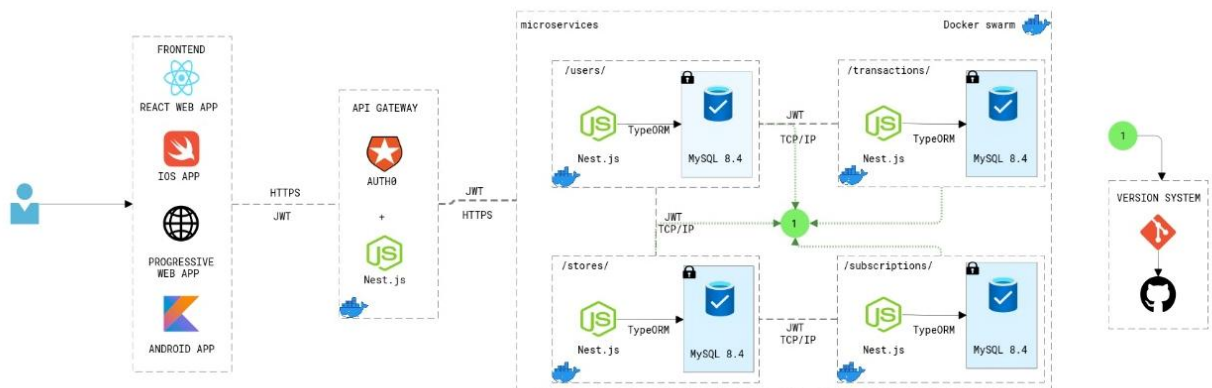


Arquitectura De MicroServicios



1. Frontend (Interfaz de usuario)

- Los usuarios interactúan con la aplicación a través de:
 - **React Web App**
 - **iOS App**
 - **Progressive Web App (PWA)**
 - **Android App**
- Estos clientes se comunican con el backend a través del **API Gateway**.

2. API Gateway

- Usa **Nest.js** como framework backend.
- Implementa autenticación con **Auth0**, emitiendo tokens **JWT**.
- Se encarga de recibir las solicitudes del frontend y redirigirlas a los microservicios correctos.
- Usa **HTTPS** para seguridad en la comunicación.

3. Microservicios

- Cada microservicio es un módulo independiente y maneja un aspecto específico del sistema:
 - **/users/** → Maneja usuarios (registro, login, perfil).
 - **/transactions/** → Maneja transacciones (pagos, compras, etc.).

- **/stores/** → Maneja tiendas (productos, catálogo, etc.).
- **/subscriptions/** → Maneja suscripciones (planes, membresías).
- Implementados con **Nest.js** y usan **TypeORM** para manejar bases de datos.
- Cada microservicio tiene su propia base de datos **MySQL 8.4**.
- Usan **JWT** para autenticación y seguridad.

4. Infraestructura con Docker Swarm

- Se usa **Docker Swarm** para orquestar los microservicios.
- Esto permite escalabilidad y alta disponibilidad.
- Los microservicios están interconectados por una red interna (indicada en verde en la imagen).

5. Sistema de Versionado

- Se usa **Git** (con GitHub) para el control de versiones.
- **Github**: Permite colaborar y gestionar cambios en el código del sistema.

Flujo de trabajo

1. Un usuario usa la aplicación web/móvil y envía una solicitud al **API Gateway**.
2. El **API Gateway** valida la autenticación con **Auth0** y verifica el **JWT**.
3. Luego, reenvía la solicitud al microservicio correspondiente.
4. Cada microservicio procesa la solicitud, interactúa con su base de datos **MySQL** y devuelve una respuesta.
5. La respuesta viaja de regreso al **API Gateway**, que la envía al usuario.