

1)

```
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3> 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 | 3.05 MiB/s, done.
Resolving deltas: 100% (6208/6208), done.
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3> mkdir -p socks-demo

Directorio: C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3

Mode                LastWriteTime         Length Name
----                -
d-----          16/10/2023    13:30             socks-demo

PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3> cd socks-demo
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\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 | 2.73 MiB/s, done.
Resolving deltas: 100% (6207/6207), done.


PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-demo> cd microservices-demo
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-demo\microservices-demo> docker-compose -f deploy/docker-compose/docker-compose.yml up -d
time="2023-10-16T13:33:02-03:00" level=warning msg="The \"MYSQL_ROOT_PASSWORD\" variable is not set. Defaulting to a blank string."
[*] Running 119/24
  ✓ orders 3 layers [#####] 08/08 Pulled 105.3s
  ✓ catalogue-db 12 layers [#####] 08/08 Pulled 178.7s
  ✓ shipping 9 layers [#####] 08/08 Pulled 74.1s
  ✓ user-sim 11 layers [#####] 08/08 Pulled 82.0s
  ✓ user 8 layers [#####] 08/08 Pulled 8.0s
  ✓ carts-db Pulled 77.6s
  ✓ queue-master 2 layers [##] 08/08 Pulled 104.1s
  ✓ user-db 11 layers [#####] 08/08 Pulled 178.6s
  ✓ orders-db 14 layers [#####] 08/08 Pulled 77.6s
  ✓ carts 3 layers [#####] 08/08 Pulled 168.8s
  ✓ catalogue 4 layers [#####] 08/08 Pulled 29.0s
  ✓ front-end 8 layers [#####] 08/08 Pulled 180.5s
  ✓ payment 3 layers [#####] 08/08 Pulled 105.6s
  ✓ rabbitmq 14 layers [#####] 08/08 Pulled 194.4s
  ✓ edge-router 4 layers [#####] 08/08 Pulled 164.6s

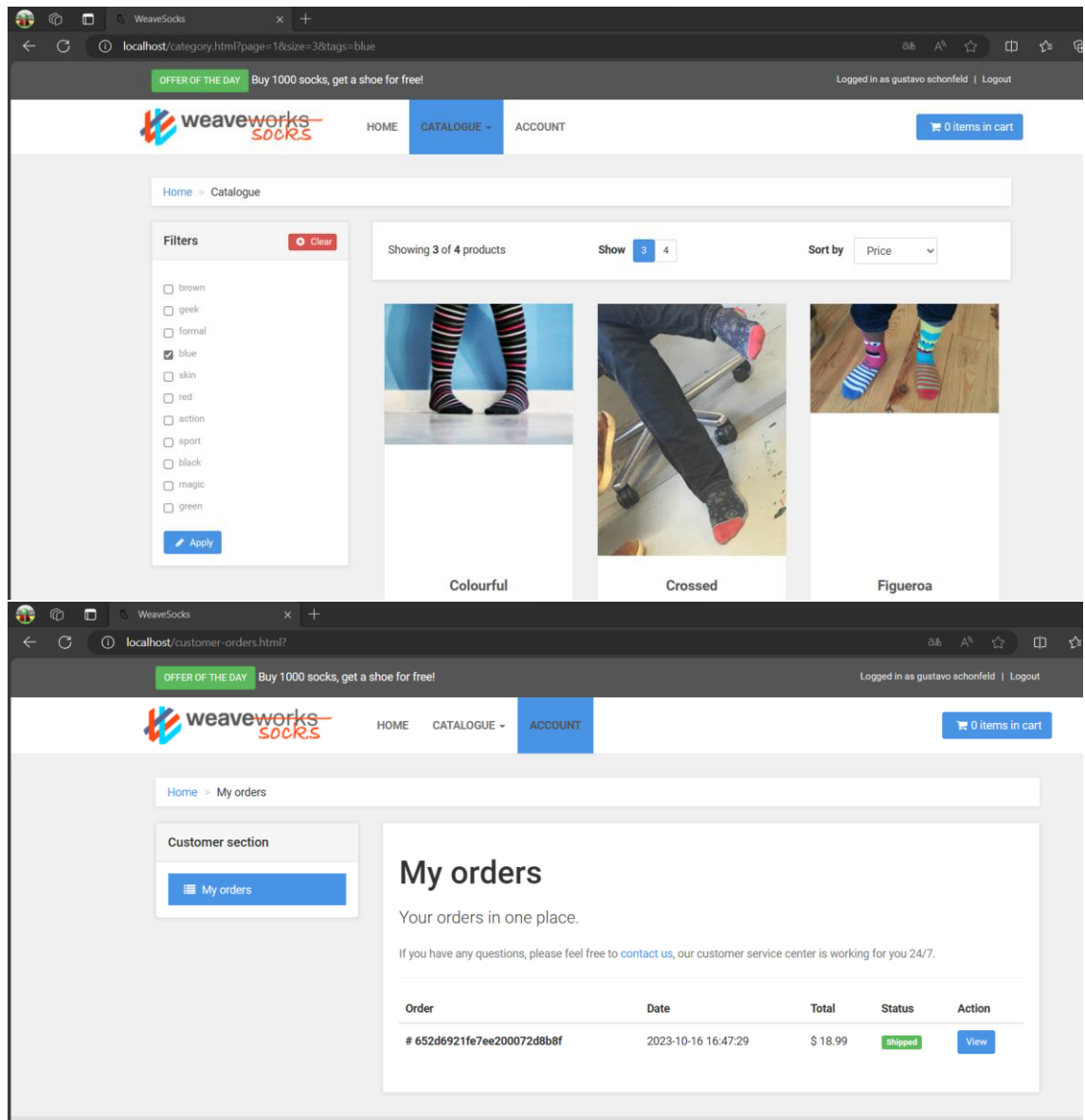
[*] Running 16/16
  ✓ Network docker-compose_default Created 0.0s
  ✓ Container docker-compose-queue-master-1 Started 8.1s
  ✓ Container docker-compose-user-db-1 Started 13.0s
  ✓ Container docker-compose-orders-1 Started 12.1s
  ✓ Container docker-compose-catalogue-1 Started 11.7s
  ✓ Container docker-compose-user-1 Started 8.9s
  ✓ Container docker-compose-payment-1 Started 8.9s
  ✓ Container docker-compose-shipping-1 Started 10.8s
  ✓ Container docker-compose-catalogue-db-1 Started 10.2s
  ✓ Container docker-compose-rabbitmq-1 Started 13.0s
  ✓ Container docker-compose-carts-db-1 Started 12.6s
  ✓ Container docker-compose-carts-1 Started 11.2s
  ✓ Container docker-compose-user-sim-1 Started 12.5s
  ✓ Container docker-compose-orders-db-1 Started 12.3s
  ✓ Container docker-compose-edge-router-1 Started 13.2s
  ✓ Container docker-compose-front-end-1 Started 12.7s
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-demo\microservices-demo>
```

WeaveSocks x +

← localhost

OFFER OF THE DAY Buy 1000 socks, get a shoe for free! Logged in as gustavo.schonfeld | Logout

 [HOME](#) [CATALOGUE](#) [ACCOUNT](#) [0 items in cart](#)



2)

Frontend: Este contenedor representa la interfaz de usuario. Es el punto de entrada para los usuarios y proporciona la funcionalidad de búsqueda, visualización de productos y más.

User: Este contenedor maneja la autenticación y la creación de usuarios en el sistema. Es esencial para la creación de cuentas de usuario.

Edge-router: Este contenedor sirve como enrutador para redirigir solicitudes a los servicios apropiados dentro del sistema. Es un componente importante para gestionar la comunicación entre los diferentes servicios de microservicios.

Cartservice: Este servicio se encarga de gestionar el carrito de compras de los usuarios.

Productcatalogservice: Este servicio proporciona información sobre el catálogo de productos disponibles en el sistema.

Shippingservice: Este servicio maneja la información de envío y la logística de entrega.

Currencyservice: Proporciona información sobre las tasas de cambio y divisas utilizadas en el sistema.

```

PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-demo\microservices-demo> cd ..
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\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 1236Receiving objects: 95% (1175/1236), 39.03 MiB | 19.02MiB/s
39.03 MiB | 19.02 MiB/s
Receiving objects: 100% (1236/1236), 47.90 MiB | 19.59 MiB/s, done.
Resolving deltas: 100% (687/687), done.
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-demo> git clone https://github.com/microservices-demo/user.git
Cloning into 'user'...
remote: Enumerating objects: 1063, done.
remote: Total 1063 (delta 0), pack-reused 1063Receiving objects: 92% (978/1063)
Receiving objects: 100% (1063/1063), 172.83 KiB | 2.01 MiB/s, done.
Resolving deltas: 100% (601/601), done.
PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\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 | 1.81 MiB/s, done.
Resolving deltas: 100% (12/12), done.

```

#### Puntos a favor:

- Separación de Responsabilidades: Los repositorios separados permiten que cada componente (servicio) tenga su propio repositorio. Esto facilita la gestión de código y configuración para un servicio específico sin afectar a otros servicios.
- Escalabilidad: Cada servicio puede evolucionar y escalar independientemente. Los equipos pueden trabajar de manera más autónoma en sus respectivos servicios sin depender de otros equipos.
- Desarrollo Paralelo: Diferentes equipos pueden desarrollar y desplegar servicios de manera paralela. Esto acelera el desarrollo y la entrega de características.
- Facilita la Adopción de Tecnologías Diversas: Diferentes servicios pueden utilizar diferentes tecnologías y lenguajes de programación según sus necesidades específicas.

#### Puntos en contra

- Mayor Complejidad: La administración de múltiples repositorios puede agregar complejidad al proceso de desarrollo y despliegue.
- Mayor Sobrecarga Inicial: Configurar y mantener múltiples repositorios puede requerir más tiempo y recursos al comienzo del proyecto.
- Coordinación Requerida: A veces la coordinación entre equipos o servicios puede ser un desafío, especialmente en proyectos grandes.
- Posible Incoherencia de Versiones: La gestión de dependencias entre servicios puede resultar en incompatibilidades y problemas de versiones.

El contenedor que hace las veces de API Gateway es el edge-router. El API Gateway es una capa que actúa como punto de entrada para las solicitudes de clientes y enruta esas solicitudes a los servicios apropiados dentro del sistema.

```

PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-demo> curl http://localhost/customers
{
  "statusCode": 200,
  "statusDescription": "OK",
  "content": {
    "embedded": {
      "customer": {
        "firstName": "Eve",
        "lastName": "Berger",
        "username": "Eve_Berger",
        "id": "57a98d98e4b00679b4a830af",
        "_links": {
          "addreses": {
            "href": "http://user/customers/57a98d98e4b00679b4a830af/a..."
          }
        }
      }
    }
  },
  "rawContent": "HTTP/1.1 200 OK\r\nDate: Mon, 16 Oct 2023 19:14:47 GMT\r\nSet-Cookie: _TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3Alk55PCNP-HwZ1CHBYX_KJ0uu5o7-CLa.31tgtM9LQrH3Tsv%2BJXxcFAaL4TQ0IgRSTwRNFSVM3dg; Path=...\r\n\r\n",
  "forms": {},
  "headers": {
    "date": "Mon, 16 Oct 2023 19:14:47 GMT",
    "set-cookie": "_TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3Alk55PCNP-HwZ1CHBYX_KJ0uu5o7-CLa.31tgtM9LQrH3Tsv%2BJXxcFAaL4TQ0IgRSTwRNFSVM3dg; Path=/; HttpOnly; [X-Powered-By, Express], [Content-Length, 1597]..."
  },
  "images": {},
  "inputFields": {},
  "links": {},
  "parsedHtml": "mshtml.HTMLDocumentClass",
  "rawContentLength": 1597
}

```

curl <http://localhost/customers>: El servicio que procesa la operación es el user. Este servicio maneja la gestión de clientes.

```

PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-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: Mon, 16 Oct 2023 19:21:36 GMT
                  Set-Cookie:
                    _TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3AFHjFNMJO7zTZRO-Ze254jC3xSgbQ-XVp.V9oJIoGx%2Boredoj4pplvUCFZYiukZroI%2FRtPtBC9kAs;
                    Pa...
Forms           : {}
Headers         : [{"Date, Mon, 16 Oct 2023 19:21:36 GMT"}, [{"Set-Cookie,
                    _TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3AFHjFNMJO7zTZRO-Ze254jC3xSgbQ-XVp.V9oJIoGx%2Boredoj4pplvUCFZYiukZroI%2FRtPtBC9kAs;
                    Path=/; HttpOnly}], [X-Powered-By, Express], [Content-Type, text/plain; charset=utf-8]...}]
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 2795

```

curl <http://localhost/catalogue>: La operación es procesada por el servicio catalogue, que se encarga de proporcionar información sobre el catálogo de productos.

```

PS C:\Users\gusta\OneDrive\Escritorio\UCC\Cuarto Año\IngenieriaSoftware 3\socks-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: Mon, 16 Oct 2023 19:22:02 GMT
                  Set-Cookie:
                    _TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3ApEEAR9rA5aiunKn9mBpzt6th_6CQ2oe0.WTvwaf3WG8rnp3JWE9FV8cGckAzKTANWyJ9tQnFFR7E;
                    Path=/...
Forms           : {}
Headers         : [{"Date, Mon, 16 Oct 2023 19:22:02 GMT"}, [{"Set-Cookie,
                    _TRAEFIK_BACKEND=http://front-end:8079,md.sid=s%3ApEEAR9rA5aiunKn9mBpzt6th_6CQ2oe0.WTvwaf3WG8rnp3JWE9FV8cGckAzKTANWyJ9tQnFFR7E;
                    Path=/; HttpOnly}], [X-Powered-By, Express], [Content-Length, 107]...}]
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 107

```

curl <http://localhost/tags>: La operación es procesada por el servicio catalogue, ya que también maneja las etiquetas (tags) asociadas a los productos en el catálogo.

Los servicios como el catálogo, carritos de compra y autenticación utilizan bases de datos relacionales para almacenar datos. Se utiliza PostgreSQL para gestionar la persistencia de datos. Esto permite el almacenamiento a largo plazo de información como productos, carritos de compra y detalles de autenticación de usuarios.

El componente encargado es Redis, utilizado para gestionar la cola de mensajes y almacenar temporalmente datos como los elementos en el carrito de compra. También se utiliza para almacenar información de autenticación de usuarios y otros datos que requieren una alta velocidad de lectura y escritura.

Utilizan una interfaz basada en HTTP/HTTPS. Cada microservicio expone API HTTP que otros microservicios pueden invocar para acceder a sus funcionalidades. Esto permite una comunicación sencilla y estandarizada entre los servicios. Los datos se intercambian en formato JSON a través de solicitudes HTTP. También se utilizan solicitudes HTTP para comunicarse con la API Gateway (edge-router).