



Plan de pruebas #505806-Buscar Por Provincia Y Localidad¹

PRUEBAS DE ACEPTACIÓN

En base a los criterios de aceptación se identifican los siguientes escenarios:

- A1. CA: Caso de éxito: Búsqueda exitosa con provincia y municipio.
1. El usuario pulsa el botón de filtros en la toolbar.
 2. El usuario selecciona la provincia en el menú desplegable e introduce el municipio.
 3. El usuario pulsa el botón “buscar”.
 4. La aplicación muestra una lista con las gasolineras encontradas en el municipio seleccionado.
 5. La aplicación muestra un toast con el número de gasolineras encontradas.
- A2. CA: Caso de éxito: Búsqueda exitosa con provincia sin municipio.
1. El usuario pulsa el botón de filtros en la toolbar.
 2. El usuario selecciona la provincia en el menú desplegable y deja vacía la localidad.
 3. El usuario pulsa el botón “buscar”.
 4. La aplicación muestra una lista con las gasolineras encontradas en la provincia seleccionada.
 5. La aplicación muestra un toast con el número de gasolineras encontradas.
- A3. CA: Caso de éxito: Búsqueda exitosa sin provincia con municipio.
1. El usuario pulsa el botón de filtros en la toolbar.
 2. El usuario no selecciona ninguna provincia e introduce el municipio.
 3. El usuario pulsa el botón de “buscar”.
 4. La aplicación muestra una lista con todas las gasolineras encontradas en el municipio.
 5. La aplicación muestra un toast con el número de gasolineras encontradas.
- A4. CA: Caso de éxito: Búsqueda exitosa sin filtros aplicados.
1. El usuario pulsa el botón de filtros en la toolbar.
 2. El usuario pulsa el botón de “buscar” sin introducir ninguna provincia ni municipio.
 3. La aplicación muestra una lista con todas las gasolineras encontradas.
 4. La aplicación muestra un toast con el número de gasolineras encontradas.
- A5. CA: Datos no válidos. (Difiere con lo definido en scrum)
1. El usuario pulsa el botón de filtros en la toolbar.
 2. El usuario introduce un municipio inexistente o que no pertenece a la provincia seleccionada.
 3. La aplicación muestra una lista vacía.
 4. La aplicación muestra un toast con 0 gasolineras encontradas.
- A6. CA: Resultado con lista vacía
1. El usuario pulsa el botón de filtros en la toolbar.
 2. El usuario selecciona una provincia y municipio donde no hay gasolineras disponibles.
 3. El usuario pulsa el botón “buscar”.
 4. La aplicación no encuentra gasolineras que coincidan con los filtros aplicados.
 5. Se muestra una lista vacía de gasolineras.
 6. La aplicación muestra un toast con 0 gasolineras encontradas.

1. Aunque se llama PorProvinciaYLocalidad nos han recomendado que se busque por municipio por cómo funciona la API.

* No se define el caso de error en acceso a BD de scrum ya que no es necesario en esta historia de usuario.



A7. CA: Ausencia de datos importantes

1. El usuario pulsa el botón de filtros en la toolbar.
2. El usuario selecciona la provincia y/o municipio.
3. El usuario pulsa el botón "buscar".
4. La aplicación encuentra gasolineras, pero algunas tienen datos incompletos (falta la provincia o el municipio).
5. La lista de gasolineras se muestra con solo aquellas que tienen datos completos.

Tabla 2. Casos de prueba de aceptación

Identificador	Entrada	Resultado
A1	("Cantabria", "Santander")	Lista con las gasolineras de Santander
A2	("Cantabria", "")	Lista con las gasolineras de Cantabria
A3	("-", "Santander")	Lista con las gasolineras de Santander
A4	("-", "")	Lista con las gasolineras de España
A5	("Asturias", "Santander")	Lista vacía + mensaje 0 resultados
A6	("Valencia", "Emperador")	Lista vacía + mensaje 0 resultados
A7		Lista sin las gasolineras sin datos

PRUEBAS UNITARIAS

En esta historia de usuario se probarán los métodos añadidos `filtrarPorProvinciaYMunicipio` de la clase `Filtros` y el método `onSearchStationsWithFilters` de la clase `MainPresenter`.

Método `filtrarPorProvinciaYMunicipio`

List<Gasolinera> `filtrarPorProvinciaYMunicipio`(List<Gasolinera> gasolineras, String provincia, String municipio)

Las pruebas para este método se realizarán haciendo uso de mocks. Los mocks de las gasolineras serán los mismo que los indicados en el Anexo 1.

- A. Provincia y municipio válidos.
- B. Provincia válida y municipio vacío.
- C. Provincia vacía y municipio válido.
- D. Provincia válida y municipio vacío sin resultados.
- E. Provincia vacía y municipio válido sin resultados.
- F. Municipio no perteneciente a provincia.
- G. Provincia y municipio válidos sin resultados.
- H. Provincia y municipio vacíos.
- I. Lista de gasolineras vacía.

Tabla 3. Casos de prueba para el método `filtrarPorProvinciaYMunicipio` de la clase `Filtros`

Identificador	Entrada	Valor esperado
UD.1A	("Cantabria", "Santander")	[Cepsa, Repsol, Carrefour]
UD.1B	("Cantabria", null)	[Cepsa, Repsol, Carrefour, Ballenoil, Shell]
UD.1C	(null, "Santander")	[Cepsa, Repsol, Carrefour]
UD.1D	("Madrid", null)	[]



UD.1E	(null, "Tineo")	[]
UD.1F	("Asturias", "Santander")	[]
UD.1G	("Asturias", "Tineo")	[]
UD.1H	(null, null)	[Cepsa, Repsol, Carrefour, Ballenoil, Shell, Petronor, Avia, Cepsa]
UD.1I	-	null

Método onSearchStationsWithFilters

onSearchStationsWithFilters(String provincia, String municipio, Boolean abierto): void

- A. Provincia y municipio válidos.
- B. Provincia válida y municipio vacío.
- C. Provincia vacía y municipio válido.
- D. Provincia válida y municipio vacío sin resultados.
- E. Provincia y municipio válidos sin resultados.
- F. Provincia vacía y municipio válido sin resultados.
- G. Municipio no perteneciente a provincia.
- H. Provincia y municipio vacíos.

Tabla 4. Casos de prueba para el método onSearchStationsWithFilters de la clase MainPresenter

Identificador	Entrada	Valor esperado
UB.1A	("Cantabria", "Santander", True)	Se llama al método <i>filtrarPorProvinciaYMunicipio</i> con finalProvincia = Cantabria y finalMunicipio = Santander. Además, se llama a <i>filtrarPorEstado</i> , <i>showStations</i> y <i>showLoadCorrect</i> (2).
UB.1B	("Cantabria", "", True)	Se llama al método <i>filtrarPorProvinciaYMunicipio</i> con finalProvincia = Cantabria y finalMunicipio = null. Además, se llama a <i>filtrarPorEstado</i> , <i>showStations</i> y <i>showLoadCorrect</i> (4).
UB.1C	("", "Santander", False)	Se llama al método <i>filtrarPorProvinciaYMunicipio</i> con finalProvincia = null y finalMunicipio = Santander. Además, se llama a <i>showStations</i> y <i>showLoadCorrect</i> (2).
UB.1D	("Madrid", "", False)	Se llama al método <i>filtrarPorProvinciaYMunicipio</i> con finalProvincia = Madrid y finalMunicipio = null. Además, se llama a <i>showStations</i> y <i>showLoadCorrect</i> (0).
UB.1E	("Asturias", "Tineo", False)	Se llama al método <i>filtrarPorProvinciaYMunicipio</i> con finalProvincia = Asturias y finalMunicipio = Tineo. Además, se llama a <i>showStations</i> y <i>showLoadCorrect</i> (0).
UB.1F	("", "Tineo", False)	Se llama al método <i>filtrarPorProvinciaYMunicipio</i> con finalProvincia = null y finalMunicipio = Tineo. Además, se llama a <i>showStations</i> y <i>showLoadCorrect</i> (0).



UB.1G	("Asturias", "Santander", False)	Se llama al método <i>filtrarPorProvinciaYMunicipio</i> con <i>finalProvincia</i> = Asturias y <i>finalMunicipio</i> = Santander. Además, se llama a <i>showStations</i> y <i>showLoadCorrect(0)</i> .
UB.1H	("-", "", True)	Se llama a <i>filtrarPorEstado</i> , <i>showStations</i> con la lista de todas la gasolineras y <i>showLoadCorrect(7)</i> .

PRUEBAS DE INTEGRACIÓN

Se probará que el presenter funciona bien en conjunto con la view y la clase creada Filtros. Los resultados obtenidos en las pruebas se filtran en base a un fichero JSON el cuál se encuentra resumido en en Anexo 1. El orden de las pruebas y los casos de prueba a realizar serían los siguientes:

Tabla 5. Casos de prueba Integración entre la Vista, el Presenter y Filtros.

Entrada View	Acción Esperada (Presenter)	Resultado Esperado (View)
("Cantabria", "Santander")	Se llama a <i>Filtros.filtrarPorProvinciaYMunicipio</i> con "Cantabria" y "Santander"	Mostrar lista y toast con 3 gasolineras [Cepsa, Repsol, Carrefour]
("Cantabria", "")	Se llama a <i>Filtros.filtrarPorProvinciaYMunicipio</i> con "Cantabria" y null	Mostrar lista y toast con 5 gasolineras [Cepsa, Repsol, Carrefour, Ballenoil, Shell]
("-", "Santander")	Se llama a <i>Filtros.filtrarPorProvinciaYMunicipio</i> con null y "Santander"	Mostrar lista y toast con 3 gasolineras [Cepsa, Repsol, Carrefour]
("-", "")	Se llama a <i>Filtros.filtrarPorProvinciaYMunicipio</i> con null y null	Mostrar lista y toast con 8 gasolineras [Cepsa, Repsol, Carrefour, Ballenoil, Shell, Petronor, Avia, Cepsa]
("Asturias", "Santander")	Lanzar error municipio	Mostrar lista anterior y toast con municipio incorrecto
("Albacete", "")	Se llama <i>Filtros.filtrarPorProvinciaYMunicipio</i> con "Albacete" y null	Lista vacía y toast sin resultados
("Galicia", "")	Se llama <i>Filtros.filtrarPorProvinciaYMunicipio</i> con "Galicia" y null	Mostrar lista y toast ausencia de datos [Avia]



PRUEBAS DE INTERFAZ

Las pruebas de interfaz coinciden con las pruebas de aceptación, sin embargo, se usarán los datos de las gasolineras del fichero JSON indicado en el Anexo 1.

Tabla 6. Casos de prueba UI.

ID	Entrada	Resultado
UI.1	("Cantabria", "Santander")	[Repsol, Carrefour] + toast 2 gasolineras
UI.2	("Cantabria", "")	[Repsol, Carrefour, Ballenoil, Shell] + toast 4 gasolineras
UI.3	("-", "Santander")	[Repsol, Carrefour] + toast 2 gasolineras
UI.4	("-", "")	[Repsol, Carrefour, Ballenoil, Shell, Petronor, Avia, Cepsa] + toast 7 gasolineras
UI.5	("Asturias", "Santander")	[] + toast 0 gasolineras
UI.6	("Albacete", "")	[] + toast 2 gasolineras
UI.7	("Galicia", "")	[Avia] + toast 1 gasolinera

REPORTE FINAL

A continuación, se describen los resultados de las pruebas realizadas en la historia de usuario asociada a la funcionalidad de filtrado de gasolineras por provincia y municipio en la aplicación. Se realizaron pruebas unitarias y pruebas de interfaz para validar el correcto funcionamiento de los métodos involucrados en esta funcionalidad.

Responsables:

- Plan de Pruebas: Pablo Canales
- Pruebas Unitarias: Pablo Canales
- Pruebas UI: Pablo Canales

Fallos Detectados:

Durante la ejecución de las pruebas no se detectaron fallos significativos. Todas las pruebas pasaron según lo esperado. Sin embargo, se identificaron las siguientes recomendaciones para futuras iteraciones:

Validación de Datos: Asegurarse de que los datos de entrada sean siempre válidos antes de realizar búsquedas para evitar errores en tiempo de ejecución.

Manejo de Errores: Considerar la implementación de mensajes de error más informativos para el usuario en caso de que no se encuentren resultados en la búsqueda.

Conclusiones:

La funcionalidad de buscar gasolineras por provincia y municipio ha sido validada exitosamente a través de pruebas unitarias y pruebas de interfaz. Los resultados indican que la aplicación se comporta como se esperaba, proporcionando al usuario los resultados correctos según los filtros aplicados.



ANEXO 1. FICHERO JSON UTILIZADO PARA LAS PRUEBAS

Para probar el correcto funcionamiento de la implementación desarrollada para esta historia de usuario se utilizarán los datos incluidos en el fichero JSON, los cuales se encuentran resumidos en la Tabla 1.

Tabla 1 Fichero JSON.

GASOLINERAS		
RÓTULO	PROVINCIA	MUNICIPIO
Repsol	Cantabria	Santander
Carrefour	Cantabria	Santander
Ballenoil	Cantabria	Guarnizo
Shell	Cantabria	Maliaño
Petronor	Asturias	Gijón
Avia	Coruña (A)	Carballo
Cepsa		Ferrol

Pablo Canales Cobo