

Sprint 1: Plan de pruebas

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.

Se comenzará realizando las pruebas unitarias, estas 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 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, se realizarán entonces pruebas sobre RemoteFetch y ParseJson, mientras que el caso de Línea es algo más llamativo, ya que únicamente contiene este tipo de métodos. Por lo tanto, debemos realizar lo siguiente:

PRUEBAS UNITARIAS

U1. ParseJson: Readlinea()

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

U2. ParseJson: ReadArrayLineasBus()

- a. Lectura lista sin líneas (mensaje listo vacía)
- b. Lectura lista de una línea (muestra la línea)
- c. Lectura lista de más de una línea (muestra lista de líneas)

U3. RemoteFetch: GetJSON()

- a. Obtención correcta (Buffer completado)
- b. Obtención incorrecta (Sin acceso a internet)
- c. Obtención incorrecta (URL incorrecta)

Para 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 ParseJson encontramos tanto el método readLinea():

Caso de prueba	Id	Nombre	Resultado
U1. a	20	ESTACIONES-BARRIO LA TORRE	Línea (20, ESTACIONES-BARRIO LA TORRE)
U1. b	-	-	Línea vacía

Como el método `readArrayLineasBus()`:

Caso de prueba	Número de líneas en lista	Parámetro	Resultado
U2. a	0	InputStream correcto	Mensaje lista vacía
U2. b	1	InputStream correcto	Muestra la línea
U2. c	>1	InputStream correcto	Muestra las líneas

En la clase `RemoteFetch` encontramos el método `GetJSON()`:

Casos de prueba	URL	Conexión a internet	Resultado
U3. a	"http://datos.santander.es/api/rest/datasets/lineas_bus.json"	Si	Buffer con datos
U3. b	"http://datos.santander.es/api/rest/datasets/lineas_bus.json"	No	Error (IOException)
U3. c	"http:erronea"	Si	Error (IOException)

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 pocos métodos, o ser estos triviales.

PRUEBAS DE INTEGRACIÓN

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 `obtenLineas()` de la capa presenter. Estas pruebas se deben realizar en diferentes situaciones, para conseguir las 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.

I1. `ListLineasPresenter: obtenLineas()`

- Obtención correcta (booleano true).
- Fallo en la obtención (URL incorrecta)
- Fallo en la obtención (Sin conexión a internet).

Casos de prueba	URL	Conexión a internet	<code>readArrayLineas</code>	Resultado
I1. a	"http://datos.santander.es/api/rest/datasets/lineas_bus.json"	Si	>0	True

	Tasets/líneas_bus.json"			
I1. b	"http: -"	Si	0	False
I1. c	"http://datos.santander.es/api/rest/datasets/líneas_bus.json"	No	-	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 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. Para realizar esta comprobación se utilizará espresso sobre la interfaz de nuestra aplicación.