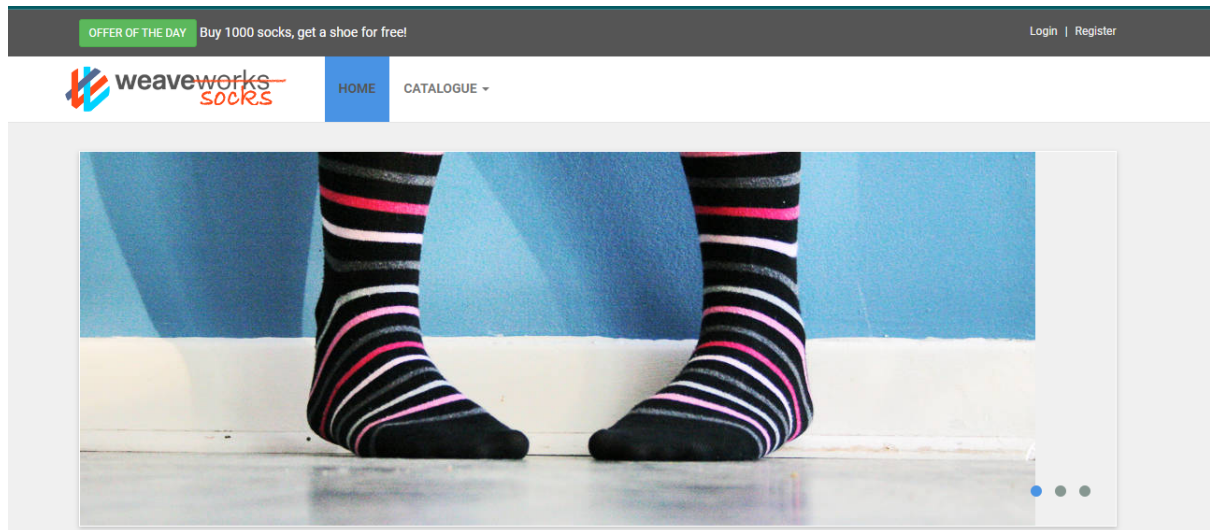


TP4

Acceder al Localhost



Generar un usuario

Register



Username

Santi

First name

Santiago

Last name

Riveros

Email

santi@gmail.com

Password


.....

Register

Realizar un Orden

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

Logged in as Santiago Riveros | Logout



HOME CATALOGUE ACCOUNT

0 items in cart

Home > My orders

Customer section

My orders

My orders

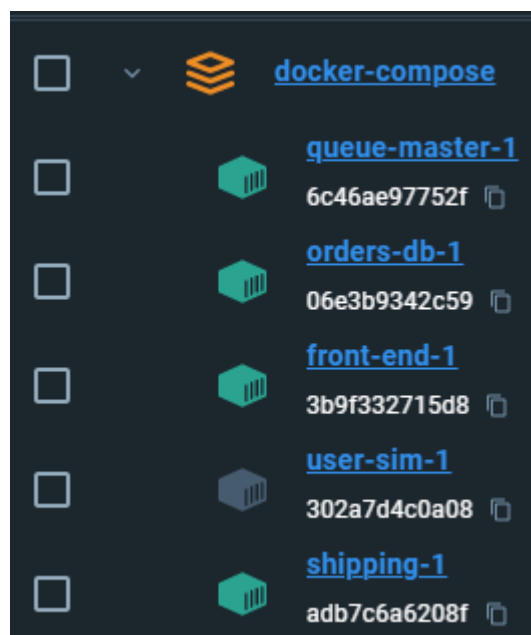
Your orders in one place.

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
# 64ee469fc44e78000795791e	2023-08-29 19:27:27	\$ 22.31	Shipped	View

Los contenedores que se crearon fueron:

1. front-end
2. edge-router
3. catalogue
4. catalogue-db
5. carts
6. carts-db
7. orders
8. orders-db
9. shipping
10. queue-master
11. rabbitmq
12. payment
13. user
14. user-db
15. user-sim



```

PS D:\Santiago\Ingenieria en Sistemas\4to Año\2do Semestre\Ingenieria de Software III\Practico\ing-software-3\TP4\microservices-demo> curl http://localhost/customers

StatusCode      : 200
StatusDescription : OK
Content         : {"_embedded":{"customer":[{"firstName":"Eve","lastName":"Berger","username":"Eve_Berger","id":"57a98d98e4b00679b4a830af","links":{"addresses":{"href":"http://user/customers/57a98d98e4b00679b4a830af/a...
RawContent      : HTTP/1.1 200 OK
                  Date: Tue, 29 Aug 2023 19:33:22 GMT
                  Set-Cookie: _TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3Aphvuj_G_aLQe00ILvSj7BaAQrrQuAbzD.CnAo%2BpuAf1zjt0%2FIsyvQQvzuBGLdSw5xBozvVX1%2FcsW; ...
Forms           : {}
Headers         : {[Date, Tue, 29 Aug 2023 19:33:22 GMT], [Set-Cookie, _TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3Aphvuj_G_aLQe00ILvSj7BaAQrrQuAbzD.CnAo%2BpuAf1zjt0%2FIsyvQQvzuBGLdSw5xBozvVX1%2FcsW; Path=/; HttpOnly], [X-Powered-By, Express], [Content-Length, 1591]...}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 1591

```

Endpoints:

```

module.exports = {
  catalogueUrl: util.format("http://catalogue%s", domain),
  tagsUrl:      util.format("http://catalogue%s/tags", domain),
  cartsUrl:     util.format("http://carts%s/carts", domain),
  ordersUrl:    util.format("http://orders%s", domain),
  customersUrl: util.format("http://user%s/customers", domain),
  addressUrl:   util.format("http://user%s/addresses", domain),
  cardsUrl:     util.format("http://user%s/cards", domain),
  loginUrl:     util.format("http://user%s/login", domain),
  registerUrl:  util.format("http://user%s/register", domain),
};

```

¿Por qué cree usted que se está utilizando repositorios separados para el código y/o la configuración del sistema? Explique puntos a favor y en contra.

Ventajas:

1. **Separación de Responsabilidades:** La división en repositorios independientes permite una distinción clara entre el código de la aplicación y la configuración del sistema. Esto facilita la administración y el mantenimiento de estos dos aspectos de la aplicación de manera autónoma.
2. **Gestión de Versiones Simplificada:** Los cambios en el código y en la configuración del sistema a menudo tienen plazos y ciclos de vida diferentes. La utilización de repositorios separados permite administrar las versiones de manera precisa y adecuada para cada componente.
3. **Facilitación de Colaboración:** Equipos distintos pueden trabajar de manera simultánea en el código y la configuración sin interferir entre sí. Esta dinámica fortalece la

colaboración y permite a los equipos concentrarse en sus áreas especializadas.

4. **Automatización y Despliegue Continuo Optimizados:** Al apartar la configuración del sistema, es posible automatizar la implementación y la configuración. Esto posibilita la adopción de prácticas como el despliegue continuo y la orquestación de infraestructura.
5. **Flexibilidad Elevada en Despliegue:** La separación entre el código y la configuración simplifica la adaptación de la infraestructura o recursos subyacentes sin necesidad de modificar el código. Esta flexibilidad facilita ajustarse a los cambios en las demandas o requisitos.

Desventajas:

1. **Incremento de Complejidad:** Manejar múltiples repositorios puede introducir complejidad adicional, especialmente en equipos pequeños o proyectos menos complejos.
2. **Sincronización Requerida:** Es esencial mantener la concordancia entre los cambios en el código y la configuración para garantizar su interoperabilidad. Esto puede demandar un esfuerzo adicional.
3. **Desafíos en la Ubicación:** Para los recién llegados al equipo, podría ser difícil identificar dónde se encuentra la configuración pertinente, lo que podría dar lugar a retrasos y confusiones.
4. **Posible Desajuste:** Si no se coordinan adecuadamente los cambios en el código y la configuración, podría surgir una desincronización que afecte su compatibilidad.
5. **Mayor Complejidad en Implementación Local:** Durante el desarrollo local o las pruebas, administrar y mantener ambos repositorios podría implicar un esfuerzo suplementario y un proceso más intrincado.

¿Cuál contenedor hace las veces de API Gateway?

El contenedor que hace de API Gateway es edge-router. El repositorio que corresponde a este contenedor tiene un archivo traefik.toml. Traefik es un *Edge Router*; esto significa que es la **puerta a la plataforma**. Se encarga de *interceptar* cada petición que se realiza y enrutarla al servicio correcto. Traefik sabe la lógica y las reglas que determinan qué servicio es el encargado de gestionar cada petición.

¿Cuál de todos los servicios está procesando la operación?

```
PS D:\Santiago\Ingenieria en Sistemas\4to Año\2do Semestre\Ingenieria de Software III\Practico\ing-software-3\TP4\microservices-demo> curl http://localhost/catalogue

StatusCode      : 200
StatusDescription : OK
Content         : [{"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 p...
RawContent      : HTTP/1.1 200 OK
                 Date: Tue, 29 Aug 2023 19:39:28 GMT
                 Set-Cookie: _TRAFFIK_BACKEND=http://front-end:8079,md.sid=s%3Aajpb7uCiDeszya_bN9E0Tmqc0Sw-uDwmJ.%2Bvc62mSfcdUxWdf7wyNaI%2FxSRCeZfXwRKN%2BgDifz%2Bz8...
Forms           : {}
Headers         : [{"Date, Tue, 29 Aug 2023 19:39:28 GMT"}, [Set-Cookie, _TRAFFIK_BACKEND=http://front-end:8079,md.sid=s%3Aajpb7uCiDeszya_bN9E0Tmqc0Sw-uDwmJ.%2Bvc62mSfcdUxWdf7wyNaI%2FxSRCeZfXwRKN%2BgDifz%2Bz8; Path=/; HttpOnly], [X-Powered-By, Express], [Content-Type, text/plain; charset=utf-8]...]
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 2795
```

```
P5 D:\Santiago\Ingenieria en Sistemas\4to Año\2do Semestre\Ingenieria de Software III\Practico\ing-software-3\TP4\microservices-demo> curl http://localhost/tags

StatusCode      : 200
StatusDescription : OK
Content         : {"tags":["brown","geek","formal","blue","skin","red","action","sport","black","magic","green"],"err":null}

RawContent      : HTTP/1.1 200 OK
                  Date: Tue, 29 Aug 2023 19:39:40 GMT
                  Set-Cookie: _TRAFFIK_BACKEND=http://front-end:8079,md.sid=s%3APDKP3sYCWuRnOYMRf-Le1B6bZ6bF3rEL.yw5KSv87ZEkbSVfPyXZgy%2B9c93ozMFsln3bk58YTOL0; Path...
Forms           : {}
Headers         : {[Date, Tue, 29 Aug 2023 19:39:40 GMT], [Set-Cookie, _TRAFFIK_BACKEND=http://front-end:8079,md.sid=s%3APDKP3sYCWuRnOYMRf-Le1B6bZ6bF3rEL.yw5KSv87ZEkbSVfPyXZgy%2B9c93ozMFsln3bk58YTOL0; Path=/; HttpOnly], [X-Powered-By, Express], [Content-Length, 107]}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 107
```

Volúmenes de Docker: Los volúmenes son una forma de persistir datos en Docker. Los volúmenes son directorios o archivos que están fuera del sistema de archivos del contenedor y se montan en el contenedor. Esto permite que los datos persistan incluso cuando el contenedor se detiene o se elimina. Los volúmenes pueden ser administrados por Docker y pueden compartirse entre varios contenedores.

Los servicios que deben almacenar datos, en este caso, user-db y catalogue-db, tienen creados volúmenes de Docker para la persistencia de datos.

¿Cuál es el componente encargado del procesamiento de la cola de mensajes?

El componente encargado del procesamiento de la cola de mensajes es el servicio llamado **queue-master**

¿Qué tipo de interfaz utilizan estos microservicios para comunicarse?

Estos microservicios utilizan principalmente comunicación basada en **HTTP/HTTPS** para interactuar entre sí. Esto se logra a través de interfaces API expuestas por los servicios que permiten a otros microservicios realizar solicitudes HTTP a rutas específicas para obtener o enviar datos.