete
982c5cda19ad: Pull complete
97746269a8d1: Pull complete
242513f1174a: Pull complete
231018400bd8: Pull complete
6571f6752797: Pull complete
Digest: sha256:536d46f8c867e4ff4c3ed69848955b487f9bec060539c169f190fe522650e5cd
Status: Downloaded newer image for weaveworksdemos/load-test:0.1.1
Creating docker-compose_orders-db_1 ... done
Creating docker-compose_catalogue-db_1 ... done
Creating docker-compose_user-db_1 ... done
Creating docker-compose_carts_1 ... done
Creating docker-compose_catalogue_1 ... done
Creating docker-compose_rabbitmq_1 ... done
Creating docker-compose_user_1 ... done
Creating docker-compose_payment_1 ... done
Creating docker-compose_user-sim_1 ... done
Creating docker-compose_shipping_1 ... done
Creating docker-compose_front-end_1 ... done
Creating docker-compose_queue-master_1 ... done
Creating docker-compose_orders_1 ... done
Creating docker-compose_carts-db_1 ... done
Creating docker-compose_edge-router_1 ... done
```

Register



Username

First name

Last name

Email

Password

 Register

OFFER OF THE DAY Buy 1000 socks, get a shoe for free!

Login | Register



HOME

CATALOGUE ▾



WE LOVE SOCKS!

Fun fact: Socks were invented by woolly mammoths to keep warm. They died out because stupid humans had to cut their legs off to get their socks.

BEST PRICES

We price check our socks with trained monkeys back at the office.


100% SATISFACTION GUARANTEED

Free returns on most items. Hamsters are non-returnable once spoken to.

HOT THIS WEEK

Filters

- ☐ brown
- ☐ geek
- ☒ formal
- ☒ blue
- ☐ skin
- ☐ red
- ☐ action
- ☐ sport
- ☐ black
- ☐ magic
- ☐ green

 Apply

Customer section

☰ My orders

My orders

Your orders in one place.

If you have any questions, please feel free to [contact us](#), our customer service center is working for you 24/7.

Order	Date	Total	Status	Action
# 65257ea530c49e0007457569	2023-10-10 16:41:09	\$ 22.99	Shipped	View

2- Investigación de los componentes

2.1)

queue-master: Este contenedor sirve como una cola maestra para la comunicación entre microservicios, pero no tiene un punto de entrada visible para los usuarios finales. Es esencial para la comunicación interna de los servicios.

user-db: Aloja la base de datos para la gestión de usuarios. Proporciona un punto de entrada a través de la red para los servicios que requieren acceso a los datos de usuario.

payment: Se encarga del procesamiento de pagos. No tiene un punto de entrada explícito para los usuarios, ya que se utiliza internamente para procesar pagos en otras partes del sistema.

catalogue: Administra la información del catálogo de productos disponibles. Proporciona puntos de entrada en /list y /detail para listar productos y obtener detalles sobre un producto específico, respectivamente.

rabbitmq: Actúa como un intermediario para la mensajería entre los servicios. No tiene un punto de entrada directo para los usuarios, pero permite la comunicación entre los microservicios.

edge-router: Este contenedor es responsable de dirigir las solicitudes de los usuarios a los servicios correspondientes. Es un punto de entrada importante para los usuarios y puede enrutar solicitudes a varios servicios, como "front-end" y otros.

shipping: Gestiona las opciones de envío de productos. Proporciona puntos de entrada en /methods para listar métodos de envío y /quote para obtener cotizaciones de envío.

catalogue-db: Aloja la base de datos que almacena la información del catálogo de productos. No tiene un punto de entrada explícito, ya que se utiliza internamente por el servicio de catálogo.

front-end: Este es el contenedor que aloja la interfaz de usuario principal del sistema y es el punto de entrada para los usuarios finales. Los usuarios pueden acceder al sistema a través de un navegador web en <http://localhost>.

orders: Administra el proceso de realizar pedidos y realizar un seguimiento de los mismos. Ofrece un punto de entrada en /list para listar pedidos y /orders para crear nuevos pedidos.

carts: Maneja la funcionalidad de carrito de compras. Los puntos de entrada principales son a través de la API en /cart, lo que permite a los usuarios agregar, eliminar y consultar productos en el carrito.

user-sim: No tiene un punto de entrada directo para los usuarios, pero se utiliza internamente para simular la autenticación de usuario y la generación de tokens.

carts-db: Aloja la base de datos que almacena información relacionada con los carritos de compras de los usuarios. Los servicios de carrito de compras pueden acceder a través de la red para almacenar y recuperar información del carrito.

user: Administra la autenticación de usuarios y la creación de cuentas. Los usuarios pueden registrarse y autenticarse a través de los puntos de entrada /register y /login.

2.2)

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ git clone https://github.com/microservices-demo/front-end.git
Cloning into 'front-end'...
remote: Enumerating objects: 1236, done.
remote: Total 1236 (delta 0), reused 0 (delta 0), pack-reused 1236
Receiving objects: 100% (1236/1236), 47.90 MiB | 1.09 MiB/s, done.
Resolving deltas: 100% (687/687), done.
```

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ git clone https://github.com/microservices-demo/user.git
Cloning into 'user'...
remote: Enumerating objects: 1063, done.
remote: Total 1063 (delta 0), reused 0 (delta 0), pack-reused 1063
Receiving objects: 100% (1063/1063), 172.83 KiB | 120.00 KiB/s, done.
Resolving deltas: 100% (601/601), done.
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ git clone https://github.com/microservices-demo/edge-router.git
Cloning into 'edge-router'...
remote: Enumerating objects: 50, done.
remote: Total 50 (delta 0), reused 0 (delta 0), pack-reused 50
Receiving objects: 100% (50/50), 14.51 KiB | 225.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.
```

2.3)

La razón principal para utilizar repositorios separados para el código y la configuración del sistema es la modularidad y la gestión eficiente de los componentes.

Ventajas:

Separación de Responsabilidades: Cada repositorio puede centrarse en una responsabilidad específica. Por ejemplo, un repositorio puede contener el código fuente de una aplicación, mientras que otro puede almacenar archivos de configuración, scripts de implementación o archivos Docker Compose.

Gestión de Versiones Independientes: Los repositorios separados permiten administrar las versiones de cada componente de manera independiente. Esto facilita la actualización y el mantenimiento de componentes individuales sin afectar a otros.

Mejor Colaboración: Los equipos de desarrollo pueden trabajar de manera más eficiente en componentes separados. Cada equipo puede gestionar su propio repositorio y realizar cambios sin interferir con otros equipos.

Reutilización y Modularidad: Los componentes, como aplicaciones o servicios, pueden reutilizarse en otros proyectos o sistemas. La modularidad facilita la creación de sistemas complejos a partir de componentes existentes.

Desventajas:

Complejidad de Gestión: Tener múltiples repositorios puede aumentar la complejidad de la gestión de código y configuración, ya que es necesario administrarlos por separado.

Posible Fragmentación: Si no se gestiona adecuadamente, la información y la documentación pueden fragmentarse en varios repositorios, lo que dificulta la ubicación de recursos específicos.

Aprendizaje Adicional: Trabajar con múltiples repositorios puede requerir un esfuerzo adicional de aprendizaje para los miembros del equipo.

2.4)

El contenedor "edge-router" es el componente del sistema que cumple el rol de API Gateway, proporcionando un punto de entrada centralizado y simplificando el acceso a los microservicios del sistema.

2.5)

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ curl http://localhost/customers
{"_embedded":{"customer":[{"firstName":"Eve","lastName":"Berger","username":"Eve_Berger","id":"57a98d98e4b00679b4a830af","_links":{"addresses":{"href":"http://user/customers/57a98d98e4b00679b4a830af/addresses"},"cards":{"href":"http://user/customers/57a98d98e4b00679b4a830af/cards"},"customer":{"href":"http://user/customers/57a98d98e4b00679b4a830af"},"self":{"href":"http://user/customers/57a98d98e4b00679b4a830af"}}}],{"firstName":"User","lastName":"Name","username":"user","id":"57a98d98e4b00679b4a830b2","_links":{"addresses":{"href":"http://user/customers/57a98d98e4b00679b4a830b2/addresses"},"cards":{"href":"http://user/customers/57a98d98e4b00679b4a830b2/cards"},"customer":{"href":"http://user/customers/57a98d98e4b00679b4a830b2"},"self":{"href":"http://user/customers/57a98d98e4b00679b4a830b2"}}}],{"firstName":"User1","lastName":"Name1","username":"user1","id":"57a98d98e4b00679b4a830b5","_links":{"addresses":{"href":"http://user/customers/57a98d98e4b00679b4a830b5/addresses"},"cards":{"href":"http://user/customers/57a98d98e4b00679b4a830b5/cards"},"customer":{"href":"http://user/customers/57a98d98e4b00679b4a830b5"},"self":{"href":"http://user/customers/57a98d98e4b00679b4a830b5"}}}]}}
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ curl http://localhost/catalogue
{"id":"03fef6ac-1896-4ce8-bd69-b798f85c6e0b","name":"Holy","description":"Socks fit for a Messiah. You too can experience walking in water with these special edition beauties. Each hole is lovingly proggled to leave smooth edges. The only sock approved by a higher power.","imageUrl":["/catalogue/images/holy_1.jpg","/catalogue/images/holy_2.jpeg"],"price":99.99,"count":1,"tag":["action","magic"]},{id":"3395a43e-2d88-40de-b95f-e00e1502085b","name":"Colourful","description":"proident occaecat irure et excepteur labore minim nisi amet irure","imageUrl":["/catalogue/images/colourful_socks.jpg","/catalogue/images/colourful_socks.jpg"],"price":18,"count":438,"tag":["brown","blue"]},{id":"510a0d7e-8e83-4193-b483-e27e09ddc34d","name":"SuperSport XL","description":"Ready for action. Engineers: be ready to smash that next bug! Be ready, with these super-action-sport-masterpieces. This particular engineer was chased away from the office with a stick.","imageUrl":["/catalogue/images/puma_1.jpeg","/catalogue/images/puma_2.jpeg"],"price":15,"count":820,"tag":["sport","formal","black"]},{id":"808a2de1-1aaa-4c25-a9b9-6612e8f29a38","name":"Crossed","description":"A mature sock, crossed, with an air of nonchalance","imageUrl":["/catalogue/images/cross_1.jpg","/catalogue/images/cross_2.jpeg"],"price":17.32,"count":738,"tag":["blue","action","red","formal"]},{id":"819e1fbf-8b7e-4f6d-811f-693534916a8b","name":"Figueroa","description":"enim officia aliqua excepteur esse deserunt quis aliquip nostrud anim","imageUrl":["/catalogue/images/WAT.jpg","/catalogue/images/WAT2.jpg"],"price":14,"count":808,"tag":["green","formal","blue"]},{id":"837ab141-399e-4c1f-9abc-bace40296bac","name":"Cat socks","description":"consequat amet cupidatat minim laborum tempor elit ex consequat in","imageUrl":["/catalogue/images/catsocks.jpg","/catalogue/images/catsocks2.jpg"],"price":15,"count":175,"tag":["brown","formal","green"]},{id":"a0a4f044-b040-410d-8ead-4de0446aec7e","name":"Nerd leg","description":"For all those leg lovers out there. A perfect example of a swivel chair trained calf. Meticulously trained on a diet of sitting and Pina Coladas. Phwarr...","imageUrl":["/catalogue/images/bit_of_leg_1.jpg","/catalogue/images/bit_of_leg_2.jpeg"],"price":7.99,"count":115,"tag":["blue","skin"]},{id":"d3588630-ad8e-49df-bbd7-3167f7efb246","name":"YouTube.sock","description":"We were not paid to sell this sock. It's just a bit geeky","imageUrl":["/catalogue/images/youtube_1.jpeg","/catalogue/images/youtube_2.jpeg"],"price":10.99,"count":801,"tag":["formal","geek"]},{id":"zzz4f044-b040-410d-8ead-4de0446aec7e","name":"Classic","description":"Keep it simple","imageUrl":["/catalogue/images/classic.jpg","/catalogue/images/classic2.jpg"],"price":12,"count":127,"tag":["brown","green"]}]
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP4/socks-demo$ curl http://localhost/tags
{"tags":["brown","geek","formal","blue","skin","red","action","sport","black","magic","green"],"err":null}
```

2.6)

curl http://localhost/customers: Esta solicitud es manejada por el servicio de "clientes" (customers).

El punto de entrada principal para todas las operaciones en esta arquitectura de microservicios es el contenedor llamado docker-compose_edge-router_1. Este contenedor actúa como un enrutador y una pasarela de API que recibe las solicitudes y las enruta al servicio correspondiente en función de la URL de la solicitud y otros criterios. En otras palabras, es el servicio que procesa las operaciones de enrutamiento y reenvío hacia los servicios adecuados.

2.7)

curl http://localhost/catalogue: Esta solicitud es manejada por el servicio de "catálogo" (catalogue).

curl http://localhost/tags: Esta solicitud es manejada por el servicio de "etiquetas" (tags).

2.8)

Los datos en los servicios de esta aplicación persisten en bases de datos como PostgreSQL y MongoDB. Además, se utilizan colas de mensajes (RabbitMQ) y sistemas de caché en memoria (Redis) para gestionar operaciones asíncronas y mejorar el rendimiento.

2.9)

El componente encargado del procesamiento de la cola de mensajes es el servicio "shipping." Este servicio utiliza RabbitMQ para gestionar las solicitudes de envío y procesarlas de manera asíncrona.

2.10)

Para la comunicación entre los microservicios, se utilizan interfaces RESTful a través de HTTP. Cada microservicio expone una API REST que permite a otros servicios comunicarse con él a través de solicitudes HTTP. Esta arquitectura basada en HTTP permite una comunicación sencilla y flexible entre los servicios y es un estándar común en el desarrollo de microservicios.