Plan de pruebas VER PARADAS LINEA

**Aceptación y de sistema:**

¿¿¿AS1: Mostrar paradas

1. Se muestran todas las paradas correctamente.
2. No se muestran todas las paradas.???

¿¿¿AS2: Buscar paradas

1. Búsqueda correcta (la parada existe).
2. Búsqueda no válida (la parada no existe).???

AS3: Actualizar paradas

1. Actualización correcta (los datos se actualizan accediendo al JSON de TUS).
2. Actualización fallida (no se puede acceder al JSON de TUS).

**Integración:**

I1: Mostrar paradas

1. Se muestran todas las paradas correctamente.

I2: Buscar paradas

1. Búsqueda correcta (la parada existe).
2. Búsqueda no válida (la parada no existe).

I3: Actualizar paradas

1. Actualización correcta (los datos se actualizan accediendo al JSON de TUS).
2. Actualización fallida (no se puede acceder al JSON de TUS).

**Casos de prueba:**

I1.a Usuario selecciona la línea “7C2” 🡪 Se accede a la base de datos y se muestran las paradas de la línea indicada.

I2.a Se busca la línea “4”, que existe 🡪 se muestra solo la línea 4.

I2.b Se busca la línea “35”, que no existe 🡪 no se muestra ninguna línea y se muestra un mensaje diciendo “No existe ninguna línea 35”.

I3.a El usuario pulsa el botón de actualizar CON conexión a Internet 🡪 Se actualizan las paradas de la línea de bus descargándolas del JSON de TUS.

I3.b El usuario pulsa el botón de actualizar SIN conexión a Internet 🡪 Se muestra un mensaje indicando al usuario que no se pueden actualizar las paradas sin conexión a internet.

**Unitarias:**

Para probar los módulos de forma independiente, se realizará con pruebas de métodos (sólo los métodos públicos).

Model

* **Línea:** Por inspección se ha comprobado que los métodos de la clase Línea, todos son getters y setters, funcionan sin problema.
* **ParserJSON:** Para probar de forma aislada esta clase, se utilizará un fichero JSON local y conocido que contenga líneas y paradas, de forma que no se tenga que acceder a datos externos. Lo casos de prueba de los métodos abarcarán los valores límite e intermedios para cada clase de equivalencia.

\* Suponiendo que el fichero JSON local contenga 4 líneas: la 1, la 2, la 19 y la 20. \*

Para el método ‘public static List<Linea> readArrayLineasBus (InputStream in)’:

|  |  |
| --- | --- |
| **Casilla a comprobar** | **Valor esperado** |
| lista[0] | Línea 1 |
| lista[1] | Línea 2 |
| lista[3] | Línea 20 |

* **RemoteFetch:** por inspección se ha comprobado que el método getBufferedData() funciona correctamente.

Para probar el método ‘public void getJSON (String urlJSON)’:

* Se ejecuta dicho método utilizando una url de un JSON conocido (p.ej: <http://datos.santander.es/api/rest/datasets/lineas_bus.json>).
* Se ejecuta el método getBufferedData(), que ya está probado que funciona, y se comprueba que los datos del BufferedInputStream que devuelve coinciden con los datos conocidos de la url que contiene el JSON.

Presenter

* **ListLineasPresenter:** por inspección se ha comprobado que el método getListaLineasBus() funciona correctamente.

Para probar el método ‘public boolean obtenLineas()’:

* Se ejecuta dicho método utilizando una url de un JSON conocido (p.ej: <http://datos.santander.es/api/rest/datasets/lineas_bus.json>).
* Se comprueba que devuelve true, que la lista listaLineasBus es del mismo tamaño que el número de líneas de bus del JSON y que los valores límite y uno intermedio son los esperados:

|  |  |
| --- | --- |
| **Casilla a comprobar** | **Valor esperado** |
| lista[0] | Línea 3 |
| lista[9] | Línea E9 |
| lista[32] | Línea 1 |

Para probar el método ‘public String getTextoLineas()’:

* Se ejecuta el método después de haber ejecutado el método obtenLineas().
* Se comprueba que el String devuelto contiene el nombre de las líneas de buses separadas por doble salto de línea.

Views

* **ListLineasAdapter:**

Para probar el método ‘public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent)’:

* Comprobar que los símbolos que se pintan son los mismos que los oficiales de TUS.

PRUEBAS LISTAR LINEAS (LUCAS)

**Aceptación y sistema.**

AS1: Mostrar líneas. (Al abrir la aplicación)

1. Selección válida. (Mostrar líneas)
2. Selección con base de datos no operativa (Se mostrarán las líneas accediendo a Internet)
3. Selección sin conexión a Internet. (Se mostrarán las líneas accediendo a la base de datos)
4. Selección sin conexión a Internet y con la base de datos no operativa. (Se mostrará la última actualización de las líneas)