Javier Diego

Buscar parada 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.

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. 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. Por lo tanto, se debe realizar lo siguiente.

PRUEBAS UNITARIAS

Para la realización de estos test se necesitará desarrollar diferentes listas de paradas que permitan conseguir las diferentes situaciones necesarias. En anteriores test ya se ha comprobado la correcta funcionalidad de métodos que se usarán durante esta historia, estos métodos junto con aquellos que sean triviales no se probarán, ya que se entiende que su funcionalidad ya es correcta.

Se deberán detallar test sobre estas capas:

CAPA DAO

Durante esta historia de usuario no se han implementado nuevos métodos en esta capa y los métodos que contiene esta capa ya han sido cotejados anteriormente, por lo que no será necesario comprobar su correcta funcionalidad.

CAPA PRESENTER

En esta capa se ha implementado una nueva clase llamada “CommonUtils”, esta clase realizara el filtrado de una lista en función a un texto coincidente, ambos parámetros serán proporcionados en los métodos filterLineas y filterParadas. En este plan se detallarán los casos de uso proporcionados para comprobar el método filterParadas, los casos de usos del otro método, filterLineas, serán detallados en el plan de pruebas destinado a la historia de usuario que realice la utilización de estos métodos.

Las pruebas a realizar sobre este método variaran según el texto filtrador y la lista sobre la que se quiere filtrar. En este caso la lista será siempre la misma, la lista completa de las paradas de la línea 11 de Santander a día de realización de las pruebas, el texto filtrador que se usará se concretará a continuación:

1. Filtrado por número
   a) Número completo existente (lista obtenida tamaño 1)
   b) Número no existente (lista vacía)
   c) Número progresivo existente (listas de diferentes tamaños)
2. Filtrado por nombre
   a) Nombre existente completo en mayúsculas (lista obtenida tamaño 1)
   b) Nombre existente completo en minúsculas (lista obtenida tamaño 1)
   c) Nombre existente completo intercalando mayúsculas y minúsculas (lista obtenida tamaño 1)
   d) Nombre existente incompleto (lista con algún elemento)
   e) Nombre no existente (lista vacía)
3. Filtrado por nombre con número
   a) Número y nombre existentes y correspondientes (lista obtenida tamaño 1)
   b) Número existente y nombre no existente (lista vacía)
   c) Número no existente y nombre existente (lista vacía)
   d) Número y nombre existentes, pero no correspondientes (lista vacía)
   e) Número y nombre existentes y correspondientes con más de un espacio de separación (lista con un elemento).

Caso de prueba	Texto filtrador	Tamaño de lista obtenida
U5.a	429	1
U5.b	90	0
U5.c	6,8*	3,1*
U5.d	PLAZA DE TOROS	1
U5.e	plaza de toros	1
U5.f	PlAzA De ToRoS	1
U5.g	Mona	1
U5.h	Plazas	0
U5.i	9 Valdecilla	1
U5.j	90 Valdecilla	0
U5.k	9 Unican	0
U5.l	45 Valdecilla	0
U5.m	9 Valdecilla(con dos espacios)	1

**ir introduciendo carácter y comprobar el tamaño de la lista progresivamente*

CAPA VISTA

Esta capa se encarga de mostrar los datos obtenidos por pantalla y los métodos que se utilizar para ello, onQueryTextSubmit y onQueryTextChange de la clase LineasFragment, son propios de Android, por lo que su funcionalidad se supone como correcta. 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

En este caso las pruebas de integración no serán realizadas, ya que únicamente necesitamos examinar un método, y este ya ha sido comprobado con los test unitarios. Aun así, la correcta funcionalidad de todos los métodos en conjunto se realizará mediante las pruebas de aceptación.

PRUEBAS 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, si se selecciona una deberá aparecer las paradas asociadas a esa línea, al introducir texto en el apartado de búsqueda esta lista deberá reducirse a únicamente a las paradas cuyo nombre o número coincidan con el texto introducido.

Para realizar estas pruebas seguiremos los casos de uso detallados en el apartado de pruebas unitarias de este mismo documento.

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.