**Plan de pruebas US505813-Eliminar punto de interés.**

Autora: Lucía Sañudo Callejo.

**DISEÑO DE LA NUEVA FUNCIONALIDAD.**

**IPointsContract#View**

* showDeleteMode() – Activa el modo de eliminación.
* showDeleteConfirmationPopup(selectedIP: int) – Muestra el cuadro de diálogo de confirmación del usuario, indicando si realmente desea eliminar el punto.
* showNormalMode() – Desactiva el modo eliminación.

**IPointsContract#Presenter**

* onConfirmDeletionClicked (idSelectedPoint: int) – borra el punto de interés pasado como parámetro y actualiza la lista para que el punto no aparezca.
* onActivateDeleteModeClicked ()- Actualiza el modo de eliminación.
* onCancelDeleteModeClicked ()- Actualiza el modo para desactivar la eliminación.
* onTrashIconClicked(selectedIP : int) – Muestra un cuadro de confirmación del usuario.

**PRUEBAS DE ACEPTACIÓN.**

Las pruebas de aceptación están definidas en la tarjeta correspondiente a la historia de usuario 505813 en ScrumDesk.

**PRUEBAS UI.**

En este caso se aplica la técnica basada en los criterios de aceptación para la definición de las pruebas a realizar. Los casos de prueba definidos serán los mismos que los de las pruebas de aceptación (renombrados como UIT.X) pero automatizados a través de Junit y Expresso.

Se asume que la lista posee algún punto de interés y que los test se ejecutan en orden.

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Entrada** | **Valor esperado** |
| UIT.1 | {Nombre:”Zona Central”,  Color:Rosa, Latitud: 87.1234, Longitud: -34.0987, Radio:10}, CONFIRMAR  Se selecciona un punto existente para eliminarlo, | {Zona Norte, Zona Sur} |
| UIT.2 | {Nombre:”Zona norte”,  Color:Gris, Latitud:40.0637, Longitud: -82.3467, Radio:20}, CANCELAR  Se selecciona un punto existente para eliminarlo. | {Zona Norte, Zona Sur} |
| UIT.3 | {Nombre:”Zona norte”,  Color:Gris, Latitud:40.0637, Longitud: -82.3467, Radio:20},  se pulsa fuera de la ventana. | {Zona Norte, Zona Sur} |
| UIT.4 | Salir del modo eliminación.  No se selecciona ningún punto. | {Zona Norte y Zona Sur} |
| UIT.5 | {Nombre:”Zona norte”,  Color:Gris, Latitud:40.0637, Longitud: -82.3467, Radio:20}, CONFIRMAR  Se selecciona un punto existente para eliminarlo. | El toast indique error en la base de datos. |

**PRUEBAS UNITARIAS/INTEGRACIÓN.**

Se van a realizar pruebas unitarias del PointsPresenter donde se va a mockear la vista del PointsView y InterestPointsDAO.

**Pruebas unitarias del Presenter(IPointsContract):**

Método onConfirmDeletionClicked(idSelectedPoint: int) : void

Método onActivateDeleteModeClicked(): void

Método onCancelDeleteModeClicked(): void

Método onTrashIconClicked(selectedIP : int) : void

Implementación del método onConfirmDeletionClicked (idSelectedPoint: int) : void

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Entrada** | **Valor esperado** |
| UD1.a | idSelectedPoint = {1}  DAO con: puntos\_interes  Punto de interés válido. | Se comprueba con verify del mock de la DAO que elimina el punto.  Se deberá de comprobar que la vista llama a showPoints. |
| UD1.b | idSelectedPoint = {4}  DAO con: puntos\_interes  Punto de interés que no es válido. | Se comprueba con verify del mock de la DAO que no se llama a deleteInterestPointelimina el punto. |
| UD1.c | idSelectedPoint = {2}  DAO con: puntos\_interes | Se comprueba verifiy que llama al delete y este devuelve una sqlLiteException  Se debe comprobar que se llama al view.showDeleteError() |

Como todos los métodos son muy sencillos se va a implementar el método **load(): void**  de la historia de usuario 506401-Crear\_punto\_de\_interés\_por\_coordenadas. Se realiza una prueba de este método debido a la implementación de una nueva funcionalidad asociada.

Implementación del método load() : void

|  |  |  |
| --- | --- | --- |
| **Identificador.** | **Entrada.** | **Valor esperado.** |
| UD2.a | DAO con puntos de interés | Se comprobará que los puntos de la DAO son ordenados y se llama a showPoints(listaDePuntos) |
| UD2.b | DAO sin puntos de interés | Se comprobará que los puntos de la DAO son ordenados y se llama a showPoints(listaVacia) |
| UD2.c | Error en la base de datos | Se llama a showLoadError() porque no hay conexión a base de datos. |

Los puntos que aparecen en la lista son los siguientes:

DAO con puntos interés:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IDENTIFICADOR | NOMBRE | COLOR | LATITUD | LONGUITUD | RADIO |
| 1 | Zona Norte | Gris | 40.0637 | -82.3467 | 20 |
| 2 | Zona Central | Rosa | 87.1234 | -34.0987 | 10 |
| 3 | Zona sur | Marrón | 34.1526 | 12.3456 | 20 |

DAO sin puntos interés:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IDENTIFICADOR | NOMBRE | COLOR | LATITUD | LONGUITUD | RADIO |

**REPORTE FINAL. (Miguel Monje Velarde)**

**Pruebas UI:**

**UIT.1**

En la ejecución de las pruebas se ha detectado que no estaba implementado el toast para avisar al usuario de la eliminación del punto de interés. Una vez corregido el problema la prueba ha pasado satisfactoriamente.

**Pruebas Unitarias:**

**Método onConfirmDeletionClicked (idSelectedPoint: int) : void**

Durante la implementación de las pruebas se ha echado en falta un caso para la opción de que se proporcione un id valido pero al eliminar de error en la base de datos se ha implementado el nuevo caso y también se ha tenido que implementar esta funcionalidad en el código ya que no estaba implementado. Una vez terminado los tests se han pasado satisfactoriamente.

**Método load()**

Una vez implementados los tests han pasado satisfactoriamente sin encontrar problemas aparentes el código.