



## PS17 – Plan de pruebas: Historia de usuario “Consultar listado de líneas”

### Resumen

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

Document ID:	PS17/00/2017-PP004-US241953-ConsultarListadoLíneas
Departamento:	Procesos de Ingeniería de Software, dentro del proyecto integrado
Tipo:	PLANIFICACIÓN
Privacidad:	CONFIDENCIAL
Estado:	ENTREGABLE
Versión:	1.0.3
Fecha:	14/11/2017
Autores:	Martínez Vila, Javier
Revisores:	Solar Iglesias, Fernando; Cerezo Fernández, Elsa; 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	06/11/2017	Creación del documento.	Martínez Vila, Javier
1.0.1	09/11/2017	Primera redacción.	Martínez Vila, Javier
1.0.2	12/11/2017	Redacción pruebas de integración y unitarias.	Martínez Vila, Javier
1.0.3	14/11/2017	Revisar y corregir documento.	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: #241953-ConsultarListadoDeLíneas, para posteriormente definir cada una de las pruebas mencionadas.

## 2. Caso de uso: Consultar listado de líneas

<b>Identificador:</b>	#241953
<b>Caso de uso:</b>	Consultar listado de líneas
<b>Descripción:</b>	El usuario visualiza el listado de todas las líneas de transporte urbano reconocidas por el Ayuntamiento de Santander. Por defecto, se visualizan por orden numérico y coloreadas según el color oficial asignado a la línea.
<b>Actores:</b>	Usuario habitual del TUS.
<b>Secuencia:</b>	<ol style="list-style-type: none"><li>1. El usuario pulsa el botón “Líneas” de la barra de navegación inferior.</li><li>2. El sistema obtiene remotamente los detalles de las líneas y las procesa.</li><li>3. El usuario visualiza el listado de líneas pudiendo desplazarse por éste.</li></ol>
<b>Extensiones:</b>	<ol style="list-style-type: none"><li>2.a. Si se carece de conexión a internet, el sistema lo notificará y listará las líneas.</li><li>2.b. Si no se puede acceder a la fuente de datos, se listarán los datos de las líneas que se guardaron en la base de datos en la última descarga.</li></ol>

## 3. Pruebas de aceptación

En el presente apartado se indicarán las pruebas de aceptación que se llevarán a cabo, de manera automatizada y utilizando las herramientas oportunas del entorno de desarrollo, para comprobar la correcta funcionalidad de las implementaciones realizadas:

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

1. El usuario pulsa el botón “Líneas” situado en la barra de navegación inferior.
2. El usuario espera a la carga de datos.
3. El sistema lista las líneas en orden numérico y coloreando las líneas según el color oficial.

El resultado esperado es poder visualizar un listado de líneas en orden numérico y coloreadas por el color oficial asignado por el Ayuntamiento de Santander.

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

1. El usuario pulsa el botón “Líneas” situado en la barra de navegación inferior.
2. El usuario espera a la carga de datos.
3. El sistema notifica la falta de conexión a internet.

El resultado esperado es poder visualizar un mensaje de error que notifique la falta de conexión a la red y liste las líneas que estén guardadas en la base de datos a partir de la última conexión. Cuando la base de datos este vacía no se mostrará nada.

**PA3: Comprobar correcto funcionamiento al listar líneas. (Fuente de datos no accesible):**

1. El usuario pulsa el botón “Líneas” situado en la barra de navegación inferior.
2. El usuario espera a la carga de datos.
3. El sistema muestra una lista con los datos guardados en la base de datos desde la última descarga.

El resultado esperado es poder visualizar un listado con las líneas que estén guardadas en la base de datos a partir de la última conexión. Cuando la base de datos este vacía no se mostrará nada.

## **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 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 por la capa *View*.

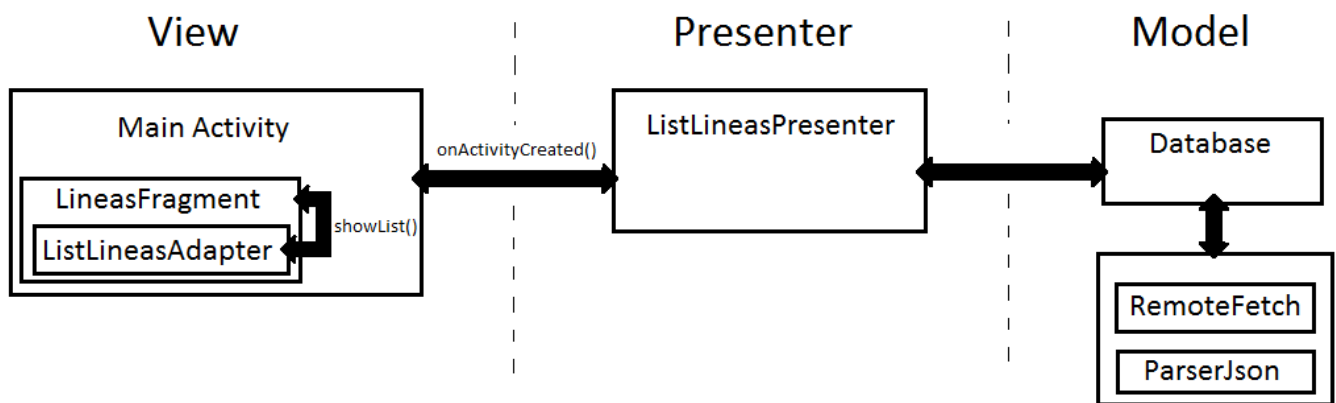


Figura 2. Diseño de arquitectura de la historia de usuario "Consultar listado de líneas"

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 "Consultar listado de líneas" es *showList()* el cuál se utiliza en la capa *View*, ya que sirve para determinar 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 las funcionalidades anteriormente implementadas. En el siguiente diagrama, se muestra la interacción que se produce entre los elementos de nuestro producto y los métodos que les da la funcionalidad a las historias de usuario "Listar paradas", "Ordenar paradas alfabéticamente", "Visualizar logo en arranque", "Buscar parada por texto" y "Consultar listado de líneas".

Tras haber comprobado mediante pruebas de integración el buen funcionamiento del sistema integrado por las funcionalidades "Listar paradas", "Ordenar paradas alfabéticamente" y "Visualizar logo en arranque" (en el anterior sprint se comprobó la correcta respuesta del sistema mostrando la actividad en la que se muestra el logo), por lo tanto, se comprobará el buen funcionamiento del producto al integrar las historias "Consultar el listado de líneas" y "Buscar parada por texto".

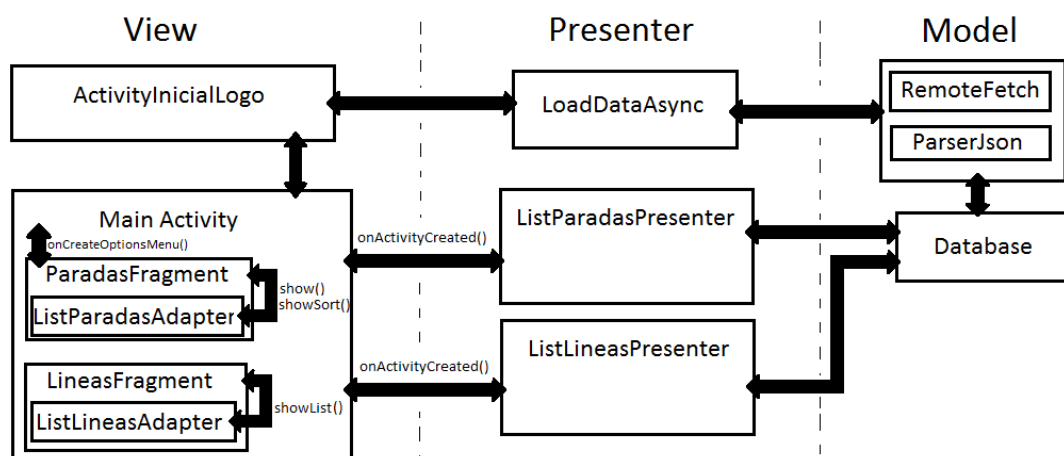


Figura 3. Diseño de arquitectura con todas las funcionalidades integradas

A partir del diseño planteado, y teniendo en cuenta las operaciones que se implican, se ha diseñado el caso de prueba que se indica a continuación:

**PI1.** Probar que se devuelve un listado de líneas ordenado numéricamente, tras visualizar el logo y pasar por la pantalla del listado de paradas. Con ello se comprobará el correcto funcionamiento de la historia de usuario “Consultar listado de líneas” integrada con el resto de funcionalidades implementadas. Este caso se realizará 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 variar invalidando las pruebas realizadas.

**PI2.** Probar la anterior prueba, pero en este caso, realizando la prueba integrando la base de datos.

## 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": 33, "items_per_page": 1000, "pages": 1, "current_page": 1 }, "resources": [
  {
    "ayto:numero": "N2",
    "dc:name": "CORBAN-COMPLEJO por G. Davila",
    "dc:modified": "2017-11-12T00:00:00.74Z",
    "dc:identifier": "102",
    "uri": "http://datos.santander.es/api/datos/lineas_bus/102.json"
  },
  {
    "ayto:numero": "20",
    "dc:name": "ESTACIONES-BARRIO LA TORRE",
    "dc:modified": "2017-09-11T23:00:00.976Z",
    "dc:identifier": "20",
    "uri": "http://datos.santander.es/api/datos/lineas_bus/20.json"
  },
  {
    "ayto:numero": "7C2",
    "dc:name": "JOAQUIN BUSTAMANTE",
    "dc:modified": "2017-11-12T00:00:00.74Z",
    "dc:identifier": "72",
    "uri": "http://datos.santander.es/api/datos/lineas_bus/72.json"
  },
  {
    "ayto:numero": "7C1",
    "dc:name": "LUIS QUINTANILLA",
    "dc:modified": "2017-11-12T00:00:00.74Z",
    "dc:identifier": "71",
    "uri": "http://datos.santander.es/api/datos/lineas_bus/71.json"
  },
  {
    "ayto:numero": "6C2",
    "dc:name": "COMPLEJO C2",
    "dc:modified": "2017-11-12T00:00:00.74Z",
    "dc:identifier": "62",
    "uri": "http://datos.santander.es/api/datos/lineas_bus/62.json"
  }
]
```



- **Casos de prueba**

**PU1.** Se deberá probar el método *“obtenLineas()”* de la clase *LoadDataAsync*, de manera que al proporcionarle el Json de entrada mostrado anteriormente, el método *“obtenLineas()”* retorne un listado como el que se muestra a continuación.

Nombre	Identificador	Número
COMPLEJO C2	62	6C2
LUIS QUINTANILLA	71	7C1
JOAQUIN BUSTAMANTE	72	7C2
ESTACIONES-BARRIO LA TORRE	20	20
CORBAN-COMPLEJO por G. Davila	102	N2

**PU2.** Se deberá probar el método *“obtenLineas()”* de la clase *LoadDataAsync*, de manera que, al proporcionarle un Json de entrada vacío, el método *“obtenLineas()”* retorne un listado vacío de paradas. Esto simula el resultado del caso en el que no sea posible obtener las líneas de forma remota y no haya ningún dato almacenado en la base de datos.

## 6. Sumario

Como resumen del contenido del documento, se ha mostrado tanto la definición del caso de uso, como el diseño de las pruebas a seguir para comprobar el buen comportamiento del sistema.