



## PS17 – Plan de pruebas: Historia de usuario “Listar paradas”

### Resumen

En el presente plan se muestra el diseño de pruebas unitarias, integración y aceptación llevado a cabo para la historia de usuario “Listar paradas”, de manera que tras su posterior implementación se pueda detectar la presencia de errores en el código desarrollado.

Document ID:	PS17/00/2017-PP002-US243482-ListarParadas
Departamento:	Procesos de Ingeniería de Software, dentro del proyecto integrado
Tipo:	PLANIFICACIÓN
Privacidad:	CONFIDENCIAL
Estado:	ENTREGABLE
Versión:	1.0.5
Fecha:	08/11/2017
Autores:	Solar Iglesias, Fernando
Revisores:	Cerezo Fernández, Elsa; Martínez Vila, Javier; Oslé García, Luis; Sainz-Maza Ruiz, 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.	Solar Iglesias, Fernando
1.0.1	26/10/2017	Descrito el caso de uso a probar.	Solar Iglesias, Fernando
1.0.2	29/10/2017	Incorporadas las pruebas de aceptación tratadas con el Product Owner.	Solar Iglesias, Fernando
1.0.3	30/10/2017	Añadidas las pruebas unitarias para "Listar paradas".	Solar Iglesias, Fernando
1.0.4	01/11/2017	Añadidas las pruebas de integración.	Solar Iglesias, Fernando
1.0.5	08/11/2017	Corregir apartado de pruebas de integración.	Cerezo Fernández, Elsa



## 1. Introducción

El presente documento recoge el conjunto de pruebas unitarias, de integración y de aceptación con la finalidad de que puedan ser usadas posteriormente para ser implementadas, comprobando así la presencia de errores en el código desarrollado por el equipo.

En primer lugar, se describe el caso de uso correspondiente a la historia de usuario objeto de pruebas: #243482-Listar paradas, para posteriormente definir cada una de las pruebas mencionadas.

## 2. Caso de uso: Listar paradas

<b>Identificador:</b>	#243482
<b>Caso de uso:</b>	Listar paradas
<b>Descripción:</b>	El usuario visualiza el listado de todas las paradas correspondientes a todas las líneas del transporte urbano de Santander (TUS). Por defecto se visualizan según el orden dado por el proveedor de la información.
<b>Actores:</b>	Usuario habitual del TUS.
<b>Secuencia:</b>	<ol style="list-style-type: none"><li>1. El usuario arranca la aplicación y visualiza el logo de la empresa desarrolladora.</li><li>2. El sistema obtiene remotamente los detalles de las paradas y las procesa.</li><li>3. El usuario espera un tiempo determinado o pulsa sobre la pantalla para continuar.</li><li>4. El sistema cambia de vista mostrando las paradas recogidas tras su procesamiento.</li><li>5. El usuario visualiza el listado de paradas pudiendo desplazarse por éste.</li></ol>
<b>Extensiones:</b>	2.a. Si no se pueden obtener los datos, el sistema informa al usuario de la situación de error.

## 3. Pruebas de aceptación

De manera automatizada y utilizando las herramientas oportunas del entorno de desarrollo, comprobar gráficamente a través de la interfaz de usuario que las siguientes interacciones muestran los resultados deseados.

### PA1: Comprobar mostrado de listado de paradas. (Con acceso a Internet)

1. El usuario pulsa sobre el icono de la app en el dispositivo móvil.
2. El usuario espera a la ejecución de la aplicación y carga de datos.
3. El sistema lista las paradas en el orden por defecto (según el proveedor de la información).

El resultado esperado es poder visualizar un listado de paradas en el orden proporcionado por el proveedor de la información, en este caso el Ayuntamiento de Santander.

**PA2: Comprobar mensaje de error al listar paradas. (Sin acceso a Internet):**

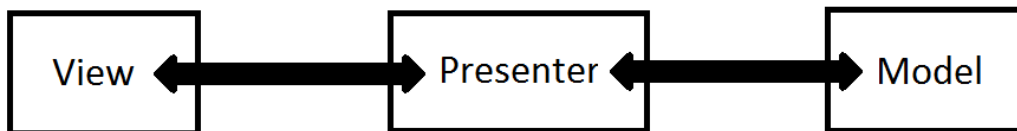
1. El usuario pulsa sobre el icono de la app en el dispositivo móvil.
2. El usuario espera a la ejecución de la aplicación y carga de datos.
3. El sistema notifica que no hay conexión a Internet.

El resultado esperado es poder visualizar un mensaje de error que notifique de la imposibilidad de recuperar el listado de paradas por falta de conexión a la red.

#### 4. Pruebas de integración

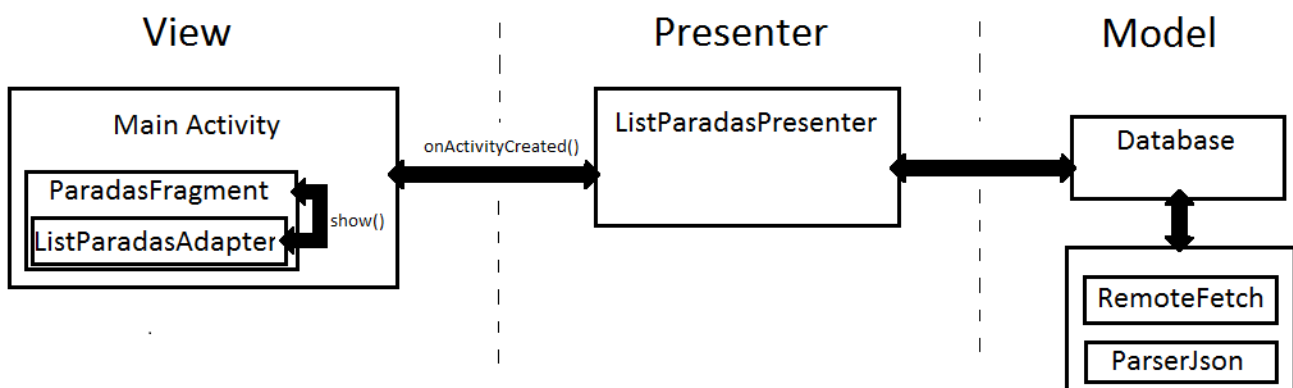
Las pruebas de integración se realizarán mediante una estrategia incremental guiada por la funcionalidad implementada.

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 "Listar paradas"*

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 "Listar paradas" es *show()* 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 “Ordenar paradas alfabéticamente” implementada por el método *showSort()* dentro de la capa *View*. Este método hace que el *ParadasFragment* muestre una lista en orden alfabético. En cambio, al usar el *show()* el listado a mostrar se muestra con un orden por defecto.

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

A continuación, se muestran los casos correspondientes a las pruebas unitarias. Para llevarlas a cabo, se toma como base el siguiente *JSON* que simula al original de datos proporcionado por el Ayuntamiento de Santander:

```
{ "summary": { "items": 4, "items_per_page": 1000, "pages": 1, "current_page": 1 }, "resources": [
  {
    "wgs84_pos:long": "-3.866588480444409",
    "ayto:numero": "499",
    "gn:coordY": "4811045.72",
    "gn:coordX": "429986.36",
    "ayto:sentido": "Centro",
    "vivo:address1": "Avda Severo Ochoa",
    "dc:modified": "2017-10-29T23:20:03.158Z",
    "wgs84_pos:lat": "43.44716242117678",
    "ayto:parada": "Camarreal Penacastillo",
    "dc:identifier": "42063",
    "uri": "http://datos.santander.es/api/datos/paradas_bus/42063.json"
  },
  {
    "wgs84_pos:long": "-3.8571753633683796",
    "ayto:numero": "500",
    "gn:coordY": "4810658.9",
    "gn:coordX": "430744.13",
    "ayto:sentido": "Centro",
    "vivo:address1": "Ortega y Gasset.28",
    "dc:modified": "2017-10-29T23:20:03.158Z",
    "wgs84_pos:lat": "43.44375026592359",
    "ayto:parada": "Ortega y Gasset.28",
    "dc:identifier": "42064",
    "uri": "http://datos.santander.es/api/datos/paradas_bus/42064.json"
  },
  {
    "wgs84_pos:long": "-3.8062240806383896",
    "ayto:numero": "505",
    "gn:coordY": "4814192.02",
    "gn:coordX": "434901.67",
    "ayto:sentido": "Puerto Chico",
    "vivo:address1": "Avenida de Cantabria nº 35",
    "dc:modified": "2017-10-29T23:20:03.158Z",
```

```

    "wgs84_pos:lat": "43.47593386282703",
    "ayto:parada": "Avenida de Cantabria.35",
    "dc:identifier": "50693",
    "uri": "http://datos.santander.es/api/datos/paradas_bus/50693.json"},
  {
    "wgs84_pos:long": "-3.8483834036096156",
    "ayto:numero": "307",
    "gn:coordY": "4810890",
    "gn:coordX": "431458",
    "ayto:sentido": "Canalejas",
    "vivo:address1": "Francisco Rivas Moreno",
    "dc:modified": "2017-10-29T23:20:03.123Z",
    "wgs84_pos:lat": "43.445896683845085",
    "ayto:parada": "Nuevo Parque",
    "dc:identifier": "100",
    "uri": "http://datos.santander.es/api/datos/paradas_bus/100.json"]}
}

```

- **Casos de prueba**

**PU1.** Se deberá probar el método “obtenParadas” de la clase “ListLineasPresenter”, de manera que al proporcionarle el JSON de entrada mostrado anteriormente, el método “getListaLineasBus” retorne un listado de las siguientes paradas.

Nombre	Identificador	Número
Camarreal Penacastillo	42063	499
Ortega y Gasset.28	42064	500
Avenida de Cantabria.35	50693	505
Nuevo Parque	100	307

**PU2.** Se deberá probar el método *obtenParadas()* de la clase *ListLineasPresenter*, de manera que, al proporcionarle un JSON de entrada vacío, el método *getListaLineasBus()* retorne un listado vacío de paradas. Esto simula el resultado del caso en el que no sea posible obtener las paradas.

## 6. Sumario

A lo largo de los puntos anteriores, se ha mostrado tanto la definición del caso de uso como el diseño completo del plan de pruebas unitarias, de integración y de aceptación, a partir del cual puede ser empleado para desarrollar la implementación de éstas.