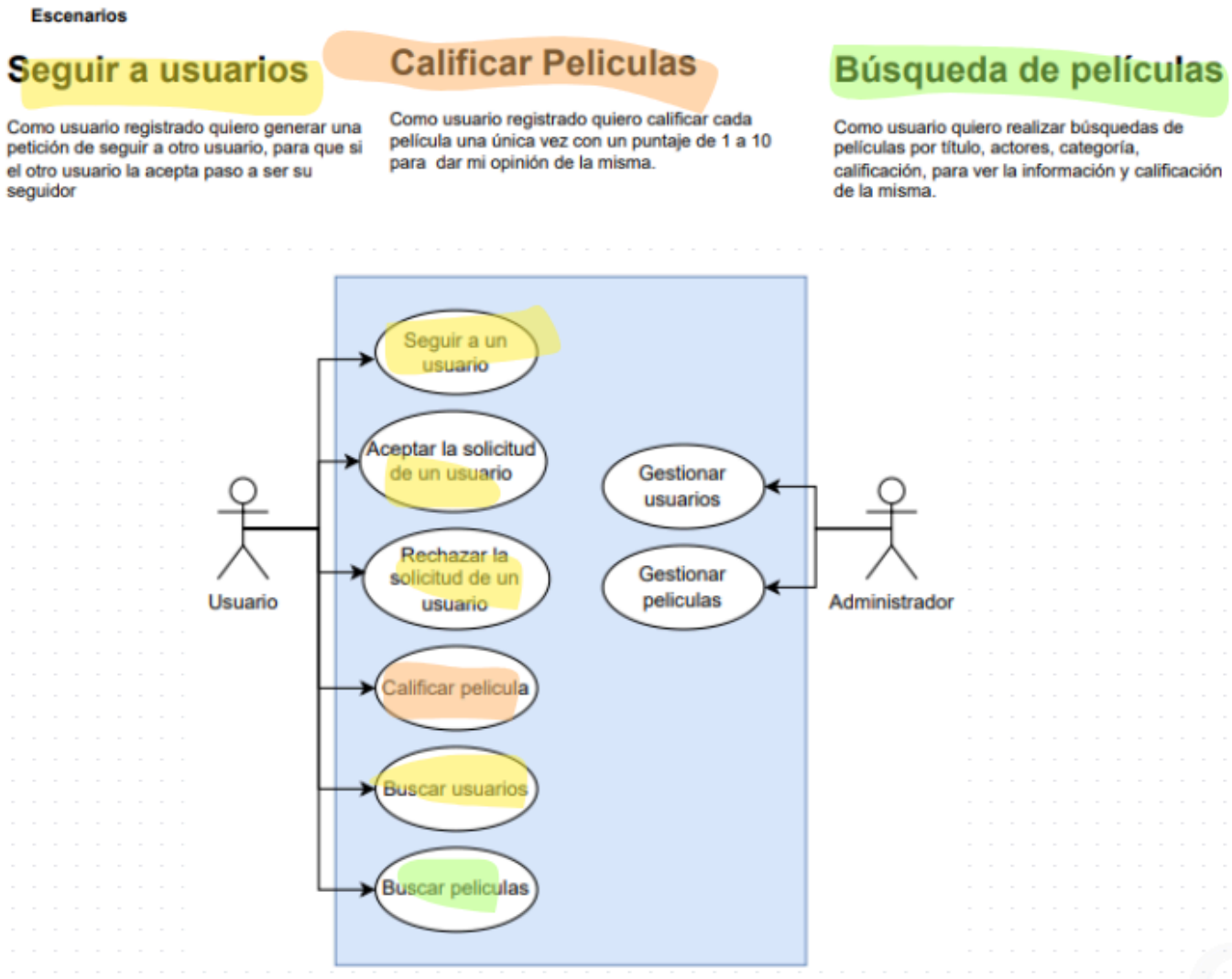
Ejemplo Modelo 4+1

+1 Vista de escenario



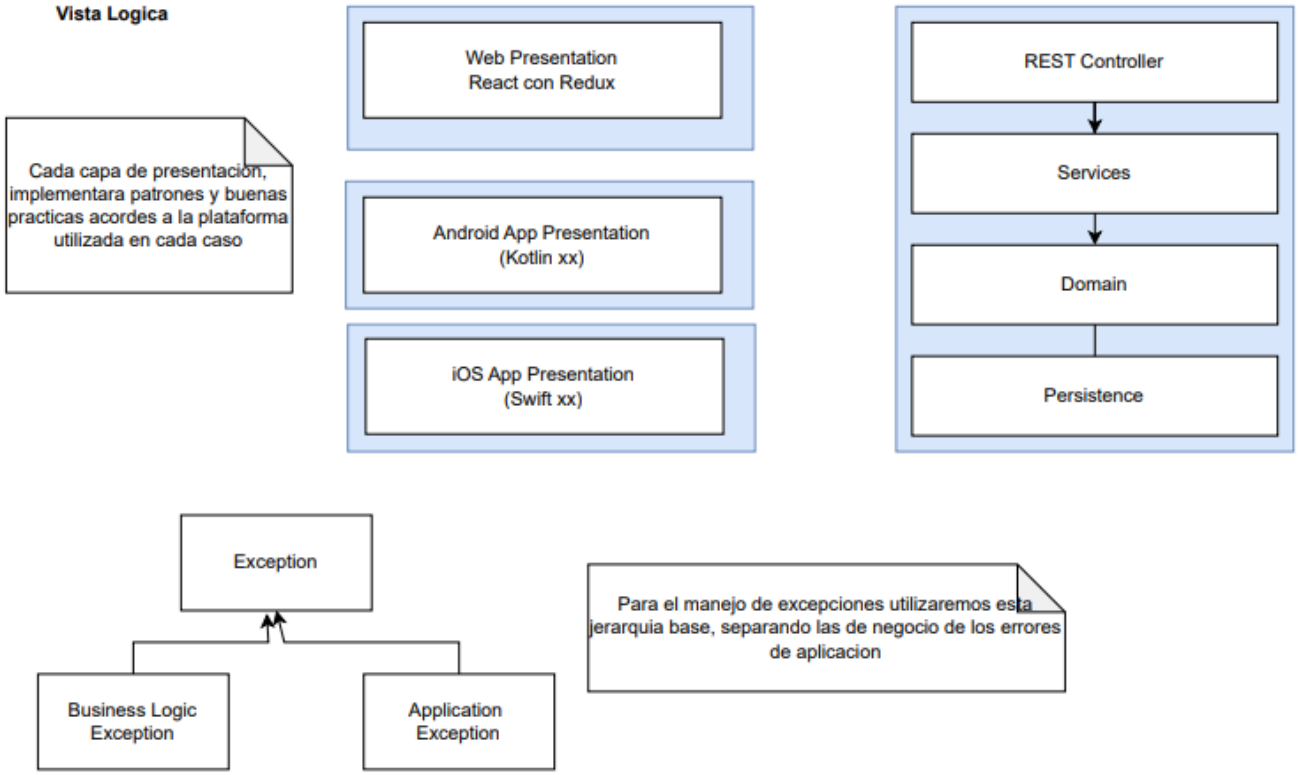
Lo generamos al inicio para tener puntos de partida para saber qué soluciones debemos plantear. Antes de modelar los demás graficos que son un poco más técnicos, primero plantemaos los atributos de calidad que queremos priorizar en nuestro servicio:

Atributos de calidad:

- *seguridad*: decimos que es seguro si el servicio requiere una autorización. los que no tienen acceso, no deben de operar.
- *performance*: se pide que cada petición de películas de la API no demore más de 500 ms para el 95% de las peticiones con las que se testea.
- *Usabilidad*: Decimos que el servicio es intuitivo y simple si al realizar una enuestra de satisfacción a una muestra inicial de usuario, hay 75% (o más) respuestas de aceptación

1. Vista de Lógica

Buscamos modelar el diseño de interés para los desarrolladores, que se ejemplifiquen los principios de encapsulación, herencia y polimorfismo. o simplemente dearrollar un esquema del patrón de arquitectura (Del código) empleada.
En este caso se modela el patrón de arquitectura de software en capas. También se modela el manejo de excepciones y las plataformas usadas en cada caso.



2. Vista de componentes (de desarrollo)

Esta vista es la encargada de facilitar la información respecto a la organización de código: paquetes, librerías, etc., además de cuestiones del lenguaje o plataforma y los componentes empleados en el sistema.

> **Tip**
>
> Podemos tomar lo que necesitamos de los diagramas de despliegue, (en particular las componentes dentro cada ambiente de desarrollo y un poco de los ambientes de desarrollo) además agregamos información útil para los desarrolladores: lenguaje de programación, dependencias y microservicios empleados.

En este caso, el diagrama se divide en *tres secciones* principales: clientes (frontend), server (backend), dependencias (tercero).

Clientes:
Puede adaptarse a diferentes presentaciones para diferentes usuarios (user web, user android, user IOs).
Para cada presentación mencionamos la plataforma que se emplea (Análogo al lenguaje de programación que se espcifica en el back).

- Web App: Web (ReactJS)
- Android App: APK (Kotlin)
- IOS App: IPA (swift)

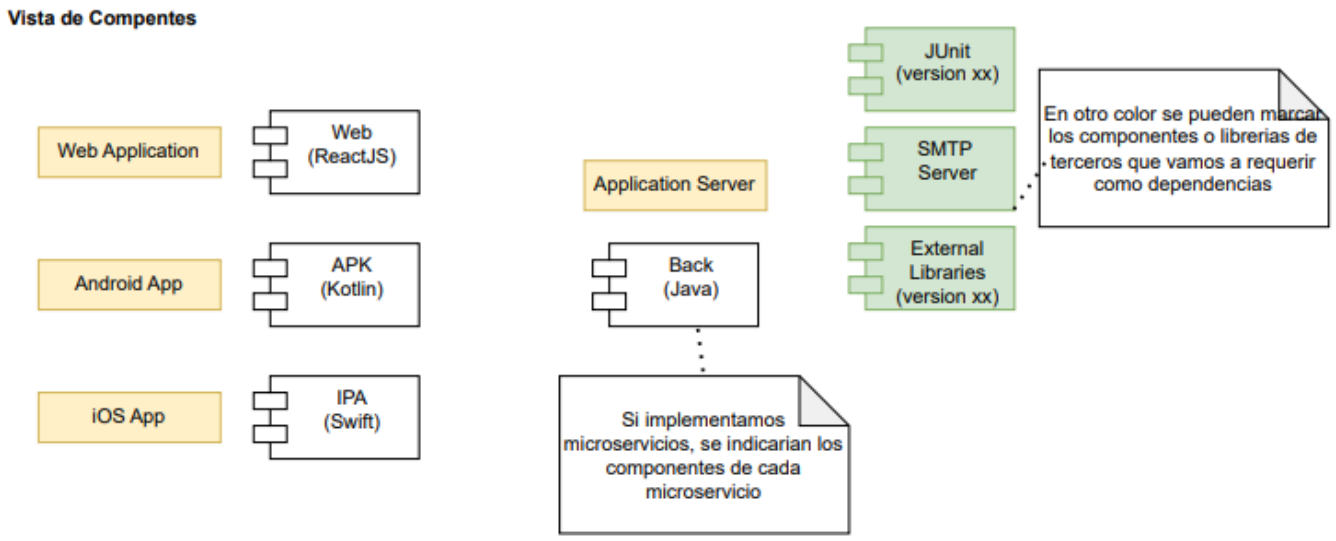
Server:

- Solo hay una aplicación web en este caso. Si el back emplea microservicios externos, se puede mencionar aquí (Digamos email, etc.).
- En este caso, como solo hay un módulo del back y tiene de dependencia a un SMTP Server, entonces se necesita enviar correos electrónicos pero

la responsabilidad está en el mimos server. Podemos delegar dicha responsabilidad a un microservicio. Este agregaría otra componente dentro del server (Email Service digamos) que exponga una API para enviar emails.

- en la aplicación desarrollada mencionamos el lenguaje de programación.

Dependencias:
En este casos son las dependencias del BACK, es decir, qué componentes externas requerimos

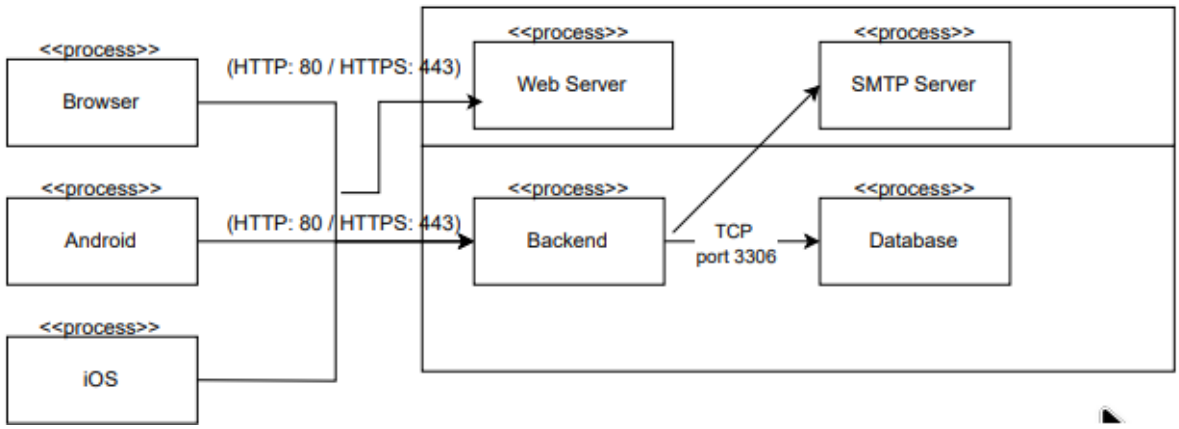


3. Vista de procesos

Aquí modelamos las interacciones en el sistema desde el punto de vista de procesos (protocolos de comunicación entre los procesos existentes).
En nuestro caso, hay un proceso del cliente que se comunican directamente con el módulo del back (android), este consume directamente de la API del backend; por otro lado, hay dos procesos que consumer desde el web server (que sirve como proie para realizar los llamados a la api de server): browser y IOS.

A su vez, el backend se comunica por TCP con el servicio SMTP y con la base de datos.

Vista de Procesos



4. Vista física (De despliegue)

Aqui podemos tomar más detalles del diagrama de despliegue: que dispositivos hay, qué entornos de desarrollo hay en estos dispositivos y podemos obviar qué componentes se emplean en cada uno. Aqui no le ponemos tanta importancia al *cómo se comunican* (protocolos, etc).

En este caso, podemos graficar un device diferente para cada dispositivo desde el cual se conecta un cliente (cada uno con el entorno de desarrollo indicado) graficamos un device donde se despliega el servidor web (que consume el back), graficamos un device donde se despliega el servidor como tal (contiene el módulo que se corre en java y se indica que tiene como dependencia el SMTP service), el último device a conectar sería donde se despliega el DB server (donde se aclara que

entorno de desarrollo se emplea: por ejemplo BD mysql).