Ver paradas de línea: Plan de pruebas

El proyecto se dividirá en tres capas: vista, presenter y DAO, en este documento se especificarán 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.

La aplicación únicamente se podrá utilizar en orientación vertical, impidiendo así la aparición de errores provocados durante el cambio de orientación.

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, Parada y Línea. 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. Tampoco se realizarán pruebas sobre métodos cuya funcionalidad ha sido comprobada anteriormente. Si se aplican estas condiciones, deberíamos realizar únicamente pruebas sobre los nuevos métodos de la clase ParserJson, ya que el resto de los métodos de la capa tendrán una funcionalidad correcta al cumplir alguna de las anteriores premisas. Por lo tanto, se debe realizar lo siguiente.

PRUEBAS UNITARIAS

Para la realización de estos test se necesitará 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:

CAPA DAO

U1. ParserJson: ReadParada()

a. Lectura correcta (parada obtenida).
b. Lectura incorrecta (parámetro JsonReader vacío)

U2. ParserJson: readParadasList ()

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

Los casos de prueba realizados en esta clase realizados para el método readParada():

Caso de prueba	**Número parada**	**Nombre**	**Resultado**
U1. a	328	Arsenio Odriozola 16	Parada (Arsenio Odriozola 16,435985.96875,4815050.5,328)
U1. b	-	-	Parada vacía

Mientras que para el método readParadasList():

Caso de prueba	**Número de paradas en lista**	**Parámetro**	**Resultado**
U2. a	0	InputStream correcto	Lista vacía
U2. b	1	InputStream correcto	Parada válida
U2. c	>1	InputStream correcto	Paradas válidas

El resto de métodos de esta capa no se probarán, al haber sido probados en fases anteriores o al ser estos “get” o “set”, como hemos detallado anteriormente.

Una vez comprobado el correcto funcionamiento de los métodos de la capa DAO, se deberá realizar el mismo procedimiento con los posibles métodos conflictivos de las otras dos capas del proyecto.

CAPA PRESENTER

En esta capa se encuentran las clases ListLineasPresenter y ListParadasLineasPresenter. El correcto funcionamiento de la primera de estas clases ya se ha comprobado anteriormente, por lo que se debe probar únicamente la funcionalidad de la clase ListParadasLineasPresenter. Se considerará que la URL desde la que se consiguen los datos es correcta y fija. El método que puede generar errores es el obtenParadasLineas() y los casos de usos con los que se comprobará su funcionalidad son los siguientes.

U3. ListLineasPresenter: obtenPradasLineas()

a. Obtención correcta (booleano true).
b. Fallo en la obtención (URL incorrecta)
c. Fallo en la obtención (Sin conexión a internet).
d. Obtención vacía (número de línea positiva pero no existente)-
e. Fallo en la obtención (número de línea negativo).

Casos de prueba	Id línea	Conexión a internet	readParadasList	Resultado
U3. a	1	Si	>0	True
U3.b	1	No	-	False
U3.c	0	Si	0	True
U3.d	>0 && No Existente	Si	0	True
U3.e	< 0	Si	-	False

CAPA VISTA

Esta capa se encarga de mostrar los datos obtenidos por pantalla, por lo que las pruebas unitarias se realizarán en función a ello. Los métodos utilizados son propios de Android, por lo que su funcionalidad se supone como óptima. Aun así, se realizarán pruebas sobre esta capa, la forma más sencilla de realizarlas es con la utilización de espresso. La realización y la descripción de dichas pruebas se detallarán durante la sección de pruebas 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 obtenParadasLineas() de la capa presenter. Los casos de prueba a aplicar son los mismos que las pruebas unitarias realizadas para esta clase, con la diferencia que ahora no se utilizarán Json locales, si no que se utilizarán las funcionalidades de la aplicación para obtener estos datos.

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, por lo que nos aseguraremos al inicio del test que nos encontramos en la situación que buscamos. Se considerará que la URL desde la que se consiguen los datos es correcta y fija.

I2. ListParadasLineasPresenter: obtenParadasLineas()

a. Obtención correcta (booleano true).
b. Fallo en la obtención (URL incorrecta)
c. Fallo en la obtención (Sin conexión a internet).
d. Obtención vacía (número de línea positiva pero no existente)-
e. Fallo en la obtención (número de línea negativo).

Casos de prueba	**Id línea**	**Conexión a internet**	**Resultado**
I2. a	1	Si	True
I2. b	1	No	False
I2.c	0	Si	True
I2.d	>0 && no existente	Si	True
I2.e	<0	Si	False

TEST DE ACEPTACIÓN

Se realizarán los test de aceptación en función a lo pedido por nuestro product owner, y sus peticiones fueron claras. Al iniciar la aplicación deberán aparecer todas las líneas 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 alguna de estas líneas, la aplicación deberá mostrar las paradas pertenecientes a dicha línea, junto a una serie de parámetros de esta.

El resultado de estas pruebas varía dependiendo de la resolución de pantalla del dispositivo sobre el cual estemos ejecutándolas, por lo que comprobaremos una serie limitada de paradas mostradas por la lista y supondremos que el resto también son correctas. Así conseguiremos que el test se supere en un número mayor de dispositivos.

Para realizar esta comprobación se realizarán las pruebas sobre un emulador de Nexus 5 con API de Android 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.