Descripción de los casos de uso:

Caso de uso: Actualizar datos.
Actor: Usuario
Secuencia de pasos:

1. El usuario abre la aplicación.
2. El sistema carga los datos.
3. El sistema muestra los datos.
4. El usuario pulsa el botón de actualizar.
5. El sistema carga los datos.
6. El sistema muestra los datos actualizados.
    a. Si no se han podido actualizar los datos se comunica el error.

Plan de pruebas a seguir:

Aceptación y de sistema

En base a los casos de uso identificamos los siguientes casos:

AS1.CU: Actualizar líneas sin Internet y sin BBDD

1. Abrir la aplicación.
2. Verificar que existe un botón con dos flechas en la action bar.
3. Seleccionar dicho botón.
4. Verificar que aparece un mensaje con el texto "No hay Internet, inténtelo más tarde".

(Para paradas y estimaciones valdría con esta prueba)

AS2.CU: Actualizar líneas con Internet y con BBDD:

1. Abrir la aplicación.
2. Verificar que existe un botón con dos flechas en la action bar.
3. Seleccionar dicho botón.
4. Verificar que aparece un mensaje con el texto "Cargando datos".
5. Verificar que se muestran los datos actualizados.
6. Verificar que aparece un toast con el mensaje "Datos obtenidos con éxito".

AS3.CU: Actualizar paradas con Internet y con BBDD:

1. Seleccionar una línea.
2. Verificar que existe un botón con dos flechas en la action bar.
3. Seleccionar dicho botón.
4. Verificar que aparece un mensaje con el texto "Cargando datos".
5. Verificar que se muestran los datos actualizados.
6. Verificar que aparece un toast con el mensaje "Datos obtenidos con éxito".

AS4.CU: Actualizar estimaciones con Internet y con BBDD:

1. Seleccionar una parada.
2. Verificar que existe un botón con dos flechas en la action bar.
3. Seleccionar dicho botón.
4. Verificar que aparece un mensaje con el texto "Cargando datos".
5. Verificar que se muestran los datos actualizados.
6. Verificar que aparece un toast con el mensaje "Datos obtenidos con éxito".

AS5.CU: Actualizar líneas sin Internet y con BBDD:

1. Abrir la aplicación.

2. Verificar que existe un botón con dos flechas en la action bar.
3. Seleccionar dicho botón.
4. Verificar que aparece un mensaje con el texto "Cargando datos".
5. Verificar que se muestran los datos sin actualizar.
6. Verificar que aparece un toast con el mensaje "No se han podido actualizar los datos".

AS6.CU: Actualizar paradas sin Internet y con BBDD:

1. Seleccionar una línea.
2. Verificar que existe un botón con dos flechas en la action bar.
3. Seleccionar dicho botón.
4. Verificar que aparece un mensaje con el texto "Cargando datos".
5. Verificar que se muestran los datos sin actualizar.
6. Verificar que aparece un toast con el mensaje "No se han podido actualizar los datos".

AS7.CU: Actualizar estimaciones sin Internet y con BBDD:

1. Seleccionar una parada.
2. Verificar que existe un botón con dos flechas en la action bar.
3. Seleccionar dicho botón.
4. Verificar que aparace un toast con el mensaje "No se han podido actualizar los datos".

Estos test se comprueban de manera visual por parte del usuario.

Pruebas unitarias:

Las pruebas unitarias correspondientes a este documento estarán disponibles en la clase "ActualizarUnitariasTest.java"

Para probar los módulos se realizarán pruebas sobre los siguientes métodos de las clases MisFuncionesBBDD.java, ListLineasPresenter.java, ListParadasPresenter.java y ListEstimacionesPresenter.java:

recargarLineas() -->el cual retorna true o false en caso de que se hayan podido actualizar o no los datos.
recargaParadas() -->el cual retorna true o false en caso de que se hayan podido actualizar o no los datos.
obtenEstimaciones() -->el cual retorna true o false en caso de que se hayan podido actualizar o no los datos.
borrarLinea(id, dataBase) --> recibe el identificador de una línea y la base de datos, y retorna true o false en caso de que se haya podido o no borrar una línea.
borrarListaLineas(lista, dataBase) --> recibe una lista de líneas y la base de datos, y retorna true o false en caso de que se haya podido o no borrar la lista de líneas.
borrarParada(id, dataBase) --> recibe el identificador de una parada y la base de datos, y retorna true o false en caso de que se haya podido o no borrar una parada.
borrarListaParadas(lista, dataBase) --> recibe una lista de paradas y la base de datos, y retorna true o false en caso de que se haya podido o no borrar la lista de paradas.

Casos de prueba

U1.a Se hace uso del método recargaLineas() --> retorna true ya que hay Internet
U1.b Se hace uso del método recargaLineas() --> retorna false ya que no hay Internet
U2.a Se hace uso del método recargaParadas() --> retorna true ya que hay Internet
U2.b Se hace uso del método recargaParadas() --> retorna false ya que no hay Internet
U3.a Se hace uso del método obtenEstimaciones() --> retorna true ya que hay Internet

U3.b Se hace uso del método obtenEstimaciones() --> retorna false ya que no hay Internet
U4.a Se hace uso del método borrarLinea(1, db) --> retorna true ya que existe la línea
U4.b Se hace uso del método borrarLinea(40, db) --> retorna false ya que no existe la línea
U5.a Se hace uso del método borrarParada(1, db) --> retorna true ya que existe la parada
U5.b Se hace uso del método borrarParada(1000, db) --> retorna false ya que no existe la parada
U6.a Se hace uso del método borrarListaLineas(listaLineas, db) -->retorna true
U6.b Se hace uso del método borrarListaLineas(listaLineas, db) -->retorna false
U7.a Se hace uso del método borrarListaParadas(listaParadas, db) -->retorna true
U7.b Se hace uso del método borrarListaParadas(listaParadas, db) -->retorna false

Pruebas de integración:

Las pruebas de integración correspondientes a este documento estarán disponibles en la clase "ActualizarIntegraciónTest.java"

Para la realización de los tests se usarán los casos de prueba definidos anteriormente como U1.a-U3.b (se renombran todos cambiando la U por LI (para líneas), PI (para paradas) y EI (para estimaciones)).

Para probar los siguientes tests se integrarán las clases LineasFragment.java, ListLineasPresenter.java, ListLineasAdapter.java, MisFuncionesBBDD.java y TUSSQLiteHelper.java

LI1.a El usuario presiona el botón de actualizar con Internet y con BBDD --> retorna true ya que la BBDD se ha llenado.
LI1.b El usuario presiona el botón de actualizar sin Internet y con BBDD --> retorna false ya que la BBDD no se ha llenado.

Para probar los siguientes tests se integrarán las clases ParadasFragment.java, ListParadasPresenter.java, ListParadasAdapter.java, MisFuncionesBBDD.java y TUSSQLiteHelper.java

PI2.a El usuario presiona el botón de actualizar con Internet y con BBDD --> retorna true ya que la BBDD se ha llenado.
PI2.b El usuario presiona el botón de actualizar sin Internet y con BBDD --> retorna false ya que la BBDD no se ha llenado.

Para probar los siguientes tests se integrarán las clases EstimacionesFragment.java, ListEstimacionesPresenter.java, ListEstimacionesAdapter.java

EI3.a El usuario presiona el botón de actualizar con Internet y con BBDD --> retorna true ya que se han podido actualizar los datos.
EI3.b El usuario presiona el botón de actualizar sin Internet y con BBDD --> retorna false ya que no se han podido actualizar los datos.

Pablo Martínez Arana