

Plan de Pruebas

Historia de Usuario: *Filtrar por tipo de evento*

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. Filtrar por categoría

- a. Filtrado válido (categoría con eventos existentes)
- b. Filtrado válido (categoría sin eventos existentes)
- c. Filtrado válido (categoría no seleccionada)

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

Prueba 00: Filtrado válido (categoría con eventos existentes)

1. El usuario selecciona la opción filtrar por categoría.
2. El usuario selecciona al menos una de las categorías disponibles.
3. El usuario selecciona la opción para aplicar el filtro.
4. El sistema carga los eventos correspondientes con esa(s) categoría(s), que se han traído anteriormente del servicio de datos.
5. El sistema muestra una lista con los eventos pertenecientes a la(s) categoría(s) seleccionada(s).

Prueba 01: Filtrado válido (categoría sin eventos existentes)

1. El usuario selecciona la opción filtrar por categoría.
2. El usuario selecciona una de las categorías disponibles.
3. El usuario selecciona la opción para aplicar el filtro.
4. El sistema intenta cargar los eventos correspondientes con esa categoría, que se han traído anteriormente del servicio de datos.
5. El sistema detecta que no hay ningún evento con la categoría seleccionada.
6. El sistema muestra un mensaje, notificando al usuario de que no existe ningún evento de la categoría seleccionada, en lugar de la lista de eventos.

Prueba 02: Filtrado válido (categoría no seleccionada)

1. El usuario selecciona la opción filtrar por categoría.
2. El usuario no selecciona ninguna categoría.
3. El usuario selecciona la opción para aplicar el filtro.

4. El sistema carga los eventos que no tienen una categoría asignada, que se han traído anteriormente del servicio de datos.
5. El sistema muestra una lista con los eventos sin categoría definida.

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 1.

Tabla 1. Casos de prueba de interfaz gráfica

Identificador	Selección	Resultado
UIT1.a	Categoría: Cultura científica	"En busca de vida en Marte: nuevas misiones y nuevos retos, Cultura científica, Miércoles 08/09/2021, a las 19:00h." "Retratos de buques, exposición temporal, Cultura científica, Sábado 31/07/2021, de 10:00 a 19:30h." "Museo Marítimo del Cantábrico, Cultura científica, Sábado 31/07/2021, de 10:00 a 19:30h." "Vida y Muerte en Cantabria, exposición temporal, Cultura científica, Sábado 31/07/2021, de 10:00 a 14:00h." "Concurso infantil de dibujo de vida extraterrestre, Cultura científica, Sábado 31/07/2021, todo el día."
UIT1.b	Categoría: Música	"No se han encontrado eventos con la(s) categoría(s) seleccionada(s)."
UIT1.c	Categoría: -	<(título, categoría, fecha) por cada evento sin categoría de agenda_cultural.json >

A modo de simplificación, se ha colocado en la columna "Resultado" el valor de cada uno de los campos que tienen que aparecer en cada fila, separados por comas (título, categoría, fecha). En el caso *UIT1.c*, deberán aparecer los campos mencionados anteriormente para cada elemento de la lista de eventos que no tuviera categoría.

PRUEBAS DE INTEGRACIÓN

En este caso probamos el funcionamiento del método `loadData` de la clase `EventsPresenter`, ya que internamente, éste llama al método `getEvents` de la clase `EventsRepository`. El método es *void* y no tiene ningún argumento de entrada, por lo que la manera de comprobar su funcionamiento es mirando el contenido de la lista `cachedEvents` de la clase `EventsPresenter`. Si no se produce ningún error se almacenan en la lista los eventos correspondientes, sino se almacena un valor nulo.

IT.1a: El método funciona correctamente ya que la lista de eventos se carga en el atributo correspondiente.

IT.1b: El método no funciona correctamente por lo que el atributo no recibe la lista de eventos sino un valor nulo.

- Método `loadData()`

Identificador	Atributo	Valor esperado
IT.1a	<code>cachedEvents</code>	<code>List<Event> listaEventosConOrdenAscendente</code>
IT.1b	<code>cachedEvents</code>	<code>null</code>

PRUEBAS UNITARIAS

Pruebas unitarias de la capa de negocio: en este caso probamos el funcionamiento de métodos de la clase `EventsPresenter`.

- Método `onApplyOptions(Options op)`

Identificador	Entrada	Valor esperado
UT.1a	Cultura científica	<code>List<Event> listaEventosCulturaCientifica</code>
UT.1b	Música	<code>List<Event> listaVacia</code>
UT.1c	-	<code>List<Event> listaEventosCompleta</code>

Para hacer la tabla más legible, en el apartado de “Entrada” aparece únicamente la categoría que se ha seleccionado. Internamente, el objeto `op` tendría los valores concretos que identifican que se ha seleccionado esa categoría en el filtro. El apartado de “Valor esperado” representa cómo tiene que quedar la lista `cachedEvents` tras la ejecución del método.

- Método `onApplyFilter(Map<String, Boolean> categories): Event[]`

Identificador	Entrada	Valor esperado
UT.2a	“Cultura científica”	<code>List<Event> listaEventosCulturaCientifica</code>
UT.2b	“Música”	<code>List<Event> listaVacia</code>
UT.2c	-	<code>List<Event> listaEventosCompleta</code>

Para hacer la tabla más legible, en el apartado de “Entrada” aparecen únicamente las keys que tienen valor 1 en el mapa de categorías. En el caso UT.2c el valor de todas las key es 0 (no se ha seleccionado ninguna).

REPORTE FINAL

Plan de pruebas realizado por: Marta Obregón Ruiz
Pruebas codificadas por: Diego San Román
Número de fallos encontrados: 0