

Plan de Pruebas

Historia de Usuario: *Añadir evento a lista*

Se aplicarán los siguientes niveles de prueba:

- **Pruebas de aceptación.** Las realizará el *Product Owner* a mano.
- **Pruebas de interfaz gráfica.** Se realizarán haciendo uso de la librería **Espresso**.
- **Pruebas de integración.** Se realizarán haciendo uso del *framework* **JUnit** y la librería **Mockito**.
- **Pruebas unitarias.** Se realizarán haciendo uso del *framework* **JUnit** y la librería **Mockito**.

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 los criterios de aceptación especificados por el *Product Owner* se identifican los siguientes escenarios:

A1. Añadir evento a lista

- a. Inserción válida (categoría con eventos existentes)
- b. No inserción válida (el evento ya está en la lista)
- c. No inserción válida (no existen listas creadas)
- d. No inserción válida (operación cancelada)

A partir de esto, el cliente realizará las siguientes pruebas, correspondientes a los casos mencionados anteriormente:

Prueba 00: Inserción válida (categoría con eventos existentes)

1. El usuario selecciona la opción 'Añadir evento a lista' de un evento concreto.
2. El sistema muestra un desplegable con los nombres de todas las listas creadas por el usuario.
3. El usuario selecciona la lista en la que quiere añadir el evento.
4. El sistema añade el evento en la lista seleccionada.
5. El sistema muestra un mensaje informando de que el evento se ha añadido a la lista seleccionada, especificando el nombre de esta.

Prueba 01: No inserción válida (el evento ya está en la lista)

1. El usuario selecciona la opción 'Añadir evento a lista' de un evento concreto.
2. El sistema muestra un desplegable con los nombres de todas las listas creadas por el usuario.
3. El usuario selecciona la lista en la que quiere añadir el evento.
4. El sistema detecta que el evento ya está añadido en la lista seleccionada.
5. El sistema muestra un mensaje informando de que el evento ya se encuentra añadido en la lista seleccionada, especificando el nombre de esta.
6. El usuario confirma haber leído el mensaje.
7. El sistema cierra el mensaje.

Prueba 02: No inserción válida (no existen listas creadas)

1. El usuario selecciona la opción 'Añadir evento a lista' de un evento concreto.
2. El sistema detecta que no hay ninguna lista creada por el usuario en la que se pueda añadir el evento.
3. El sistema muestra un desplegable vacío, sin opciones seleccionables.
4. El usuario selecciona otro elemento de la interfaz, fuera del desplegable.
5. El sistema cierra el desplegable, cancelando la operación.

PRUEBAS DE INTERFAZ GRÁFICA

En este caso, se programarán utilizando **Espresso** los escenarios definidos en el apartado anterior. La nomenclatura A1.x de las pruebas de aceptación se corresponde con UIT1.x en las pruebas de interfaz gráfica.

Los casos de prueba definidos para este escenario, suponiendo que el estado inicial del sistema es el reflejado en el archivo [agenda_cultural.json](#), son los que se muestran en la tabla siguiente:

Identificador	Selección	Resultado
UIT1.a	Evento seleccionado: "Abierto el plazo de inscripción para el Concurso Internacional de Piano de Santander Paloma O'Shea" Lista seleccionada: "Lista1" Estado de la lista: Vacía	"Se ha añadido un evento a la lista Lista1"
UIT1.b	Evento seleccionado: "Abierto el plazo de inscripción para el Concurso Internacional de Piano de Santander Paloma O'Shea" Lista seleccionada: "Lista1" Estado de la lista: "Abierto el plazo de inscripción para el Concurso Internacional de Piano de Santander Paloma O'Shea"	"El evento seleccionado ya está en la lista"
UIT1.c	Evento seleccionado: "Abierto el plazo de inscripción para el Concurso Internacional	El evento no se añade a ninguna lista.

	de Piano de Santander Paloma O'Shea"	
	Lista seleccionada: No existen listas creadas	

A modo de simplificación, se ha colocado también en la columna "Selección" el estado de la lista a la que se quiere añadir el evento en el momento de realizar la prueba. Además, para el propósito de este plan de pruebas, suponemos que la lista "Lista1" está ya creada y en la situación descrita en cada caso.

PRUEBAS DE INTEGRACIÓN

En este caso probamos el funcionamiento del método `onAddEventClicked()` de la clase `EventsPresenter` (definido en la interfaz `IEventsContract.Presenter`), ya que internamente, éste llama al método `addEvent()` de la clase `GestionarListasUsuario` (definido en la interfaz `IGestionarListasUsuario`), que gestiona la inserción del evento en la lista.

Comprobamos su funcionamiento según el valor que retorna y mirando el contenido de la lista, verificando si los eventos se han añadido o no.

IT.1a: El evento se añade correctamente en la lista.

IT.1b: El evento ya se encuentra en la lista, por lo que no se añade.

IT.1c: El evento se añade correctamente en la lista, que ya tiene eventos guardados.

IT.1d: No existe ninguna lista con el nombre especificado, por lo que el evento no se añade.

IT.1e: No existe ningún evento con el índice indicado.

IT.1f: El índice introducido no es válido (límite inferior de la lista).

IT.1g: Primer índice válido (límite inferior de la lista).

IT.1h: Último índice válido (límite superior de la lista).

IT.1i: El índice introducido no es válido (límite superior de la lista).

- Método `onAddEventClicked(eventIndex : int, sharedPref : IGestionarListasUsuario, listaEscogida : String) : boolean`

Identificador	Entrada	Estado inicial de la lista	Estado final de la lista	Resultado
IT.1a	(1, sharedPref, "Lista1")	"	"1,"	true
IT.1b	(1, sharedPref, "Lista1")	"1,"	"1,"	false
IT.1c	(2, sharedPref, "Lista1")	"1,"	"1,2"	true
IT.1d	(1, sharedPref, "Lista2")	No existe	No existe	false
IT.1e	(999999, sharedPref, "Lista1")	"1,2,"	"1,2,"	false
IT.1f	(-1, sharedPref, "Lista1")	"1,2,"	"1,2,"	false

IT.1g	(0, sharedPref, "Lista1")	"1,2,"	"1,2,0,"	true
IT.1h	(cachedEvents.size()-1, sharedPref, "Lista1")	"1,2,0,"	"1,2,0,<cachedEvents.size()-1>,"	true
IT.1i	(cachedEvents.size(), sharedPref, "Lista1")	"1,2,"	"1,2,"	false

En la columna "Estado inicial de la lista" se ha indicado si la lista en la que se está intentando añadir un evento existe o no, y el contenido de ésta si lo hace (será un string con los identificadores de sus eventos, separados por comas).

En la columna "Estado final de la lista" se muestra cómo debe quedar tras la operación.

Por simplicidad, se ha supuesto que en el momento de ejecutar el caso "IT.1e" no existe ningún evento con el índice "999999".

En el caso IT.1h, "<cachedEvents.size()-1>" será el índice del último evento de la lista de eventos principal, *cachedEvents*; por lo que dependerá del estado de esta en el momento de ejecutar la prueba. Por ejemplo, si la lista con la que se hacen las pruebas tiene 4 eventos, "<cachedEvents.size()-1>" será "3".

PRUEBAS UNITARIAS

En este caso probamos el funcionamiento del método `onAddEventClicked()` de la clase `EventsPresenter` (definido en la interfaz `IEventsContract.Presenter`), y el método `addEvent()` de la clase `GestionarListasUsuario` (definido en la interfaz `IGestionarListasUsuario`).

- Método `onAddEventClicked(eventIndex : int, sharedPref : IGestionarListasUsuario, listaEscogida : String) : boolean`

Identificador	Entrada	Estado de la lista	Resultado
UT.1a	(1, sharedPref, "Lista1")	""	true, Llamada al método <code>IGestionarListasUsuario.addEvent(1, cachedEvents, "Lista1")</code>
UT.1b	(1, sharedPref, "Lista1")	"1,"	false, Llamada al método <code>IGestionarListasUsuario.addEvent(1, cachedEvents, "Lista1")</code>
UT.1c	(2, sharedPref, "Lista1")	"1,"	true, Llamada al método <code>IGestionarListasUsuario.addEvent(2, cachedEvents, "Lista1")</code>
UT.1d	(1, sharedPref, "Lista2")	No existe	false, No llama a <code>IGestionarListasUsuario.addEvent</code>
UT.1e	(999999, sharedPref,	"1,2,"	false,

	"Lista1")		No llama a IGestionarListasUsuario.addEvent
UT.1f	(-1, sharedPref, "Lista1")	"1,2,"	false, No llama a IGestionarListasUsuario.addEvent
UT.1g	(0, sharedPref, "Lista1")	"1,2,"	true, Llamada al método IGestionarListasUsuario.addEvent (0, cachedEvents, "Lista1")
UT.1h	(cachedEvents.size()-1, sharedPref, "Lista1")	"1,2,"	true, Llamada al método IGestionarListasUsuario.addEvent (cachedEvents.size()-1, cachedEvents, "Lista1")
UT.1i	(cachedEvents.size(), sharedPref, "Lista1")	"1,2,"	false, No llama a IGestionarListasUsuario.addEvent

En la columna "Estado de la lista" se ha indicado si la lista en la que se está intentando añadir un evento existe o no, y el contenido de ésta si lo hace (será un *string* con los identificadores de sus eventos, separados por comas).

En la columna "Resultado" se especifica el valor que retornará el método (*true* o *false*), y si realiza alguna llamada a otro de la clase IGestionarListasUsuario, y con qué valores.

Por simplicidad, se ha supuesto que en el momento de ejecutar el caso "UT.1e" no existe ningún evento con el índice "999999".

El objeto `cachedEvents` es la lista de eventos guardada en la clase `EventsPresenter`.

- **Método** `addEvent(eventIndex : int, cachedEvents : Event[*], listaEscogida : String): boolean`

Identificador	Entrada	Estado de la lista	Resultado
UT.2a	(1, cachedEvents, "Lista1")	"	true, Llamada al método SharedPreferences.Editor.putString("Lista1", "1,")
UT.2b	(1, cachedEvents, "Lista1")	"1,"	false, No llama a SharedPreferences.Editor.putString()
UT.2c	(2, cachedEvents, "Lista1")	"1,"	true, Llamada al método SharedPreferences.Editor.putString("Lista1", "2,")

UT.2d	(0, cachedEvents, "Lista1")	"1,2,"	true, Llamada al método SharedPreferences.Editor.putString("Lista1", "0,")
UT.2e	(cachedEvents.size()-1, cachedEvents, "Lista1")	"1,2,0,,"	true, Llamada al método SharedPreferences.Editor.putString("Lista1", "<cachedEvents.size()-1>,")

En la columna "Resultado" se especifica el valor que retornará el método (*true* o *false*), y si realiza alguna llamada a otro de la clase `SharedPreferences.Editor`, y con qué valores. En el caso UT.2e, "<cachedEvents.size()-1>" será el índice del último evento de la lista de eventos principal, *cachedEvents*; por lo que dependerá del estado de esta en el momento de ejecutar la prueba. Por ejemplo, si la lista con la que se hacen las pruebas tiene 4 eventos, "<cachedEvents.size()-1>" será "3".

REPORTE FINAL

Plan de pruebas realizado por: Iván González del Pozo

Pruebas codificadas por:

- Iván González del Pozo (test "UIT1.c").
- Sara Grela Carrera (todos los demás).

Número de fallos encontrados: 0