

1)

```
ignacioachaval@Ignacios-MacBook-Pro ingsoft3 % mkdir -p socks-demo
ignacioachaval@Ignacios-MacBook-Pro ingsoft3 % cd socks-demo
ignacioachaval@Ignacios-MacBook-Pro 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 | 1.47 MiB/s, done.
Resolving deltas: 100% (6208/6208), done.
```

```
ignacioachaval@Ignacios-MacBook-Pro socks-demo % cd microservices-demo

ignacioachaval@Ignacios-MacBook-Pro microservices-demo % docker-compose -f deploy/docker-compose/docker-compose.yml up -d

WARNING: The MYSQL_ROOT_PASSWORD variable is not set. Defaulting to a blank string.
Creating network "docker-compose_default" with the default driver
Pulling front-end (weaveworksdemos/front-end:0.3.12)...
0.3.12: Pulling from weaveworksdemos/front-end
709515475419: Pull complete
8a7a0a7b6b80: Pull complete
79dadb976eeb: Pull complete
f48430971a7c: Pull complete
69ee1f19577a: Pull complete
89bb0a032ada: Pull complete
01d99281a40a: Pull complete
07a3ae088a4f: Pull complete
a9efb8659a27: Pull complete
Digest: sha256:26a2d9b6b291dee2dca32fca3f5bff6c2fa07bb5954359afcbc8001cc70eac71
Status: Downloaded newer image for weaveworksdemos/front-end:0.3.12
Pulling edge-router (weaveworksdemos/edge-router:0.1.1)...
0.1.1: Pulling from weaveworksdemos/edge-router
b7f33cc0b48e: Pull complete
6c1bb5764098: Pull complete
3b141d7c0818: Pull complete
820d487bb3eb: Pull complete
2261519a3626: Pull complete
Digest: sha256:72855b1ab6ec5b8b7e03429af9fa04a2a87f97a8c139724afc75a47970fa6438
Status: Downloaded newer image for weaveworksdemos/edge-router:0.1.1
Pulling catalogue (weaveworksdemos/catalogue:0.3.5)...
0.3.5: Pulling from weaveworksdemos/catalogue
709515475419: Already exists
a1a52e31c764: Pull complete
0c10f86b9838: Pull complete
449f4beba5c2: Pull complete
eba920f195c8: Pull complete
Digest: sha256:0147a65b7116569439eefb1a6dbed455fe022464ef70e0c3cab75bc4a226b39b
Status: Downloaded newer image for weaveworksdemos/catalogue:0.3.5
Pulling catalogue-db (weaveworksdemos/catalogue-db:0.3.0)...
0.3.0: Pulling from weaveworksdemos/catalogue-db
75a822cd7888: Pull complete
b8d5846e536a: Pull complete
b75e9152a170: Pull complete
832e6b030496: Pull complete
fe4a6c835905: Pull complete
```

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

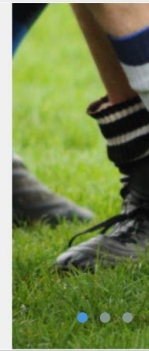
Login | Register



HOME

CATALOGUE ▾

0 items in cart



### 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

Username

ignacioachaval

First name

first name

Please fill out this field

Last name

last name

Email

email

Password

.....

➔ Register

Filters

Clear


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

Apply


Showing 5 of 5 products

Show 3 5


Sort by Price



Holy




Colourful



Crossed

Shopping cart

Product	Quantity	Unit price	Discount	Total
 <div>Holy</div>	1	\$99.99	\$0.00	\$99.99
Total				\$99.99

Continue shopping

Update basket

Proceed to checkout

Order summary

Shipping and additional costs are calculated based on the values you have entered.

Order subtotal	\$99.99
Shipping and handling	\$4.99
Tax	\$0.00
<b>Total</b>	<b>\$104.98</b>

2)

### **weaveworksdemos/load-test**

Descripción: Realiza pruebas de carga en la aplicación para evaluar su rendimiento.

### **weaveworksdemos/catalogue**

Descripción: Gestiona la información del catálogo de productos disponibles en la plataforma.

### **weaveworksdemos/user**

Descripción: Gestiona la funcionalidad relacionada con usuarios, como la autenticación y los perfiles de usuario.

### **weaveworksdemos/payment**

Descripción: Maneja el procesamiento de pagos de las transacciones en la plataforma.

### **weaveworksdemos/front-end**

Descripción: Representa la interfaz de usuario con la que los clientes interactúan al explorar y comprar productos.

### **weaveworksdemos/shipping**

Descripción: Gestiona el envío y la logística de los productos, incluyendo el cálculo de costos de envío.

### **weaveworksdemos/carts**

Descripción: Administra los carros de compra de los usuarios, permitiéndoles agregar, quitar y actualizar elementos.

### **weaveworksdemos/orders**

Descripción: Se encarga de la creación y gestión de las órdenes de compra de los clientes.

### **weaveworksdemos/queue-master**

Descripción: Actúa como componente central para administrar colas de mensajes y tareas en segundo plano.

### **weaveworksdemos/edge-router**

Descripción: Dirige y equilibra el tráfico entrante a diferentes partes de la aplicación.

### **weaveworksdemos/catalogue-db**

Descripción: Es una base de datos que almacena datos relacionados con el catálogo de medias.

### **weaveworksdemos/user-db**

Descripción: Es una base de datos que almacena datos relacionados con los usuarios, como perfiles y datos de autenticación.

```
ignacioachaval@Ignacios-MacBook-Pro socks-demo % cd socks-demo
git clone https://github.com/microservices-demo/front-end.git
git clone https://github.com/microservices-demo/user.git
git clone https://github.com/microservices-demo/edge-router.git
.
.
cd: no such file or directory: socks-demo
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 | 3.51 MiB/s, done.
Resolving deltas: 100% (687/687), done.
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 | 570.00 KiB/s, done.
Resolving deltas: 100% (601/601), done.
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 | 136.00 KiB/s, done.
Resolving deltas: 100% (12/12), done.
.: not enough arguments
.: not enough arguments
ignacioachaval@Ignacios-MacBook-Pro socks-demo %
```

## **Uso de Repositorios Separados:**

Los repositorios separados para código y configuración del sistema son una práctica común en la arquitectura de microservicios. Aquí están algunos puntos a favor y en contra:

### **Puntos a favor:**

1. Separación de Responsabilidades: Al separar el código y la configuración, se siguen los principios de separación de responsabilidades. Los desarrolladores pueden centrarse en escribir código de calidad y las operaciones pueden administrar y modificar la configuración sin afectar directamente el código.
2. Flexibilidad: Permite cambios en la configuración sin necesidad de modificar el código fuente. Esto facilita la adaptación del sistema a diferentes entornos (desarrollo, pruebas, producción) sin tocar el código.
3. Gestión más Sencilla: La gestión de versiones y despliegues se simplifica. Los cambios en la configuración pueden realizarse de forma independiente de los cambios en el código, lo que facilita el mantenimiento y las actualizaciones.

### **Puntos en contra:**

1. Complejidad Adicional: La separación de código y configuración puede agregar una capa adicional de complejidad al sistema. Se deben administrar múltiples repositorios y procesos de despliegue.
2. Posibilidad de Desincronización: Existe la posibilidad de que la configuración en el repositorio esté desactualizada con respecto al código, lo que podría llevar a problemas de compatibilidad.
3. Mayor Sobrecarga de Gestión: La administración de múltiples repositorios y la sincronización entre ellos puede requerir más trabajo y recursos.

## **Contenedor que hace las veces de API Gateway:**

El contenedor que actúa como API Gateway puede ser ``weaveworksdemos/edge-router``. El API Gateway es un componente que maneja solicitudes de clientes, enrutando el tráfico a los microservicios apropiados y realizando funciones como autenticación, autorización y equilibrio de carga.

## **Procesamiento de Operación curl http://localhost/customers:**

Para responder a la operación ``curl http://localhost/customers``, el servicio ``weaveworksdemos/user`` procesa la solicitud relacionada con los datos de clientes.

### **Procesamiento de Operación curl http://localhost/catalogue:**

La operación `curl http://localhost/catalogue` es procesada por el servicio `weaveworksdemos/catalogue`, que se encarga de la gestión del catálogo de productos.

### **Procesamiento de Operación curl http://localhost/tags:**

La operación `curl http://localhost/tags` es procesada por el servicio `weaveworksdemos/catalogue`, ya que generalmente las etiquetas o "tags" están relacionadas con la información del catálogo de productos.

### **Persistencia de Datos:**

Los datos en los servicios pueden persistirse de diversas formas, pero la mayoría de estos microservicios probablemente utilicen bases de datos para almacenar datos de manera persistente. Cada servicio puede tener su propia base de datos, como `weaveworksdemos/catalogue-db` y `weaveworksdemos/user-db`, para almacenar datos relacionados con catálogo y usuarios.

### **Procesamiento de la Cola de Mensajes:**

El componente encargado del procesamiento de la cola de mensajes es `weaveworksdemos/queue-master`. Se encarga de gestionar las operaciones relacionadas con la cola de mensajes, como la distribución de tareas a los servicios apropiados.

### **Tipo de Interfaz para Comunicación:**

Los microservicios suelen utilizar una interfaz de comunicación basada en HTTP o RESTful API para interactuar entre sí. Se comunican mediante solicitudes HTTP para realizar operaciones y compartir datos, lo que facilita la interoperabilidad y la integración de servicios.