**ANALISIS DE PRUEBA DEL CASO PRESENTADO**

Después de revisar la documentación, yo propongo los siguientes tipos de prueba y explico el porqué.

1. Pruebas de componentes: Que me permita identificar de manera individual que cada servicio utilizado funciona correctamente, en caso de que algo falle puedo detectar exactamente el servicio que está fallando y hacer la alerta temprana.
2. Pruebas de integración: Con estas pruebas voy a validar como se comporta todos los servicios integrados, por ejemplo, validar cómo se comporta la app con todos los servicios y bases de datos integradas y al mismo tiempo se valida como se comporta la app en cuanto a tiempos de respuesta. también al estar todo integrado se valida que desde la app se visualice y apliquen correctamente los descuentos cuando se tenga un descuento configurado, también se valida que se pueda hacer los pagos de forma efectiva
3. Pruebas de seguridad y rendimiento, aunque no tengo mucha experiencia con las pruebas no funcionales, de igual forma se recomienda hacer estas pruebas para validar que no exista vulnerabilidades criticas que puedan afectar o comprometer la seguridad de la app como por ejemplo un ataque que tumbe la aplicación o fraude con transacciones fantasmas. Y en rendimiento para validar como se comporta la todo el ecosistema en ambientes donde probablemente se tenga mucha concurrencia debida algún evento especial o promoción, para así estar preparados para un evento muy grande.

Ya entrando mas a detalle y después de validar el diagrama de flujo, creo que no hay una validación o algo que controle y que permita validar si una placa tiene un servicio activo y por error o algo se trata se solicitar un nuevo servicio con la misma placa, parece que el sistema permite crear el tiquete con la misma placa para un nuevo servicio. En este caso podría ser un problema porque se generaría 2 cobros a una misma placa.