

Plan de pruebas

Autor: Asier López Uriona

Introducción:

En el presente documento se hará una especificación de las pruebas necesarias para acreditar el correcto funcionamiento del código asociado a la historia de usuario “*US242835-RealizarBusquedaParadasDeLinea*”. Estas pruebas se dividirán según sean de aceptación, de sistema, unitarias o de integración.

Diagrama de componentes orientativo

Este diagrama indica las dependencias lógicas de la aplicación.

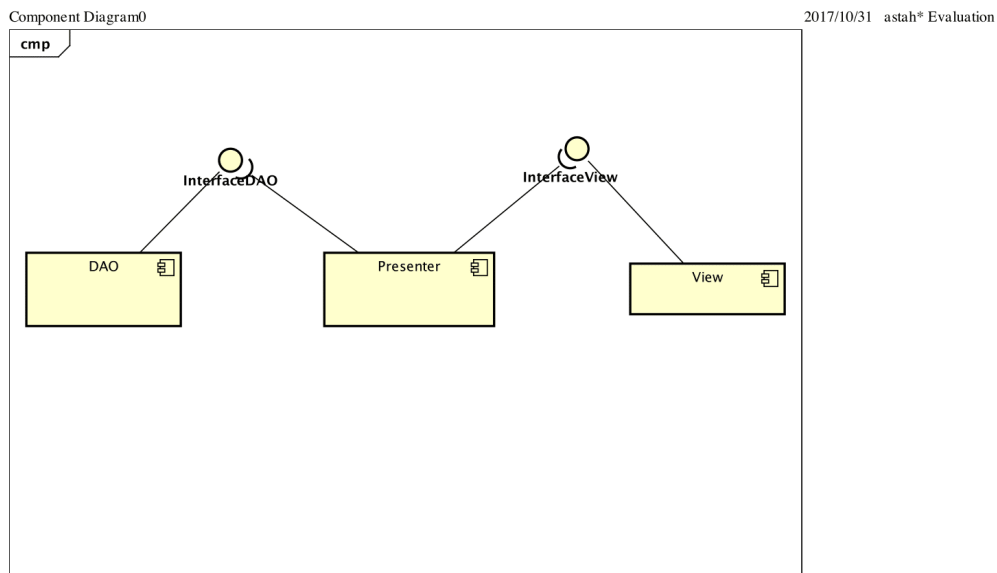


Diagrama de clases orientativo



Pruebas de aceptación:

Las pruebas de aceptación consistirán en la comprobación del correcto funcionamiento de la nueva funcionalidad de búsqueda de paradas de una línea. Por lo tanto, las pruebas a realizar serán:

A1: Búsqueda mediante nombre de parada completo.

1. Se escribe en el campo de búsqueda el nombre completo de una parada, por ejemplo: “*Canalejas 42*”.
2. Se mostrará solamente esa parada.

A2: Búsqueda mediante nombre de parada incompleto.

1. Se escribe en el campo de búsqueda el nombre incompleto de una parada, por ejemplo: “*Canale*”.
2. Se irán mostrando en tiempo real todas las paradas de esa línea que empiecen por esas letras, en este caso *Canalejas 42* y *Canalejas 26*.

A3: Búsqueda con tildes.

1. Se escribe en el campo de búsqueda un nombre de una parada que contenga tildes correctamente, por ejemplo: *"General Dávila 41"*.
2. Se mostrará esa parada.

A4: Búsqueda sin tildes.

1. Se escribe en el campo de búsqueda el nombre de una parada que contenga tildes, pero sin escribirlas. Por ejemplo: *"General Davila 41"*.
2. Se mostrará esa parada, a pesar de no haberla escrito correctamente.

A5: Búsqueda con caracteres especiales.

1. Se escribe en el campo de búsqueda el nombre de una parada que contenga caracteres especiales (ñ, °). Por ejemplo: *"Peñacastillo"*.
2. Se mostrará esa parada.

A6: Búsqueda con mayúsculas.

1. Se escribe en el campo de búsqueda el nombre de una parada que contenga mayúsculas, por ejemplo: *"CANALEJAS 42"*.
2. Se mostrará esa parada, a pesar de no haberla escrito correctamente.

A7: Búsqueda sin mayúsculas.

1. Se escribe en el campo de búsqueda el nombre de una parada que contenga mayúsculas, pero escribiéndolo en minúsculas. Por ejemplo: *"canalejas 42"*.
2. Se mostrará esa parada, a pesar de no haberla escrito correctamente.

A8: Búsqueda mediante identificador completo.

1. Se escribe en el campo de búsqueda el identificador completo de una parada. Por ejemplo: *"310"*.
2. Se mostrará la parada correspondiente que es *"Francisco Tomas y Valiente 23"*.

A9: Búsqueda mediante identificador incompleto.

1. Se escribe en el campo de búsqueda el identificador incompleto de una parada, por ejemplo *"30"*.
2. Se irán mostrando en tiempo real todas las paradas de esa línea que empiecen o contengan esos números, en este caso: *305, 306, 307, 308, 309 y 130*.

A10: Búsqueda con un número.

1. Se escribe en el campo de búsqueda un numero (ya sea el identificador de una parada o parte de su nombre). Por ejemplo: *"35"*.
2. Se irán mostrando en tiempo real todas las paradas de esa línea cuyo identificador contenga ese número o sea parte de su nombre. En este caso serán: *"352", "353", "355" y "Paseo de Pereda n° 35"*.

A11: Parada buscada no existe.

1. Se escribe en el campo de búsqueda un identificador o un nombre. Por ejemplo: *"ya"*.
2. No se encuentra ninguna parada que cumpla ese criterio.
3. Se muestra un mensaje de error: *"No hay resultados"*.

Pruebas de sistema

Estas pruebas verifican el correcto comportamiento de la aplicación en diferentes versiones y dispositivos, se ha solicitado que funcione de forma que sea lo más compatible posible. Estas pruebas se harán sobre la versión más antigua compatible.

Dado que las pruebas que vamos a realizar son sobre el sistema operativo de Android con la versión 15 (IceCream), consideramos que las pruebas de sistema están incluidas en las propias pruebas de aceptación.

Pruebas unitarias:

Las pruebas unitarias que realizaremos sobre ListParadasPresenter son para probar el método *“filtra(String s)”*, que devolverá una lista de paradas con todas aquellas que coincidan con el parámetro de búsqueda *“s”* que se le pasa.

Se comprobará su correcto funcionamiento ejecutando el método y comprobando que la lista de paradas que retorna se rellena con los datos correctos. También se comprobará el caso en que no encuentre ninguna parada con esos datos.

Además, se comprobará según la longitud de la lista: si está vacía, si sólo tiene un elemento y si tiene varios.

Para la realización de estas pruebas se utilizarán mocks de la clase ParadasActivity.

Los datos con los que trabajaremos serán:

```
Parada p1 = new Parada(0, 10, 1.0, 1.0, 1.0, 1.0, 38);  
Parada p2 = new Parada(1, 20, 1.0, 1.0, 1.0, 1.0, 31);  
Parada p3 = new Parada(1, 30, 1.0, 1.0, 1.0, 1.0, 33);  
Parada p4 = new Parada(2, 40, 1.0, 1.0, 1.0, 1.0, 4);
```

Y se les pondrán los siguientes nombres:

"Parada 1", "Parada 2", "Parada 3" y "Parada Ávila 4".

Pruebas de integración:

Las primeras pruebas de integración que realizaremos serán las de la capa Presenter (ListParadasPresenter) junto a la capa DAO o Models (DatabaseHelper) haciendo las pruebas unitarias anteriores, pero con la base de datos local.

Seguidamente, comprobaremos la capa View (ParadasActivity) con la capa Presenter (ListParadasPresenter), para así cerciorarnos de que se muestran las paradas de la base de datos correctamente repitiendo las pruebas unitarias anteriores y, en el caso de que la búsqueda no devolviera ningún resultado, mostrar un mensaje indicando que la lista retornada está vacía.

Estas pruebas se dan por superadas al haber comprobado el correcto funcionamiento de los test de aceptación, ya que estos también trabajan sobre las 3 capas luego no sería necesario repetirlos.