



## Plan de pruebas – US437797-OrdenarSobreListaFiltrada (Sergio Gallego Álvarez)

Los niveles de prueba que se van a aplicar son los siguientes:

- Pruebas de aceptación: las pruebas de aceptación se definirán siguiendo una estrategia basada en historias de usuario y serán ejecutadas de forma manual.
- Pruebas de integración: estas pruebas verifican la interacción entre clases. Se llevarán a cabo usando “JUnit” y “Robolectric”.
- Pruebas de interfaz: se prueban todos los elementos desde la interfaz, comprobando que la funcionalidad es correcta y que las salidas son las esperadas respecto a las entradas que se han introducido. Para ello, se ha usado el “framework Espresso”, que nos permite automatizar las pruebas.

### PRUEBAS DE ACEPTACIÓN

Se muestran las pruebas en base a los criterios de aceptación acordados con el *Product Owner*.

#### *Prueba 00: Éxito ordenar sobre lista sin filtrar*

1. El usuario selecciona ordenar.
2. El sistema crea una ventana emergente mostrando dos opciones, ascendente y descendentemente.
3. El usuario selecciona de qué manera desea ordenar la lista.
4. El usuario selecciona aplicar.
5. Se cierra la ventana emergente.
6. El sistema muestra la lista de eventos sin filtrar de manera ordenada a criterio del usuario.

#### *Prueba 01: Éxito ordenar sobre lista filtrada*

1. El usuario selecciona ordenar.
2. El sistema crea una ventana emergente mostrando dos opciones, ascendente y descendentemente.
3. El usuario selecciona de qué manera desea ordenar la lista.
4. El usuario selecciona aplicar.
5. Se cierra la ventana emergente.
6. El sistema muestra la lista de eventos filtrada de manera ordenada a criterio del usuario.



*Prueba 02: Éxito filtrar sobre lista ordenada*

1. El usuario selecciona ordenar.
2. El sistema crea una ventana emergente mostrando dos opciones, ascendente y descendente.
3. El usuario selecciona de qué manera desea ordenar la lista.
4. El usuario selecciona aplicar.
5. Se cierra la ventana emergente.
6. El usuario selecciona filtrar.
7. El sistema crea una ventana emergente con los tipos de evento.
8. El usuario selecciona que tipos de evento desea mostrar.
9. El usuario selecciona aplicar.
10. Se cierra la ventana emergente.
11. El sistema muestra la lista de eventos filtrada de manera ordenada a criterio del usuario.

*Prueba 03: Cancelar*

1. El usuario selecciona ordenar.
2. El sistema crea una ventana emergente mostrando dos opciones, ascendente y descendente.
3. El usuario selecciona de qué manera desea ordenar la lista.
4. El usuario selecciona filtrar.
5. Se cierra la ventana emergente.
6. El sistema muestra la lista de eventos previa a la selección del botón ordenar.

## PRUEBAS DE INTEGRACIÓN

Método `onOrdenarCategoriaClicked(int tipoOrdenacion)` de la clase **EventsPresenter.java**

Casos de prueba válidos:

Identificador	Entrada	Resultado
IT. 1A	tipoOrdenacion = 0, eventosEnFiltrosCombinados = lista vacía	La lista "eventosEnFiltrosCombinados" pasa a ser una copia de "cachedEvents" ordenada de manera ascendente.
IT. 1B	tipoOrdenacion = 1, eventosEnFiltrosCombinados = lista vacía	La lista "eventosEnFiltrosCombinados" pasa a ser una copia de "cachedEvents" ordenada de manera descendente.
IT.1C	tipoOrdenacion = 0, eventosEnFiltrosCombinados = lista no vacía	La lista "eventosEnFiltrosCombinados" ordenada de manera ascendente.
IT.1D	tipoOrdenacion = 1, eventosEnFiltrosCombinados = lista no vacía	La lista "eventosEnFiltrosCombinados" pasa a estar ordenada de manera descendente.



Casos de prueba no válidos:

Identificador	Entrada	Resultado
IT. 1E	2 eventosEnFiltrosCombinados = lista vacía	No actualiza nada, se queda todo como estaba anteriormente.

Método onFiltrarClicked(List<String> checkboxSeleccionados) de la clase **EventsPresenter.java**

Casos de prueba válidos:

Identificador	Entrada	Resultado
IT.1A	checkboxSeleccionados = Lista vacía. ordenFiltrado = 2	Se actualiza la lista de eventos "eventosEnFiltrosCombinados" conteniendo la lista original de eventos sin que estén ordenados.
IT.1B	checkboxSeleccionados = Lista con el tipo de evento Música. ordenFiltrado = 0	Se actualiza la lista de eventos "eventosEnFiltrosCombinados" conteniendo los de tipo Música ordenados de manera alfabética ascendente.
IT.1C	checkboxSeleccionados = Lista con todos los tipos de evento existentes. ordenFiltrado = 1	Se actualiza la lista de eventos "eventosEnFiltrosCombinados" con todos los tipos de eventos exceptuando los eventos sin tipo ordenados de manera alfabética descendente.
IT.1D	checkboxSeleccionados = Lista vacía. ordenFiltrado = 0	Se actualiza la lista de eventos "eventosEnFiltrosCombinados" conteniendo la lista original de eventos ordenados de manera alfabética ascendente.

Casos de prueba no válidos:

Identificador	Entrada	Resultado
IT. 1E	Lista que contiene un tipo de evento no existente. ordenFiltrado = 2	No se actualiza la lista de eventos "eventosEnFiltrosCombinados", conteniendo de esta manera la lista original de eventos.
IT. 1F	Lista con elemento null. ordenFiltrado = 2	NullPointerException
IT. 1G	checkboxSeleccionados = Lista con el tipo de evento Música. ordenFiltrado = 3	Se actualiza la lista de eventos "eventosEnFiltrosCombinados" conteniendo los de tipo Música pero no se ordena de ninguna manera.



\* Nota: el atributo ordenFiltrado: puede tener 3 valores válidos, 0 significa que la lista está ordenada alfabéticamente ascendente, 1 significa que la lista está ordenada alfabéticamente descendente, 2 significa que la lista no está ordenada (por ejemplo: valor que tomaría al arrancar la aplicación). Cualquier otro valor no sería válido.

## PRUEBAS DE INTERFAZ DE USUARIO

Se realizarán seis pruebas:

### A. Ordenación ascendentemente sobre lista sin filtrar

1. Se abrirá la interfaz de ordenar pulsando el botón “Ordenar”.
2. Se comprobará que no hay ningún tipo de ordenación seleccionado.
3. Se seleccionará el botón de ordenación ascendente.
4. El usuario pulsará el botón “Aplicar” de la ventana flotante.
5. Se comprobará que la lista de eventos está ordenada alfabéticamente (A-Z) por tipo.

### B. Ordenación descendentemente sobre lista sin filtrar

1. Se abrirá la interfaz de ordenar pulsando el botón “Ordenar”.
2. Se comprobará que no hay ningún tipo de ordenación seleccionado.
3. Se seleccionará el botón de ordenación descendente.
4. El usuario pulsará el botón “Aplicar” de la ventana flotante.
5. Se comprobará que la lista de eventos está ordenada alfabéticamente de manera inversa (Z-A) por tipo.

### C. Ordenación ascendentemente sobre lista filtrada

1. Se abrirá la interfaz de filtrar pulsando el botón “Filtrar”.
2. Se seleccionarán dos tipos de evento.
3. El usuario pulsará el botón “Aplicar” de la ventana flotante.
4. Se abrirá la interfaz de ordenar pulsando el botón “Ordenar”.
5. Se seleccionará el botón de ordenación ascendente.
6. El usuario pulsará el botón “Aplicar” de la ventana flotante.
7. Se comprobará que la lista de eventos está ordenada alfabéticamente ascendente (A-Z) por tipo tiene el mismo tamaño que tenía anteriormente.



#### D. Ordenación descendentemente sobre lista filtrada

1. Se abrirá la interfaz de filtrar pulsando el botón “Filtrar”.
2. Se seleccionarán dos tipos de evento.
3. El usuario pulsará el botón “Aplicar” de la ventana flotante.
4. Se abrirá la interfaz pulsando el botón “Ordenar”.
5. Se seleccionará el botón de ordenación descendente.
6. El usuario pulsará el botón “Aplicar” de la ventana flotante.
7. Se comprobará que la lista de eventos está ordenada alfabéticamente descendentemente (Z-A) por tipo tiene el mismo tamaño que tenía anteriormente.

#### E. Filtración sobre lista ordenada ascendentemente

1. Se abrirá la interfaz de ordenar pulsando el botón “Ordenar”.
2. Se seleccionará el orden ascendentemente.
3. El usuario pulsará el botón “Aplicar” de la ventana flotante.
4. Se comprobará que la lista de eventos está ordenada alfabéticamente.
5. Se abrirá la interfaz de filtrar pulsando el botón “Filtrar”.
6. Se comprobará que no hay ningún tipo de ordenación seleccionado.
7. Se seleccionarán dos tipos de evento.
8. El usuario pulsará el botón “Aplicar” de la ventana flotante.
9. Se comprobará que la lista de eventos está filtrada y que está ordenada alfabéticamente.

#### F. Filtración sobre lista ordenada descendentemente

1. Se abrirá la interfaz de ordenar pulsando el botón “Ordenar”.
2. Se seleccionará el orden descendentemente.
3. El usuario pulsará el botón “Aplicar” de la ventana flotante.
4. Se comprobará que la lista de eventos está ordenada alfabéticamente de manera inversa.
5. Se abrirá la interfaz de filtrar pulsando el botón “Filtrar”.
6. Se comprobará que no hay ningún tipo de ordenación seleccionado.
7. Se seleccionarán dos tipos de evento.
8. El usuario pulsará el botón “Aplicar” de la ventana flotante.
9. Se comprobará que la lista de eventos está filtrada y que está ordenada alfabéticamente de manera inversa.



## Informe de pruebas – US437797-OrdenarSobreListaFiltrada (Adrián García Cubas)

### PRUEBAS DE INTERFAZ:

- He implementado los casos C,D,E,F.
- aquí tuve problemas para comprobar el tamaño de la lista de eventos principal, pero me lo resolvió un docente.

### -PRUEBAS DE INTEGRACION

- He implementado todos los casos de “onOrdenarCategoriaClicked”.
- He implementado los casos A, B, C, D, E de “onFiltrarClicked”.