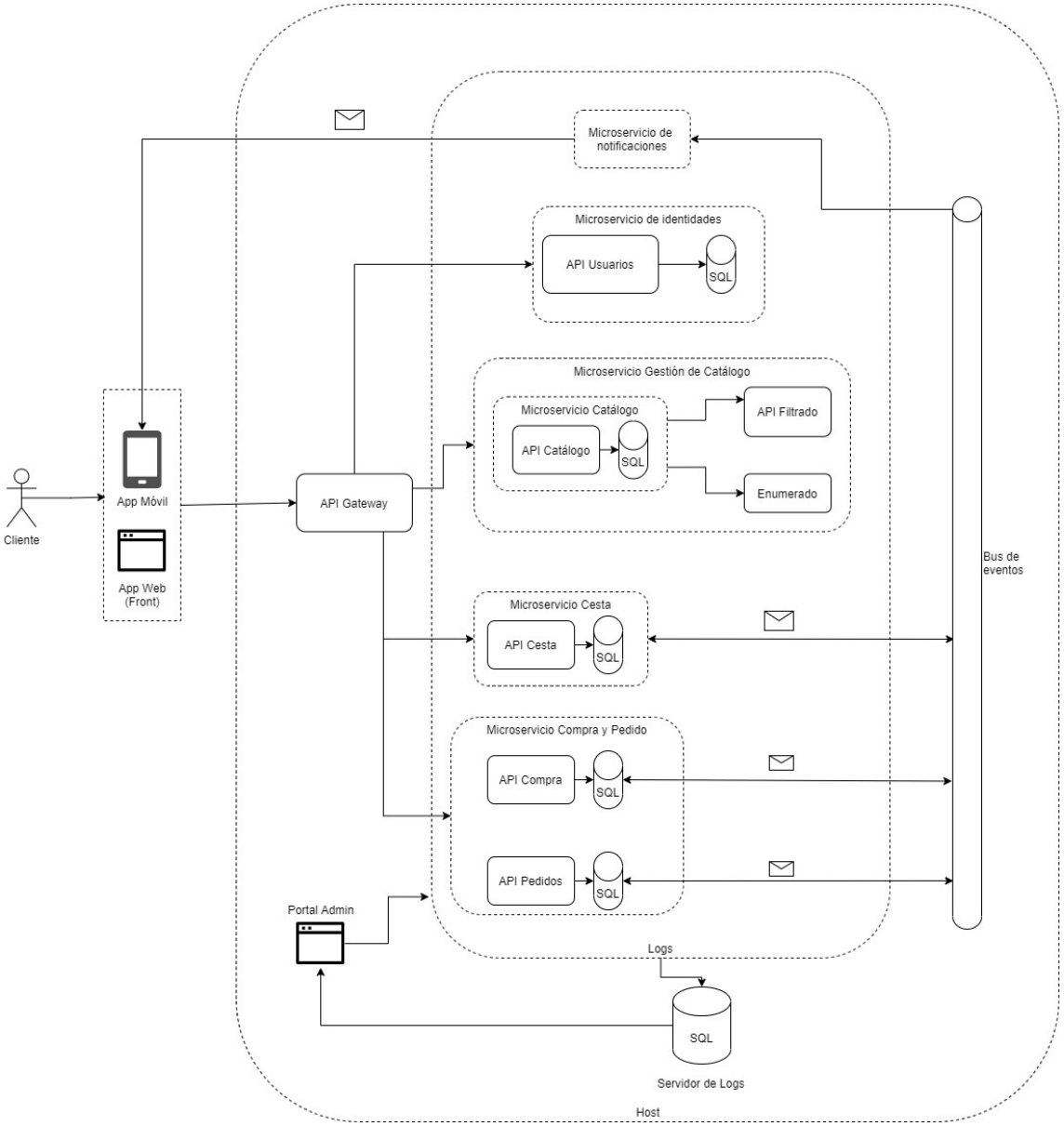
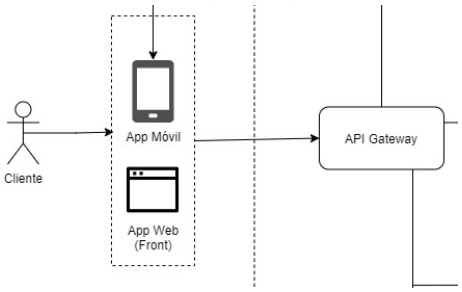
Diseño de Arquitecturas Distribuidas

Sistemas Distribuidos

Comenzaré explicando las tomas de decisiones y las funciones de los servicios y APIs utilizadas. Para las comunicaciones entre la plataforma (bus de eventos) he utilizado una comunicación asíncrona con tecnologías como pueden ser Kafka o RabbitMQ.



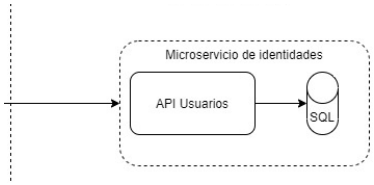
Apps Cliente y Login/Register



Los clientes utilizarán las Apps Móvil o Web para acceder a toda la plataforma, mediante el Front diseñado para ello. Desde aquí accederemos a la API Gateway, que será la encargada de separar y actuar como enlace entre los servicios internos y externos, además de otorgar una capa de seguridad.

Tras la conexión la API Gateway utilizará la conexión el Microservicio de identidades , que explicaré a continuación, para realizar el procedente registro o inicio de sesión de los usuarios. Una vez completado el microservicio de identidades devolverá un token de sesión con el que los usuarios podrán acceder al resto de servicios de la plataforma.

Microservicio de identidades



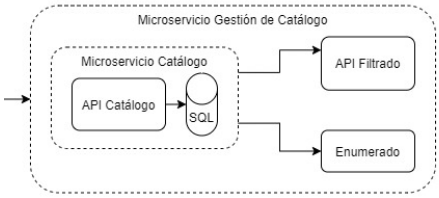
Este microservicio recibe peticiones de Login o Register por parte de la API Gateway con la información de un usuario.

En el caso de Login, la API Usuarios se encargará de establecer la conexión con la base de datos de usuarios para comprobar que la información es correcta, tras esto devolverá un token en función del resultado de esta comprobación.

En el caso de Register, la API Usuarios se encargará de comprobar si la información del usuario no está duplicada consultando la base de datos, tras esto hará las operaciones necesarias para almacenar la nueva información del usuario.

Cualquier tipo de información que el microservicio deba devolver a la API Gateway lo hará mediante el uso de tokens, tanto los de inicio de sesión (para poder usar el resto de servicios) como tokens de error si alguna de las operaciones ha resultado con errores.

Microservicio Gestión de Catálogo



El Microservicio de gestión de catálogo se encargará de las operaciones relacionadas con el catálogo, para ello hará uso de el Microservicio de Catálogo y la API de Filtrado y Enumerado.

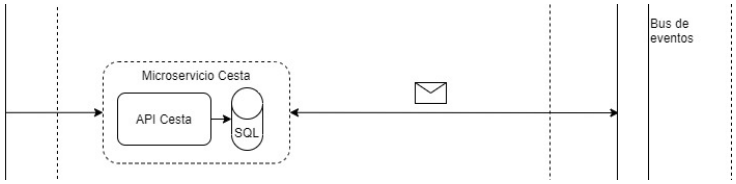
El Microservicio de catálogo se encargará , mediante el uso de la API de Catálogo, de la enumeración de los artículos y el filtrado de estos. Para ello hará las determinadas conexiones a la base de datos y la API de Filtrado y Enumerado.

Por otro lado la API de Filtrado será la encargada de según la petición de filtro (por marca o tipo) de filtrar los artículos del catálogo.

Mientras tanto la API de Enumerado será la encargada de cuando se accede a ella enumerar los artículos del catálogo.

Además, la API Catálogo tendrá asignada la tarea de almacenar la información del cliente que está accediendo al catálogo y , según sus movimientos, guardar información de marketing personalizada.

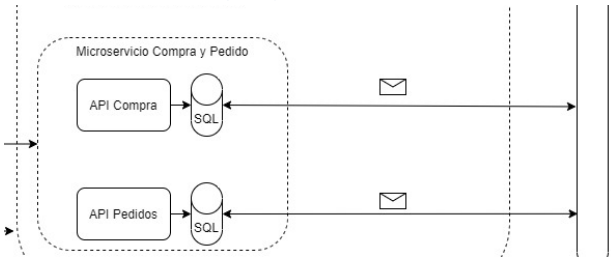
Microservicio Cesta



El microservicio de cesta se encargará mediante el uso de la API Cesta de la adición, edición y eliminación de artículos a la cesta de compra del usuario en cuestión.

Además, añadimos una conexión con el Bus de eventos para que, en caso de compra, el microservicio cesta pueda comunicarse con el microservicio de pedidos y así poder completar la compra o devolver un error (por falta de stock u otros) mediante el microservicio de notificaciones directamente al cliente.

Microservicio Compra y Pedido



Este servicio utilizará las API Compra y Pedidos para conformar la compra de artículos de nuestra plataforma.

La API compra se conectará al bus de eventos con el fin de comunicarse con el microservicio de cesta y así poder obtener los artículos en cesta del cliente y procesar la compra, tras ello se realizan las operaciones necesarias para completar la compra.

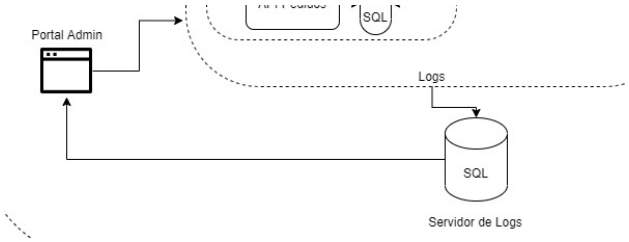
Si todo ha salido bien se volverá a conectar al bus de eventos para que la API de pedidos pueda realizar el pedido correctamente. De lo contrario enviará mediante el bus de datos un mensaje de error al microservicio de notificaciones que se mostrará al usuario.

La API de Pedidos simplemente se encargará de recibir la información de la API Compra para la realización de pedidos y se comunicará con esta en caso de que hubiese algún error.

Microservicio de Notificaciones

El microservicio de notificaciones es simple, se encargará de recibir la información del bus de eventos y tras procesarla la enviará directamente a las Apps front del cliente.

Portal Admin



Mediante un front especializado para Administradores, estos son capaces de acceder tanto a los logs de todos los microservicios de la plataforma, como a los microservicios en sí. Para facilitar mantenimientos u otras tareas.

Contratos APIs

API Usuarios

Objetos Usuario:

```
{
  id: integer
  username: string
  email: string
  created_at: datetime
  updated_at: datetime
}
```

Get /users : Devuelve los usuarios del sistema -> [{Usuario},{Usuario}…]

Get /users:id : Devuelve un usuario concreto del sistema -> {Usuario}

POST /users : Crea un Usuario dentro del sistema y lo devuelve -> {Usuario}

PATCH /users:id : Modifica un Usuario dentro del sistema y lo devuelve -> {Usuario}

DELETE /users:id : Borra un Usuario dentro del sistema

API Catálogo

Objetos Artículo:

```
{
  id: integer
  name: string
  stock: boolean
  created_at: datetime
  updated_at: datetime
}
```

Get /articles : Devuelve los artículos del sistema -> [{ Artículo },{ Artículo }…]

Get /articles:id : Devuelve un Artículo concreto del catálogo -> { Artículo }

POST /articles : Crea un Artículo dentro del catálogo y lo devuelve -> { Artículo }

PATCH /articles:id : Modifica un Artículo dentro del catálogo y lo devuelve -> { Artículo }

DELETE /articles:id : Borra un Artículo del catálogo

API Filtrado

Objetos Artículo:

```
{
  id: integer
  name: string
  stock: boolean
  created_at: datetime
  updated_at: datetime
}
```

Get /articles:filter : Devuelve los artículos filtrados según el escogido

API Cesta

Objetos Artículo:

```
{
  id: integer
  name: string
  created_at: datetime
  updated_at: datetime
}
```

Get /articles : Devuelve los artículos de la cesta -> [{ Artículo },{ Artículo }…]

Get /articles:id : Devuelve un artículos concreto de la cesta -> { Artículo }

POST /articles:id : Copia un Artículo dentro de la cesta

PATCH /users:id : Modifica un artículo dentro de la cesta y lo devuelve -> { Artículo }

DELETE /users:id : Borra un Artículo dentro de la cesta

API Compra

Objetos Artículo:

```
{
  id: integer
  name: string
  created_at: datetime
  updated_at: datetime
}
```

Get /articles : Devuelve los artículos comprados -> [{Artículo}]

POST /articles : Crea una orden de pedido de los artículos comprados -> [{Artículo}]

API Pedidos

Objetos Artículo:

```
{
  id: integer
  name: string
  created_at: datetime
  updated_at: datetime
}
```

Objetos Estado:

```
{
  id: integer
  status: string

}
```

Get /articles : Devuelve los artículos del pedido -> [{Artículo}]

Get /status : Devuelve el estado de un pedido -> Estado

DELETE /article:id : Cancela y elimina un pedido en curso y devuelve el último estado-> Estado