Adrián Becerril

Plan de pruebas: Ver estimación de autobuses

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, Parada, Database y Estimación. 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. En esta historia se han agregado métodos a la clase ParserJson y se ha creado la clase Estimación, los métodos que se encuentran en esta última clase son “get” o su código es suficientemente trivial como para no necesitar una comprobación especifica de su correcto funcionamiento.

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

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: ReadEstimacion()

a. Lectura satisfactoria de dos estimaciones (estimaciones obtenida).
b. Lectura satisfactoria de una estimación (estimación obtenida)
c. Lectura no satisfactoria (parámetro JsonReader vacío)

ParserJson: ReadArrayEstimacionesParada()

a. Lectura lista sin estimaciones (lista vacía)
b. Lectura lista de una estimación (retorna la estimación)
c. Lectura lista de más de una estimación (retorna lista de estimaciones)

DetallesParadaPresenter: obtenEstimacionesParadas()

a. Obtención satisfactoria de todas las estimaciones (Buffer completado)
b. Obtención no satisfactoria (Sin acceso a internet)
c. 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 ReadEstimacion (se le pasarán dos bloques de datos ya que existe la opción de tener un máximo de dos estimaciones para la misma línea de una parada)

Caso de prueba	**Número parada**	**Número línea**	**Estimación 1**	**Estimación 2**	**Resultado**
U13.a	463	20	1549	3350	ListaEstimacion = Estimación (20, 463, 25) && Estimación (20, 463, 55)
U13.b	463 -	20	1549	-	ListaEstimacion = Estimación (20, 463, 25)
U13. c	-	-	-		-

Como el método ReadArrayEstimacionesParada ()

Caso de prueba	**Número de líneas en lista**	**Parámetro**	**Resultado**
U14. a	0	InputStream correcto	Lista vacía
U14. b	1	InputStream correcto	Muestra la estimación
U14. c	>1	InputStream correcto	Muestra las estimaciones

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. En la capa presenter 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.

En la clase DetallesParadaPresenter encontramos el método obtenEstimacionesParadas () que ha sido añadido al proyecto durante la realización de esta historia de usuario.

Casos de prueba	Id de parada	Conexión a internet	Resultado
U15. a	463	Si	True (Buffer con todas las estimaciones)
U15. b	463	No	False
U15. c	-5	Si	False

La correcta funcionalidad de la capa vista se comprobará al realizar los test de aceptación.

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 obtenEstimacionesParadas () 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 U15, este caso usaremos toda la aplicación y no lo aislaremos. Cambiaremos el nombre para diferenciar los test, pasándose este a llamar I5.

Casos de prueba	Id de parada	Conexión a internet	Resultado
I5. a	463	Si	True (Buffer con todas las estimaciones)
I5. b	463	No	False
I5. c	-5	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 alcanzar algún apartado que muestre una lista de paradas (paradas totales o paradas de una línea) deberán aparecer una serie de paradas existentes en Santander en la fecha de realización del test, con una serie de parámetros propios como son su nombre y su número. Al seleccionar una de estas líneas deberá aparecer una pantalla que muestre el tiempo estimado para que diferentes líneas alcancen esas paradas. 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.

La organización de los Json desde los que se obtienen los datos puede variar, por lo que existe la posibilidad de fallo en los test realizados por Espresso. Para aumentar la probabilidad de encontrar potenciales fallos distribuiremos la aplicación a diferentes dispositivos y les pediremos a sus usuarios que realicen una serie de acciones en función a unas pautas que le serán facilitadas por la empresa. Las pautas asociadas a esta historia serán las siguientes:

1. Seleccionar una línea/Seleccionar botón destinado a las paradas.
2. En la lista de paradas mostrada, seleccionar la parada sobre la que se desea consultar las estimaciones de los próximos autobuses.
3. Verificar que se muestra una lista de estimaciones de todos los autobuses que van a llegar a la parada seleccionada y coinciden con los tiempos mostrados en otra aplicación similar. La lista se encontrará entre una foto y los datos correspondientes a la parada. La lista estará formada por las líneas que pasan por esa parada ordenadas en función al tiempo de llegada. En cada elemento de la lista sale la línea con su ID y con la próxima estimación de llegada en minutos.

Comprobar funcionamiento sin acceso a Internet.

1. Seleccionar una línea/Seleccionar botón destinado a las paradas.
2. Seleccionar una parada.
3. Verificar que no se muestra ningún elemento en la lista de estimaciones.