

## **Plan de pruebas US 506401-Crear puntos de interés por coordenadas**

Autor: Sergio Salas

### **PRUEBAS DE ACEPTACIÓN**

Las pruebas de aceptación están definidas en la tarjeta correspondiente a la historia de usuario 506401 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 Espresso.

Se asume que la lista está inicialmente vacía y que los tests se ejecutan en orden.

<b>Identificador</b>	<b>Entrada</b>	<b>Valor esperado</b>
UIT.1	{Nombre: "Zona norte", Color: GRIS, Latitud: 40.0637, Longitud: - 82.3467, Radio: 20} Se introducen datos correctos.	("Zona norte", AZUL, 40.0637, -82.3467, 20) Se comprueba que esos datos aparecen en el formulario y que el punto de interés se muestra en la lista. Comprobar el toast. (La lista debe tener 1 punto de interés)
UIT.2	{Nombre: "Zona *", Color: AZUL, Latitud: 40.0637, Longitud: -82.3467, Radio: 20}, Se intenta introducir un nombre incorrecto (*).	("Zona", AZUL, 40.0637, - 82.3467, 20) Se comprueba que en el nombre se escribe "Zona" y que el punto de interés se muestra en la lista. Comprobar el toast. (La lista debe tener 2 puntos de interés) El punto añadido debe ser el último de la lista.

UIT.3	{Nombre: "Zona norte", Color: AZUL, Latitud: 40.063723, Longitud: - 82.346723, Radio: 20} Se introducir una longitud y latitud con más de 4 decimales.	("Zona Norte", AZUL, 40.0637, -82.3467, 20) Se comprueba que en los campos latitud y longitud se escribe el valor con 4 decimales y que el punto de interés se muestra en la lista. Comprobar el toast. (La lista debe tener 3 puntos de interés) El punto añadido debe ser el último de la lista.
UIT.4	{Nombre: "Zona norte", Color: AZUL, Latitud: 360.0637, Longitud: - 82.3467, Radio: 20} Se introduce un valor de latitud fuera de rango.	Valor incorrecto en el campo latitud. Se comprueba que no se muestra el punto de interés y que se muestra un mensaje de valor incorrecto. (La lista debe tener 3 puntos de interés)
UIT.5	{Nombre: "Zona norte", Color: AZUL, Latitud: 40.0637, Longitud: - 382.3467, Radio: 20} Se introduce un valor de longitud fuera de rango.	Valor incorrecto en el campo longitud. Se comprueba que no se muestra el punto de interés y que se muestra un mensaje de valor incorrecto. (La lista debe tener 3 puntos de interés)
UIT.6	{Nombre: "Zona norte", Color: AZUL, Latitud: 40.06, Longitud: -82.34, Radio: 20} Se introduce un valor válido de longitud o latitud con menos de 4 decimales.	{"Zona", AZUL, 40.0600, - 82.3400, 20} Se comprueba que en los campos latitud y longitud se rellena el valor con 4 decimales y que el punto de interés se muestra en la lista. Comprobar el toast. (La lista debe tener 4 puntos de interés) El punto añadido debe ser el último de la lista.

UIT.7	{Nombre: "Zona norte", Color: AZUL, Latitud: 40.0637, Longitud: - 82.3467, Radio: 20,45} Se introduce un valor del radio con más de un decimal.	{“Zona”, AZUL, 40.0637, - 82.3467, 20,4} Se comprueba que se muestra el valor del radio solo con un decimal y que el punto de interés se muestra en la lista. Comprobar el toast. (La lista debe tener 5 puntos de interés) El punto añadido debe ser el último de la lista.
UIT.8	{Nombre: "Zona norte", Color: AZUL, Latitud: 40.0637, Longitud: - 82.3467, Radio: -34} Se introduce un valor del radio no válido	Valor incorrecto en el campo Radio. Se comprueba que no se muestra el punto de interés y que se muestra un mensaje de valor incorrecto. (La lista debe tener 5 puntos de interés)
UIT.9	{Nombre: "", Color: AZUL, Latitud: 40.0637, Longitud: -82.3467, Radio: 20} No se introduce nada en el campo de nombre.	Falta por rellenar el campo Nombre. Se comprueba que no se crea el punto de interés y que se muestra un mensaje de campo sin rellenar. (La lista debe tener 5 puntos de interés)
UIT.10	{Nombre: "Zona norte", Color: AZUL, Latitud: 40.0637, Longitud: - 82.3467, Radio: 20} Se introducen valores correctos.	Se pulsa cancelar y se comprueba que no se ha añadido el punto de interés a la lista. (La lista debe tener 5 puntos de interés)
UIT.11	{Nombre: "Zona norte", Color: AZUL, Latitud: 40.0637, Longitud: - 82.3467, Radio: 20} Error en la base de datos	Se comprueba que se muestra un mensaje error en la base de datos.

## PRUEBAS UNITARIAS

Se van a realizar pruebas unitarias del PointsPresenter donde se van a mockear la vista del PointsView , el InterestPointsDAO y la interfaz IInterestPointsDAO.

### Pruebas unitarias del Presenter (IPointsContract)

Método init(View view): void

Método onHomeClicked(): void

Método onCreatePointOfInterestClicked(): void

Método onAcceptNewPointOfInterestClicked(InterestPoint newInterestPoint): void

Se asume que se ejecutan en orden

### **Método onAcceptNewPointOfInterestClicked(InterestPoint point): void**

<b>Identificador</b>	<b>Entrada</b>	<b>Valor esperado</b>
UP.1	Punto interés válido point= {"Punto", AZUL, 40.0637, -82.3467, 20}	point= {"Punto", AZUL, 40.0637, -82.3467, 20} Se comprueba con verify del mock de la DAO que se llama al insert con el objeto.
UP.2	Punto interés válido point= {"Punto 2", GRIS, 41, -3, 60}	point= {"Punto 2", GRIS, 41, - 3, 60} Se comprueba con verify del mock de la DAO que se llama al insert con el objeto. Se comprueba que se ha añadido en la posición correcta (El último de todos).
UP.3	Valor Latitud no válido (Superior) point= {"Punto", AZUL, 100, -82.3467, 20}	latitudInvalidaExcepcion
UP.4	Valor Latitud no válido (Inferior) point= {"Punto", AZUL, -96, -82.3467, 20}	latitudInvalidaExcepcion
UP.5	Valor Longitud no válido (Superior) point= {"Punto", AZUL, 40.0637, 300, 20}	longitudInvalidaExcepcion

UP.6	Valor Longitud no válido (Inferior) point= {"Punto", AZUL, 40.0637, -234, 20}	longitudInvalidaExcepcion
UP.7	Valor radio no válido point= {"Punto", AZUL, 40.0637, -82.3467, 0}	radiInvalidoExcepcion

#### Pruebas unitarias de la clase dominio (InterestPointValidator):

Método estático checkFields(InterestPoint interestPoint): void

Método estático checkLongitude(double longitude) throws LongitudInvalidaException

Método estático checkLatitude(double longitude) throws LatitudInvalidaException

Método estático checkRadius(double longitude) throws RadiInvalidoException

Se probará el siguiente método:

#### Método checkFields(InterestPoint point): void

Identificador	Entrada	Valor esperado
UD.1	point {"Punto", AZUL, 40.0637, -82.3467, 20}	
UD.2	point {"Punto", AZUL, 91, -82.3467, 20}	latitudInvalidaExcepcion
UD.3	point {"Punto", AZUL, -91, -82.3467, 20} (-180,1)	latitudInvalidaExcepcion
UD.4	point {"Punto", AZUL, 40.0637, 182, 20} ()	longitudInvalidaExcepcion
UD.5	point {"Punto", AZUL, 40.0637, -181, 20}	longitudInvalidaExcepcion
UD.6	point {"Punto", AZUL, 40.0637, -82.3467, 0}	radiInvalidoExcepcion
UD.7	point {"Punto", AZUL, 40.0637, -82.3467, -1}	radiInvalidoExcepcion

#### IMPLEMENTACION DE LAS PRUEBAS:

##### - PRUEBA UNITARIA DE LA CLASE DE DOMINIO MÉTODO CHECKFIELDS

Se han programado las pruebas según los datos indicados en el plan de pruebas, en este caso mediante la creación de un punto de Interés al que se le iban modificando los datos con los setters para comprobar el lanzamiento de las excepciones. No ha habido problemas de ningún tipo en la ejecución de estos test.

- **PRUEBA UNITARIA DE LA CLASE PRESENTER MÉTODO  
OnAcceptNewPointOfInterestClicked(InterestPoint point): void**

Se han programado las pruebas según los datos indicados en el plan de pruebas. Se han realizado los mocks indicados para la ejecución de las pruebas, y se han utilizado objetos de la clase de dominio “InterestPoints”.

No ha habido problemas en la ejecución de las pruebas, sin embargo, se ha añadido un caso en el que se valore el radio negative como parámetro. Aunque a nivel de pruebas de UI no se vaya a dar ese caso, debido a que el usuario no va a poder introducir un valor negativo en el campo del radio, se ha decidido probar el caso por completitud de las pruebas. Al ejecutar este caso de pruebas, se ha visto que no lanzaba la excepción, ya que en el método checkRadius, se valoraba únicamente el caso en que el radio introducido fuese 0, no menor o igual que cero. Se ha cambiado esto, y los test han ejecutado sin ningún tipo de problemas.

Además, se han añadido comprobaciones del tamaño de la lista retornada por la DAO una vez se ha añadido un punto de interés, y una vez se ha lanzado una excepción, para comprobar que el tamaño se modifica o permanece tal como debería estar y se comprueban las llamadas a los métodos de la Interfaz de la DAO para obtener los puntos y para añadirlos.

Test ejecutados por Lucía Fernández Mancebo.