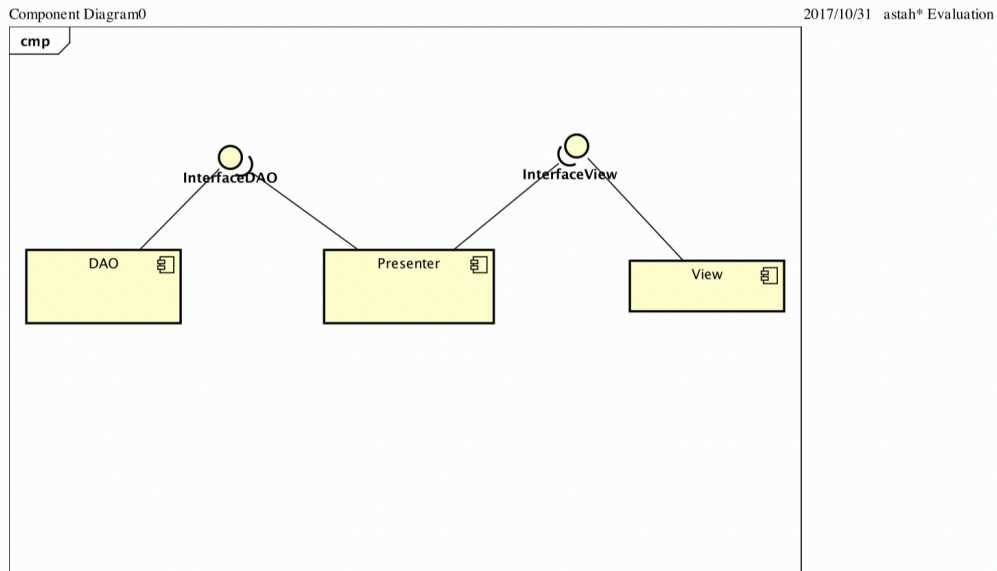


# Plan de pruebas

Luis Martin de la Rubia

## Diagrama de componentes orientativo

Este diagrama indica las dependencias lógicas de la aplicación.



[illegible]

Estas pruebas comprobaran el funcionamiento de todos los componentes de la aplicación estas pruebas están recogidas en el paquete estimaciones de la carpeta aceptación de androidTest.

1. Se selecciona la opción de Líneas en el menú principal

1. Se selecciona la opción de Líneas en el menú principal
2. Se selecciona una línea de autobuses.
3. Se selecciona una parada de esa línea.
4. El sistema muestra las estimaciones de llegada para esa parada mostrando en primer lugar la línea previamente seleccionada.

1. Se selecciona la opción de Líneas en el menú principal

1. Se selecciona la opción de Líneas en el menú principal
2. Se selecciona una línea de autobuses.
3. Se selecciona una parada de esa línea.
4. Se da al botón de recargar en la parte superior.
5. Se comprueba que se recargan los datos mostrados de las estimaciones.

### Pruebas unitarias:

Estas pruebas se recogerán en el paquete estimaciones del paquete pruebasunitarias del paquete androidTest.

En primer lugar, realizaremos pruebas de la función de obtener los datos del JSON con las estimaciones usando un pequeño JSON de prueba con el mismo formato que el obtenido del servicio del ayuntamiento.

U1. La entrada será el fichero JSON de prueba con los siguientes datos:

```
{ "summary": { "items": 1897, "items_per_page": 1897, "pages": 1, "current_page": 1, "resources": [
  {
    "ayto:tiempo1": "66",
    "ayto:distancia2": "14507",
    "ayto:destino1": "PIQUIO",
    "ayto:distancia1": "7445",
    "ayto:tiempo2": "225",
    "ayto:paradaId": "1",
    "ayto:destino2": "PIQUIO",
    "ayto:fechaActual": "2017-11-26T15:17:09.237Z",
    "dc:modified": "2017-11-26T15:17:17.236Z",
    "dc:identifier": "125",
    "ayto:etiqLinea": "3",
    "uri": "http://datos.santander.es/api/datos/control_flotas_estimaciones/125.json"},
  {
    "ayto:tiempo1": "88",
    "ayto:distancia2": "5724",
    "ayto:destino1": "CONSUELO BERGES",
    "ayto:distancia1": "608",
    "ayto:tiempo2": "1137",
    "ayto:paradaId": "2",
    "ayto:destino2": "CONSUELO BERGES",
    "ayto:fechaActual": "2017-11-26T15:17:09.237Z",
    "dc:modified": "2017-11-26T15:17:17.236Z",
    "dc:identifier": "126",
    "ayto:etiqLinea": "2",
    "uri": "http://datos.santander.es/api/datos/control_flotas_estimaciones/126.json"}
  ] }
```

y la salida esperada es una lista con las estimaciones del json.

A continuación, probaremos el funcionamiento del método ordena() de la clase EstimacionesPresenter, para ello realizaremos las siguientes pruebas:

U2. Con una lista de estimaciones con los siguientes datos:

```
e1 = new Estimacion("1", "12", "641", "5", "1241243456412");
e2 = new Estimacion("2", "22", "451", "4", "1241adad21412");
e3 = new Estimacion("3", "42", "121", "5", "12412");
```

Para la línea: 1 y la parada: 5

Se espera que devuelva una lista ordenada por el primer tiempo de llegada en la cual el primer elemento sea la línea.

U3. Con una lista de estimaciones vacía para la línea: 1 y la parada: 5

Se espera que devuelva la lista vacía.

U4. Con una lista de estimaciones con los siguientes datos:

```
e1 = new Estimacion("1","12","641","5","1241243456412");  
e2 = new Estimacion("2","22","451","4","1241adad21412");  
e3 = new Estimacion("3","42","121","5","12412");
```

Para la línea: 13 y la parada: 5

Aunque la línea no se encuentra en la lista de estimaciones se espera que devuelva la lista ordenada por el primer tiempo

U5. Con una lista de estimaciones con los siguientes datos:

```
e1 = new Estimacion("1","12","641","5","1241243456412");  
e2 = new Estimacion("2","22","451","4","1241adad21412");  
e3 = new Estimacion("3","42","121","5","12412");
```

Para la línea: 1 y la parada: 25

Como la parada no se encuentra en la lista de estimaciones se espera que devuelva una lista vacía.

### Pruebas de Integración:

Estas pruebas estarán recogidas en el paquete estimaciones del paquete integración de la carpeta androidTest.

Se comprobará el funcionamiento en primer lugar del presenter con la parte de datos, en este caso la clase EstimacionesPresenter con el FetchJson y parseJson.

Estas pruebas se realizarán sobre el método obténData de la clase EstimacionesPresenter.

I1. Se probará a ejecutar el método y se comprobará que ocurre cuando todo funciona correctamente que debería devolver true y rellenar la lista de estimaciones.

I2. Se probará a ejecutar el método cuando no existe conexión a internet cambiando el valor del atributo activeNetworkInfo a null y comprobando que el método retorna false indicando el error.