



PREPÁRATE
PARA SER EL
MEJOR



+ **ENTREMIENTO
EXPERIENCIA**



BIENVENIDOS.



Microservicios con .NET Core: Arquitectura, Contenerización y Orquestación

Sesión 06

Ing. Erick Arostegui Cunza
Instructor

earostegui@galaxy.edu.pe



AGENDA

ARQ. DE MICROSERVICIOS (ASPECTOS TRASNVER. Y OBSERVABILIDAD)

- ▶ Implementando una interfaz de usuario que consuma directamente los microservicios (Angular 8).
- ▶ Centralización de accesos a los microservicios utilizando el patrón API Gateway.
- ▶ Implementación de un API Gateway con Azure Api Management.
- ▶ Actualización de la interfaz de usuario para el consumo del API Gateway.



¿Qué es un API Gateway?



Puerta de enlace o puerta de acceso: es un punto de entrada por el que debe pasar si desea acceder a un lugar protegido.

Imaginemos que desea visitar Machu Picchu, un famoso lugar en Cuzco. En primer lugar, tendrá que comprar un boleto y luego pasar por la puerta de peaje (puerta de enlace). En la puerta de peaje, el guardia de seguridad verificará su boleto, si es válido, puede acceder, si no, debe regresar o tener que comprar un boleto válido.



Puerta de enlace o puerta de acceso: es un punto de entrada por el que debe pasar si desea acceder a un lugar protegido.

Imaginemos que desea visitar Machu Picchu, un famoso lugar en Cuzco. En primer lugar, tendrá que comprar un boleto y luego pasar por la puerta de peaje (puerta de enlace). En la puerta de peaje, el guardia de seguridad verificará su boleto, si es válido, puede acceder, si no, debe regresar o tener que comprar un boleto válido.

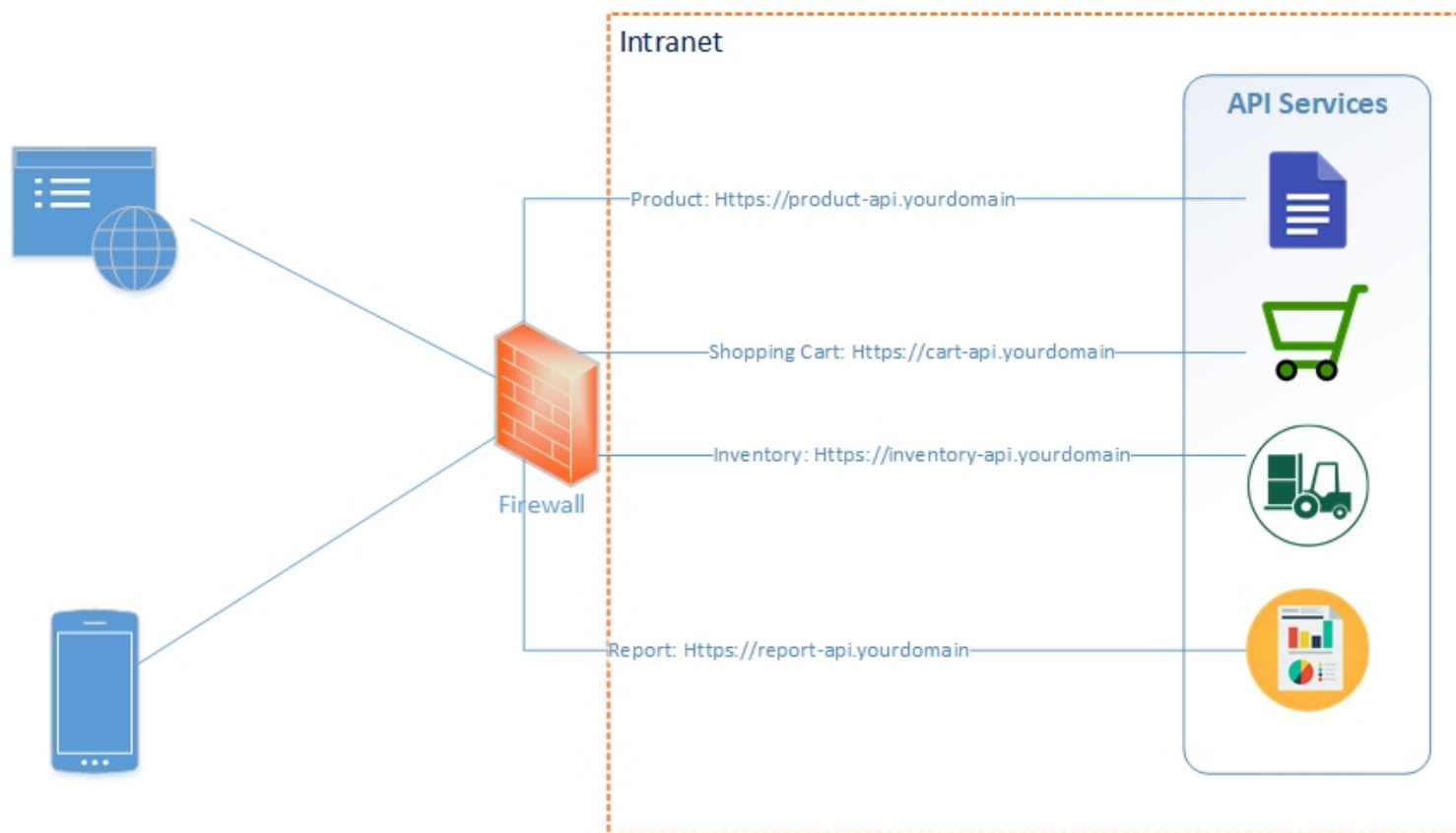


Del mismo modo, un API Gateway es un componente que se encuentra frente a sus API y dentro de una intranet o firewall. La implementación de API Gateway ayudará a garantizar que cada solicitud desde el exterior (internet) tenga que pasar por ella antes de llegar a sus API.

Nota: Podemos tener varias API Gateways en una sola aplicación. Por ejemplo, API Gateway para aplicaciones móviles y API Gateway para aplicaciones de escritorio.

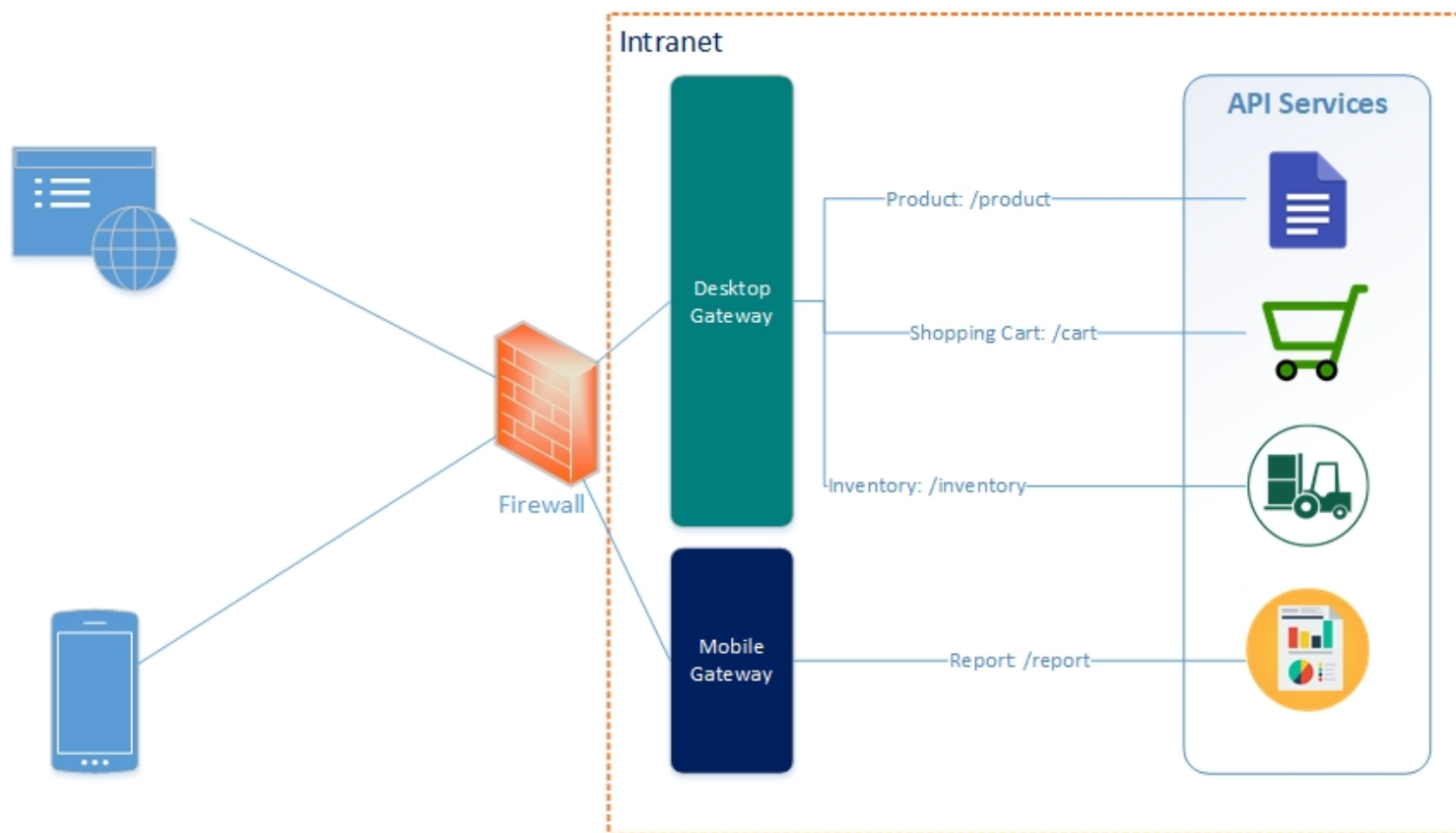


Centralización de accesos a los microservicios utilizando el patrón API Gateway.





Centralización de accesos a los microservicios utilizando el patrón API Gateway.





Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Ventajas de un API Gateway



El mayor beneficio de un API Gateway es que los clientes solo tienen que hablar con la puerta de enlace en lugar de llamar a una API específica. También simplifica los códigos del cliente.



Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Intermedio

Desventajas de un API Gateway



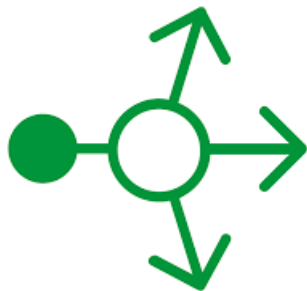
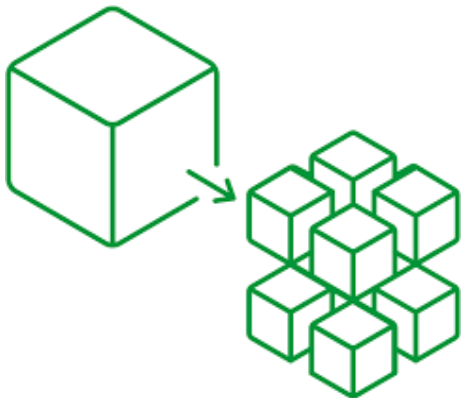
API Gateway también se convierte en un cuello de botella de desarrollo. Debido a que los desarrolladores tienen que actualizar la API Gateway si desean exponer sus API a externos. Es por eso que el proceso para actualizar API Gateway debe ser lo más liviano posible. De lo contrario, el desarrollador tendrá que esperar en línea para actualizar API Gateway.



Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Características de un API Gateway



Enrutamiento : Esta es una de las funciones clave de un API Gateway. Básicamente, cuando un API Gateway recibe una solicitud del exterior, reenviará la solicitud a la API interna por correspondencia en función del mapa de enrutamiento.

Esta función es idéntica a la función de proxy inverso proporcionada por servidores web como NGINX.

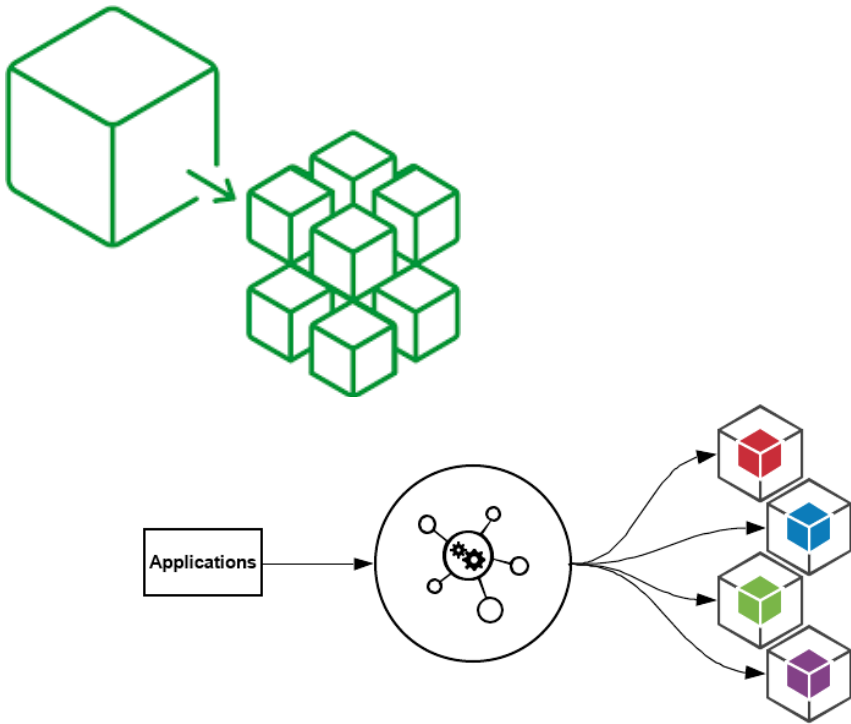


Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Intermedio

Características de un API Gateway



Composición de API : Un API Gateway no solo actúa como proxy inverso, sino que también implementa algunas operaciones de API mediante la composición de API.

Por ejemplo, su aplicación proporciona una función de "Informes", esta función requiere la participación de 3 API's (GetProducts, GetBill y GetInventory) que pertenecen a 3 servicios diferentes: Product, Payment e Inventory. Sin un API Gateway, su cliente tendrá que realizar 3 llamadas a las 3 API's. En su lugar, solo tenemos que implementar una API Gateway utilizando la función de composición de API para exponer solo un endpoint (Informes) que realice las 3 llamadas. Esto ayudará al cliente a recuperar los datos de manera eficiente mediante una sola solicitud al API Gateway.

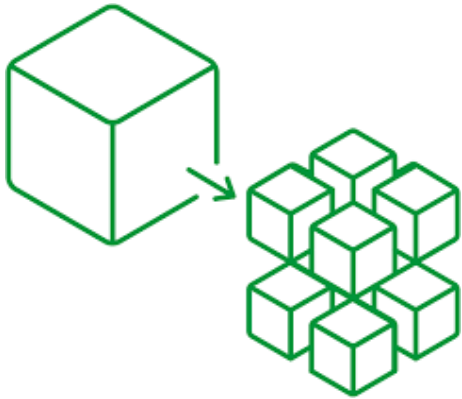


Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Intermedio

Características de un API Gateway



CACHING

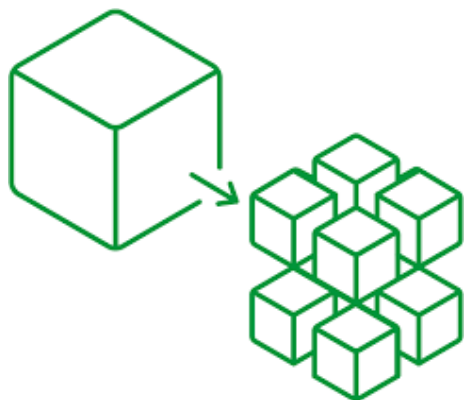
Almacenamiento en caché : Respuestas de caché para reducir las solicitudes a los servicios.



Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Características de un API Gateway



Logging : la función de logging es necesaria para el seguimiento y la depuración.

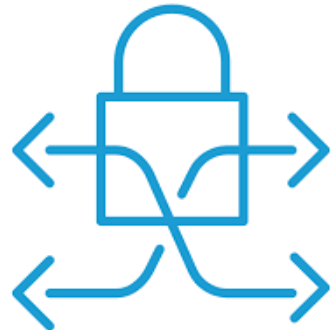
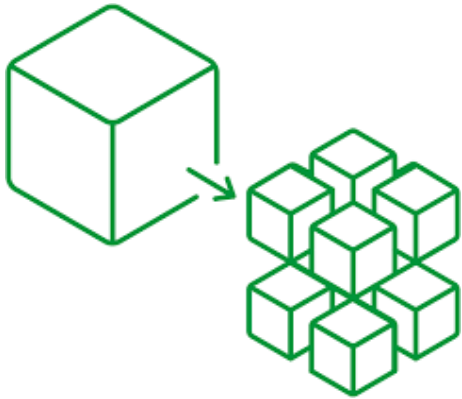


Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Intermedio

Características de un API Gateway



Integración con proveedores de identidad: En lugar de integrar con Identity Provider para la autenticación y autorización en cada servicio, podemos centralizar esto dentro de un API Gateway.

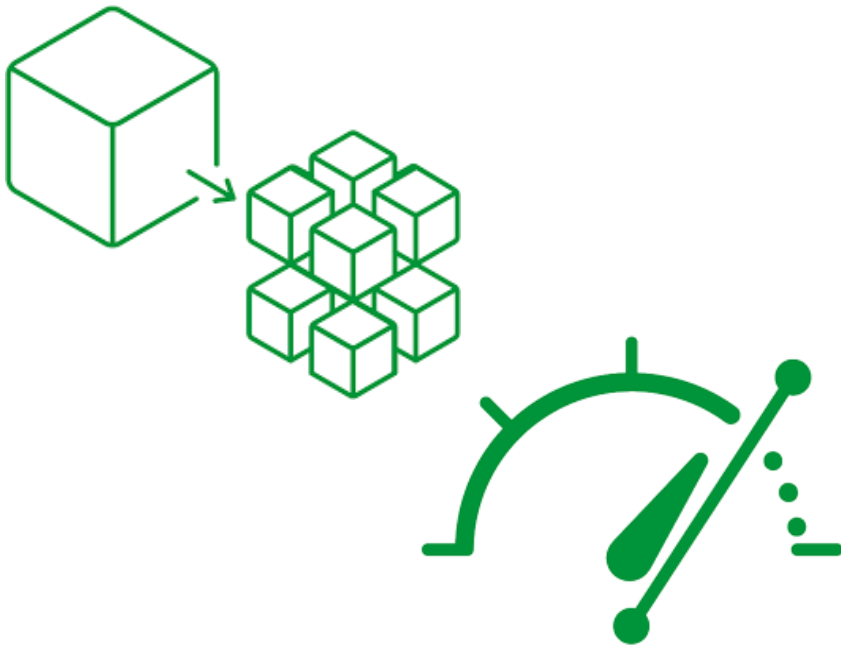


Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Intermedio

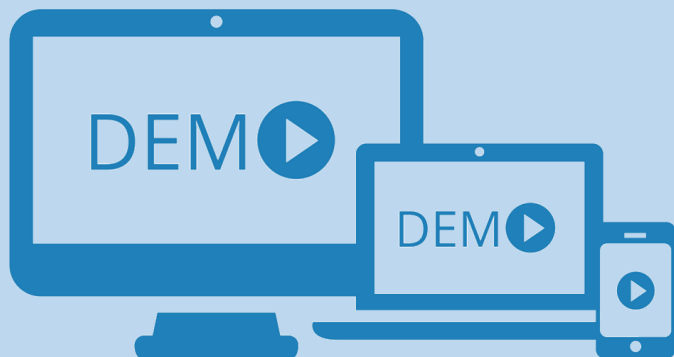
Características de un API Gateway



Limitación de velocidad : limitar cuántas solicitudes por segundo se permiten en un cliente específico o en todos los clientes.



Centralización de accesos a los microservicios utilizando el patrón API Gateway.



Construyendo .NET Core API Gateway con Ocelot



GALAXY
TRAINING