



Plan de Pruebas de la historia de usuario 464974:

Ver Convenio de Precios

Los niveles de prueba a aplicar son los siguientes:

1. Pruebas de aceptación: Las pruebas de aceptación se definirán siguiendo una estrategia basada en la historia de usuario específica y se ejecutarán de forma manual.
2. Pruebas de integración
3. Pruebas unitarias: Para las pruebas unitarias se utilizará la técnica de prueba de métodos, usando técnicas de caja negra para la definición de casos de prueba de cada método de clase. Será necesaria la utilización de JUnit, Mockito y FEST.

A continuación, se muestra una especificación detallada de los casos de prueba a aplicar en cada nivel mencionado anteriormente.

PRUEBAS DE ACEPTACIÓN

En base a la historia de usuario se identifican los siguientes escenarios:

A1.HU: Ver convenios

- a. Consulta válida, la lista tiene elementos.
- b. Consulta válida, el usuario puede volver a la vista inicial con el botón de Android.
- c. Consulta válida, el usuario está usando el dispositivo en orientación horizontal.
- d. Consulta válida, inicialmente existe un fallo en el acceso a datos, se muestra una ventana emergente con un botón de reintentar y uno de aceptar, al pulsar el botón de reintentar, el usuario tiene el acceso a los datos de manera correcta.
- e. Consulta no válida, inicialmente existe un fallo en el acceso a datos, al pulsar el botón de reintentar, el usuario vuelve a tener la ventana emergente.
- f. Consulta no válida, inicialmente existe un fallo en el acceso a datos, se muestra una ventana emergente con un botón de reintentar y uno de aceptar, al pulsar el botón de aceptar, el usuario vuelve a la página principal.
- g. Consulta válida, la aplicación indica que la lista de convenios está vacía.
- h. Consulta válida, la aplicación muestra el listado de convenios aun así tengan datos anómalos.

Los criterios de confirmación, definidos para cada uno de los escenarios, son los que se especifican:

A1.a El usuario elige la opción de Convenios del menú de opciones, la nueva ventana se carga de manera correcta mostrando la lista de los convenios correspondientes al usuario.

A1.b El usuario elige la opción de Convenios del menú de opciones, la nueva ventana se carga de manera correcta mostrando la lista de los convenios correspondientes al usuario y al pulsar el botón de atrás de Android retorna a la página principal.

A1.c El usuario elige la opción de Convenios del menú de opciones, la nueva ventana se carga de manera correcta mostrando la lista de los convenios correspondientes al usuario y al cambiar la rotación del teléfono orientación horizontal la ventana se adapta a la orientación.

A1.d El usuario elige la opción de Convenios del menú de opciones, al cargar la nueva ventana se visualiza una ventana emergente indicando un error en el acceso a datos. La ventana tiene un botón



para reintentar y uno para aceptar. Al pulsar el botón de reintentar el acceso a datos es correcto y se puede visualizar la lista de convenios correspondientes.

A1.e El usuario elige la opción de Convenios del menú de opciones, al cargar la nueva ventana se visualiza una ventana emergente indicando un error en el acceso a datos. La ventana tiene un botón para reintentar y uno para aceptar. Al pulsar el botón de reintentar el error persiste y se vuelve a mostrar la ventana emergente.

A1.f El usuario elige la opción de Convenios del menú de opciones, al cargar la nueva ventana se visualiza una ventana emergente indicando un error en el acceso a datos. La ventana tiene un botón para reintentar y uno para aceptar. Al pulsar el botón de reintentar el error persiste y se vuelve a mostrar la ventana emergente, el usuario pulsa el botón de aceptar y se vuelve a la página principal.

A1.g El usuario elige la opción de Convenios del menú de opciones, al cargar la nueva ventana se visualiza un mensaje que indica que la lista de convenios está vacía.

A1.h El usuario elige la opción de Convenios del menú de opciones, al cargar la nueva ventana se carga de manera correcta mostrando la lista de convenios que contienen datos anómalos.

PRUEBAS DE INTEGRACIÓN

El orden de las pruebas y los casos de prueba a realizar serían los siguientes:

- Probar el presentador con la DAO

Para este nivel de prueba se utiliza un mock de la vista y se maneja la DAO de los convenios, obteniéndola a partir de la base de datos. Se realizan pruebas simulando dos escenarios, uno en el que la DAO tiene datos, añadidos de manera estática y otro en el que no tiene datos.

PRUEBAS DE INTERFAZ

En este caso se aplica la técnica basada en historias de usuario para la definición de las pruebas a realizar. Los casos de prueba definidos serán los mismos que los de las pruebas de aceptación, pero automatizados a través de JUnit y Espresso. Se debe de tener en cuenta que no todos los escenarios son posibles de probar.

Escenarios para probar:

- Consulta válida, la lista no tiene elementos.
- Consulta válida, la lista no tiene elementos y al pulsar el botón de atrás vuelve a la vista principal.

En este caso no se tiene implementada la funcionalidad de poder añadir los convenios, por lo cual no existen convenios añadidos.

PRUEBAS UNITARIAS

Se aplican técnicas de clases de equivalencia y AVL para la definición de los casos de prueba de cada método.

Pruebas unitarias de la capa de negocio



Para poder llevar a cabo estas pruebas, será necesario el uso de objetos Mock para la `IConveniosContract.View`, para `ConveniosDao` y para `GasolineraDataBase`. Se aplica prueba de métodos, siendo los casos de prueba definidos para cada método los siguientes:

- Método `init`:

Identificador	Entrada	Valor esperado
HUCC.a	DAO de convenios con datos [[Campsa,20), (Galp,5]]	[[Campsa,20), (Galp,5]] El presentador obtiene la lista y la vista carga los datos de manera correcta.
HUCC.d	Error en la carga de datos, <code>SQLiteException</code> .	Null El presentador recibe la excepción generada por la DAO y la vista alerta del error producido.
HUCC.g	DAO de convenios vacía []	[] La vista ejecuta el método de mostrar la lista de convenios vacía que le muestra un mensaje al usuario indicando que no hay convenios.
HUCC.h	DAO de convenios con datos anómalos [[Campsa,20), (Galp,5), (Repsol, -10), (]]	[[Campsa,20), (Galp,5), (Repsol, -10), (]] El presentador obtiene la lista y la vista carga los datos de manera correcta mostrando los datos anómalos.

- Método `onErrorAceptarClicked`:

Este método es ejecutado cuando a partir de un error al cargar los datos de la DAO. La vista es la encargada de mostrar un error y al pulsar el botón de aceptar, el presentador debe de indicarle a la vista que abra la vista principal.

- Método `onErrorReintentarClicked`

Este método es ejecutado cuando a partir de un error al cargar los datos de la DAO. La vista es la encargada de mostrar un error y al pulsar el botón de reintentar, el presentador debe de indicarle a la vista que refresque la página.

REPORTE

Se implementaron las pruebas unitarias para la clase del presentador de los convenios. Dicha clase implementaba pocos métodos, sobre todo para el método `init` se pudieron realizar las pruebas a partir de los escenarios que fueron posibles. A partir del uso de los mocks se pudo simular una base de datos con datos, anómalos y no anómalos, una base de datos vacía y forzar un error en la base de datos. Las pruebas unitarias pasaron sin mayor dificultad.

En el caso de las pruebas de interfaz, se tuvo limitaciones al poder probar distintos escenarios debido a que las funcionalidades de poder añadir los datos no están disponibles. Por lo cual fue posible probar el escenario en el que no existen datos y se verifica que la lista está vacía y con el botón de atrás de



Android puede volver a la página principal. Se asume el compromiso de implementar las pruebas de interfaz de manera completa en el caso de realizar la historia de usuario correspondiente a añadir los datos.

Para las pruebas de integración, igualmente se probaron los métodos de la clase del presentador. Simulando los escenarios del método `init` considerando una base de datos real y un mock de la vista. Debido al uso de la base de datos real, se probaron dos escenarios, uno en el que no se añadían datos a la base de datos y otro en el que, si se añadían datos, anómalos y no anómalos. Ambas pruebas pasaban de manera independiente, pero no de manera conjunta. Con ayuda del docente de Calidad se pudo identificar que este comportamiento se debía al comportamiento de los hilos al querer acceder a la base de datos. Se intentaron diferentes maneras de lograr el comportamiento esperado y finalmente se implementó una variación al método `getDB` de la clase `GasolineraDatabase`, donde se le manda un nuevo parámetro del tipo booleano para “forzar” el comportamiento de los hilos. Gracias a esto se logró que la clase `ConveniosPresenterITest` pueda ejecutar ambas pruebas que reflejan los escenarios previamente mencionados.

En esta historia de usuario, el plan de pruebas y la implementación de estas, estuvo a cargo de mi persona.

Micaela Camila Patiño Hermosa