Elisa Sobaler

Plan de pruebas: Ver Parada

Nuestro proyecto se dividirá en tres capas: vista, presenter y DAO, en este documento se especificará las diferentes pruebas que se realizarán para comprobar el correcto funcionamiento de las capas conjunta e independientemente. El tipo de pruebas a realizar sobre el código son unitarias, de integración y de aceptación.

Las pruebas que requieran la comprobación de la disponibilidad de internet deberán ser realizadas dos veces. Esto se debe a la imposibilidad de desactivar la conexión de datos móviles en Android a partir de la versión 5.0. También se deberán añadir los permisos necesarios para realizar las acciones relacionadas con internet.

Se comenzará realizando las pruebas unitarias, éstas se agruparán dependiendo de la capa en la que se encuentren los métodos que se vayan a comprobar. En la capa DAO se encontrarán las clases RemoteFetch, ParseJson, Línea y Parada. Se obviará realizar test sobre los métodos “get” y “set” de estas clases, los cuales trabajan con atributos privados de la clase, suponiendo así que en estos métodos no puede haber error alguno y sobre métodos cuya funcionalidad ya se ha probado anteriormente. Por lo tanto, las pruebas se realizarán sobre ParseJson.

En la capa presenter se deberán probar métodos recientemente añadidos a la clase ListParadasLineaPresenter.

El caso de la capa vista es diferente, ya que los métodos existentes en este nivel son propios de Android o triviales, así que no se verificará su correcto comportamiento. Aun conociendo estos datos, el correcto funcionamiento de la capa vista podrá ser comprobado durante las pruebas de aceptación. Por lo tanto, si se tiene en cuenta estas pautas, deberemos realizar los siguientes casos de prueba:

CASOS DE PRUEBA

ParserJson: ReadParadaGlobal()

a. Lectura satisfactoria (parada obtenida).
b. Lectura no satisfactoria (parámetro JsonReader vacío)

ParserJson: ReadParadasTodasList()

a. Lectura lista sin paradas (lista vacía)
b. Lectura lista de una parada (retorna la parada)
c. Lectura lista de más de una parada (retorna lista de paradas)

ListParadasLineaPresenter: obtenParadasLineas()

a. Obtención satisfactoria de todas las paradas (Buffer completado)
b. Obtención satisfactoria de secuencia de paradas (Buffer completado)
c. Obtención no satisfactoria (Sin acceso a internet)
d. Obtención no satisfactoria (id incorrecto)

PRUEBAS UNITARIAS

Para implementar estas pruebas se utilizarán los casos de prueba detallados utilizando únicamente el método cuya funcionalidad quiere ser comprobada. Durante la realización de estos test será necesario desarrollar diferentes Json locales que permitan conseguir las diferentes situaciones necesarias, si se sigue el mismo orden en el que se ha detallado, se debería implementar unos test sobre estas características:

En la clase ParserJson encontramos tanto el método readParadaGlobal()

Caso de prueba	Nombre parada	Cordenada x	Cordenada y	Número	Resultado
U6. a	La Pereda	435031.38	4814426.15	463	Parada (La Pereda 435031.38, 4814426.15, 463)
U6. b	-	-	-	-	-

Como el método ReadParadasTodasList()

Caso de prueba	Número de líneas en lista	Parámetro	Resultado
U7. a	0	InputStream correcto	Lista vacía
U7. b	1	InputStream correcto	Muestra la parada
U7. c	>1	InputStream correcto	Muestra las paradas

En la clase ListParadasLineaPresenter encontramos el método obtenParadasLineas() que ha sido modificado y se deberá comprobar de nuevo.

Casos de prueba	Id de línea	Conexión a internet	Resultado
U8. a	-10	Si	True (Buffer con todas las paradas)
U8. b	!=-10 && >0	Si	True (Buffer con secuencia de paradas)
U8. c	-	No	False
U8.d	<0	Si	False

Una vez comprobado el correcto funcionamiento de los métodos de la capa DAO, se debería realizar el mismo procedimiento con los posibles métodos conflictivos de las otras dos capas del proyecto. Tanto en la capa presenter como en la capa vista, se considerará que la comprobación de su correcto comportamiento se puede realizar mediante los siguientes tipos de test al contener estas capas métodos cuya funcionalidad ya ha sido probada o ser estos triviales.

PRUEBAS DE INTENGRACIÓN

Para la realización de estas pruebas se deberán comprobar la correcta funcionalidad entre capas, y no de forma independiente como se ha realizado anteriormente. Se probará el método obtenParadasLineas() de la capa presenter. Estas pruebas se deben realizar en diferentes situaciones, para conseguirlas se sobrescribirá el método para conseguir las diferentes características. Una de ellas es la conexión o desconexión a internet, en este caso por Wifi, por lo que nos aseguraremos al inicio del test que nos encontramos en la situación que buscamos.

Los casos de prueba son los mismos que los realizados en U8, este caso usaremos toda la aplicación y no lo aislaremos. Cambiaremos el nombre para diferenciar los test, pasándose este a llamar I3.

Casos de prueba	Id de línea	Conexión a internet	Resultado
I3. a	-10	Si	True
I3. b	!= -10 && >0	Si	True
I3. c	-	No	False
I3.d	<0	Si	False

Este sería el único método, de los que se deben comprobar su funcionamiento, que interactúa con la capa DAO, habrá una serie de métodos que interaccionarán con la capa vista, pero la correcta funcionalidad de estos métodos se constatará con los test de aceptación.

TEST DE ACEPTACIÓN

Se realizarán los test de aceptación en función a lo pedido por el product owner, y sus peticiones fueron claras. Al iniciar la aplicación y seleccionar el apartado de paradas, deberán aparecer todas las paradas existentes en Santander en la fecha de realización del test, con una serie de parámetros propios como son su nombre y su número. A estas indicaciones se le añadirán una serie de pruebas alternativas para aumentar la cobertura de estas. Para realizar esta comprobación se realizarán las pruebas sobre un Nexus 5 API 25, se proporciona este detalle ya que los test dependen de la posición en la que la aplicación muestra los datos, y esta varía dependiendo del tamaño de la pantalla en la que se ejecute.