



## PS17 – Plan de Pruebas: Historia de usuario “Ordenar paradas alfabéticamente”

### Resumen

En el presente documento se especifica el plan de pruebas que se desarrollará para comprobar el correcto funcionamiento de la historia de usuario “Ordenar paradas alfabéticamente”.

Document ID:	PS17/00/2017-PP002-US242062-OrdenarParadasAlfabéticamente
Departamento:	Procesos de Ingeniería del Software dentro del proyecto integrado.
Tipo:	PLANIFICACIÓN
Privacidad:	CONFIDENCIAL
Estado:	ENTREGABLE
Versión:	1.0.8
Fecha:	08/11/2017
Autores:	Cerezo Fernández, Elsa
Revisores:	Solar Iglesias, Fernando; Oslé García, Luis; Sainz-Maza Ruiz, Javier; Martínez Vila, Javier



## HISTORIAL DE CAMBIOS

<b>Versión</b>	<b>Fecha</b>	<b>Cambio</b>	<b>Responsable</b>
1.0.0	26/10/2017	Creación del documento y primera redacción.	Cerezo Fernández, Elsa
1.0.1	27/10/2017	Redacción de pruebas de aceptación, integración y unitarias.	Cerezo Fernández, Elsa
1.0.2	30/10/2017	Corrección de pruebas de aceptación, Integración y unitarias.	Cerezo Fernández, Elsa
1.0.3	31/10/2017	Nueva redacción de las pruebas de integración.	Cerezo Fernández, Elsa
1.0.4	31/10/2017	Continuación de redacción de las pruebas de integración.	Cerezo Fernández, Elsa
1.0.5	01/11/2017	Retoca los distintos apartados.	Cerezo Fernández, Elsa
1.0.6	02/11/2017	Correcciones mínimas de redacción.	Cerezo Fernández, Elsa
1.0.7	08/11/2017	Sustituir diagramas y corrección del plan.	Cerezo Fernández, Elsa



## 1. Introducción

El documento contiene la definición del plan de pruebas que se seguirá para probar el buen funcionamiento de la funcionalidad “Ordenar paradas alfabéticamente”, además de probar su integración con la historia de usuario “Listar paradas”.

Se recogen en el informe las pruebas de aceptación, especificadas por el *Product Owner*, el Sr. Medina, las pruebas de integración, cuyo desarrollo se especificará más adelante, y las pruebas unitarias.

## 2. Definición del caso de uso

<b>Identificador:</b>	#242062
<b>Caso de uso:</b>	Ordenar paradas alfabéticamente
<b>Descripción:</b>	El usuario visualizará la lista de paradas ordenada alfabéticamente.
<b>Actor:</b>	Usuario habitual del TUS
<b>Secuencia:</b>	<ol style="list-style-type: none"><li>1. El usuario habitual de TUS pulsa el despliegue de acciones.</li><li>2. El sistema muestra distintos criterios de ordenación a aplicar sobre la lista de paradas.</li><li>3. El usuario habitual de TUS pulsa en la opción “Ordenar alfabéticamente”.</li><li>4. El sistema muestra una lista de paradas ordenadas por orden alfabético.</li></ol>
<b>Extensiones:</b>	<ol style="list-style-type: none"><li>4.a. Si no hay Internet, se ordenan los datos cargados en la última conexión a Internet.</li><li>4.b. Si el sistema no tiene datos, el sistema no mostrará cambios.</li></ol>

## 3. Pruebas de aceptación

En el presente apartado se indicarán las pruebas de aceptación que se llevarán a cabo para comprobar la correcta funcionalidad de las implementaciones realizadas:

### PA1: Se prueba ordenar alfabéticamente tras acceder a la aplicación con acceso a Internet

1. El usuario pulsa en el despliegue de acciones.
2. El usuario selecciona la opción “Ordenar alfabéticamente”.
3. El sistema lista las paradas en orden alfabético.

El resultado esperado es poder visualizar un listado de paradas ordenadas alfabéticamente y que los datos coincidan con los sustraídos del JSON.

### PA2: Se prueba ordenar alfabéticamente sin acceso a Internet, habiendo accedido a la aplicación con acceso a Internet

1. El usuario pulsa en el despliegue de acciones.
2. El usuario selecciona la opción “Ordenar alfabéticamente”.
3. El sistema lista las paradas en orden alfabético.

El resultado esperado es visualizar un listado de paradas ordenadas alfabéticamente.

**PA3:** Se prueba ordenar alfabéticamente tras acceder a la aplicación, por primera vez, sin acceso a Internet

1. El usuario pulsa en el despliegue de acciones.
2. El usuario selecciona la opción "Ordenar alfabéticamente".
3. El sistema no muestra cambios.

El resultado esperado es que el sistema no responda ante la petición de ordenar paradas alfabéticamente, ya que no hay datos que ordenar.

#### 4. Pruebas de integración

Las pruebas de integración se realizarán mediante una estrategia incremental guiada por la funcionalidad implementada.

A continuación, se muestra el diseño en el que se basa la arquitectura de nuestro producto, el cual es un modelo MVP (*Model View Presenter*):

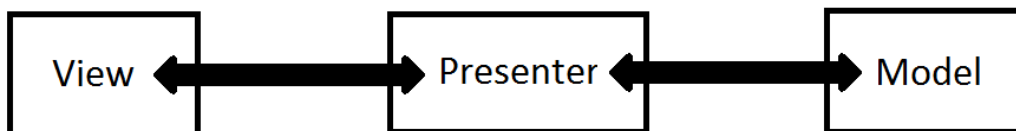


Figura 1. Diseño de arquitectura global

En la capa *Model*, se recogen aquellos fragmentos de la aplicación dedicados a la captura de datos. La capa *Presenter* se dedica al tratamiento de los datos obtenidos, preparándolos para ser mostrados en la capa *View*.

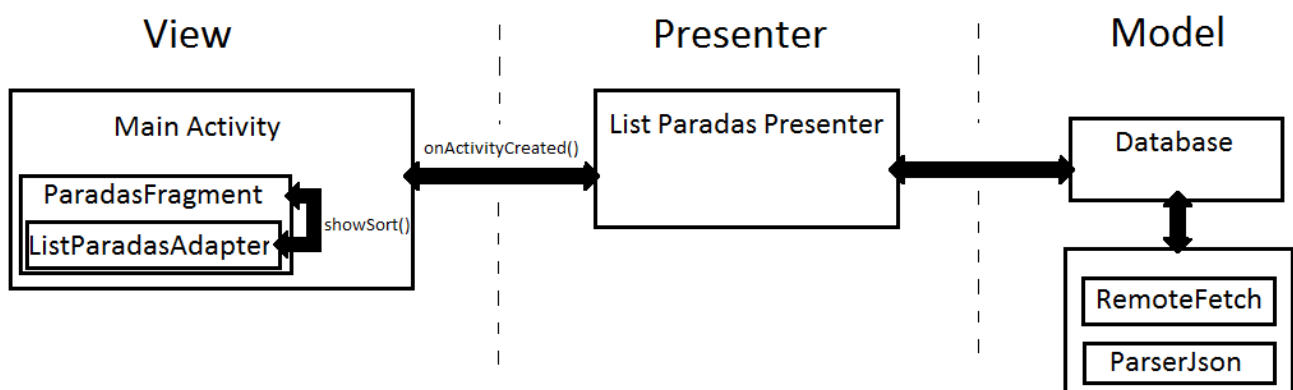


Figura 2. Diseño de arquitectura de la historia de usuario "Ordenar paradas alfabéticamente"

En el anterior diagrama (Figura 2) se puede observar los distintos fragmentos de la aplicación implementada a nivel arquitectónico. El método que da funcionalidad a la historia de usuario "Ordenar paradas alfabéticamente" es *showSort()* el cuál se utiliza en la capa *View*, ya que sirve para modificar la forma en la que se muestran los datos.

En este apartado se diseñan las pruebas de integración, encargadas de comprobar si las funcionalidades implementadas funcionan correctamente al integrarse. Por ello, al diagrama anterior (*Figura 2*) se le añade la funcionalidad de “*Listar paradas*” implementada por el método *show()* dentro de la capa *View*. Este método hace que el *ParadasFragment* muestre una lista en un orden por defecto. En cambio, al usar el *showSort()* el listado a mostrar se ordenaría alfabéticamente.

A partir del diseño planteado, y teniendo en cuenta las operaciones que se implican, se han diseñado los dos casos de prueba que se indican a continuación:

**PI1.** Probar que se devuelve un listado de paradas ordenado alfabéticamente, tras obtener un listado de paradas gracias a la funcionalidad “*Listar paradas*”, la cual extraerá los datos de un *Json* local.

*Nota:* los casos se realizarán obteniendo los datos de un *Json* local, dado que los ficheros del repositorio remoto son actualizados, por lo tanto, los datos de estos archivos pueden cambiar invalidando las pruebas realizadas.

## 5. Pruebas unitarias

En este apartado se especificarán los casos de prueba que se tendrán en cuenta para la implementación de las pruebas unitarias.

El caso de uso será probado con la siguiente lista de paradas:

“Camarreal Penacastillo”, “499”, 42063  
“Ortega y Gasset.28”, “500”, 42064  
“Avenida de Cantabria.35”, “505”, 50693  
“Nuevo Parque”, “307”, 100

- **Casos de prueba**

**PU1.** Al facilitar la anterior lista de paradas, tras realizarse el caso de uso “Ordenar paradas alfabéticamente” con acceso a internet, el resultado esperado es obtener una lista en el siguiente orden:

“Avenida de Cantabria.35”, “505”, 50693  
“Camarreal Penacastillo”, “499”, 42063  
“Nuevo Parque”, “307”, 100  
“Ortega y Gasset.28”, “500”, 42064

**PU2.** Al facilitar la lista de paradas vacía, tras realizarse el caso de uso “Ordenar paradas alfabéticamente”, el resultado esperado es obtener un listado vacío.

## 6. Sumario

En el documento se ha expuesto una definición del plan de pruebas que seguirá la empresa *Titan Solutions* para probar los requisitos especificados por el *Product Owner* para validar su buen comportamiento.