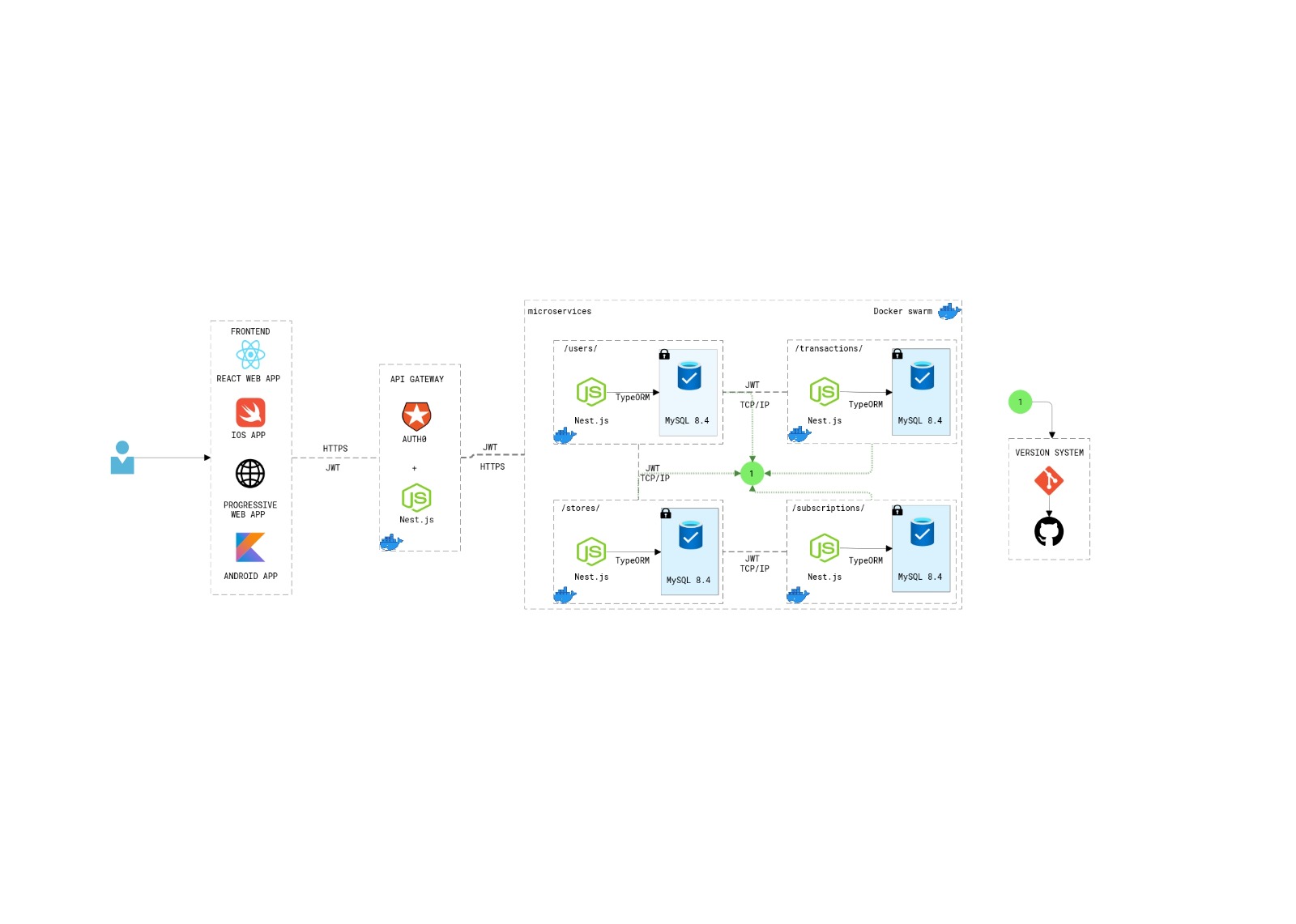
Funcionamiento de una Arquitectura de un Microservicios.



**1. Frontend (Interacción con el usuario)**

El sistema tiene varias opciones de frontend para que los usuarios interactúen con la aplicación, el usuario puede acceder a esta aplicación desde alguna de las siguientes interfaces:

* React Web App (navegador de escritorio o móvil).
* iOS App (aplicación para iPhone).
* Progressive Web App (PWA) (página web con capacidades de app).
* Android App (aplicación en Android).

Cada una de estas interfaces está diseñada para enviar solicitudes al API Gateway utilizando HTTPS, asegurando la privacidad de los datos transmitidos.

**2. API Gateway**

* Se estará trabajando con Nest.js.
* Después de validar el token, el API Gateway reenvía la solicitud al microservicio correspondiente a través de HTTPS.

Ejemplo: Si el usuario solicita información de su perfil, el API Gateway redirige la solicitud al servicio /users/.

**3. Microservicios (Backend)**

El backend está diseñado usando el patrón de microservicios, donde cada funcionalidad de la aplicación es manejada por un servicio independiente.

Cada microservicio tiene lo siguiente:

* Trabaja o implementa Nest.js.
* Usa TypeORM para la comunicación con la base de datos.
* Cada uno trabaja con su propia base de datos MySQL 8.4.
* Se comunican entre sí mediante JWT y TCP/IP dentro de un clúster de Docker Swarm.

Los microservicios que se muestran en la imagen son:

* /users/ → Maneja información de usuarios.
* /transactions/ → Gestiona transacciones.
* /stores/ → Administra tiendas.
* /subscriptions/ → Maneja suscripciones.

**4. Versionamiento**

Se utiliza Git para el control de versiones del código fuente.

Se utilizará GitHub para el trabajo colaborativo y realizar modificaciones sobre el proyecto.

**Funcionamiento General**

1. Un usuario accede a la aplicación desde una de las interfaces de frontend.
2. La solicitud pasa por la API Gateway, donde se valida la autenticación con Auth0.
3. Una vez autenticado, el API Gateway reenvía la solicitud al microservicio correspondiente.
4. Cada microservicio maneja sus propias operaciones y bases de datos de manera independiente.
5. La información se devuelve al frontend después de procesarla.