

<b>Título</b>	TTT_Android_Viajero_Documentación_Final		
<b>Proyecto</b>	Práctica de la asignatura Proyectos Informáticos: Tiempos del Transporte de Teruel (TTT)		
<b>Equipo</b>	Jorge Navarrete Argilés  Jesús Fuentes Romero  Óscar Munárriz Sesma  Ángel Sánchez Espílez		
<b>Código</b>	TTT_Andr_Via_Docu	<b>Fecha</b>	16/06/11
<b>Profesor</b>	Félix Serna Fortea	<b>Versión</b>	1.0

## 0. ÍNDICE

<b>0. ÍNDICE.....</b>	<b>2</b>
<b>1. INTRODUCCION.....</b>	<b>4</b>
1.1 Propósito.....	4
1.2 Alcance.....	4
1.3 Definiciones, acrónimos y abreviaturas.....	5
<b>2. DESCRIPCIÓN GLOBAL.....</b>	<b>6</b>
2.1 Perspectiva del producto.....	6
2.1.1 Interfaz de sistema.....	6
2.1.2 Interfaz software.....	6
2.1.3 Diagramas de Casos de Uso.....	7
<b>3. CARACTERÍSTICAS DEL PRODUCTO.....</b>	<b>9</b>
3.1 Características específicas.....	9
3.2 Restricciones de sistema .....	9
3.3 Restricciones de usabilidad.....	9
<b>4. REQUISITOS ESPECÍFICOS.....</b>	<b>10</b>
4.1 Requisitos generales.....	10
4.2 Requisitos específicos.....	10
4.2.1 General.....	10
4.2.2 Vista lista de líneas.....	10
4.2.3 Vista detalle de línea.....	10
4.2.4 Vista detalle de parada.....	10
4.2.5 Vista de mapa.....	10
4.2.6 Favoritos.....	11
4.3 Requisitos de sistema.....	11
<b>5. DIAGRAMAS UML.....</b>	<b>12</b>
5.1 Diagramas Clases.....	12
5.2 Diagramas Actividad.....	14
5.3 Diagramas Secuencia.....	16

<b>6. INTERFACES DE USUARIO.....</b>	<b>18</b>
<b>6.1 Análisis de requisitos.....</b>	<b>18</b>
A.- Desarrollo del concepto de producto.....	18
A.1.- Escenarios / storyboards.....	18
A.2.- Análisis competitivo.....	19
B.- Análisis de requisitos. Informe de usuarios / roles de la aplicación.....	19
B.1.1.- Posibles usuarios de la aplicación.....	19
B.1.2.- Fichas de usuarios.....	19
C.- Tareas.....	20
C.1.- Especificar las tareas.....	20
C.2.- Diagrama HTA de cada tarea.....	20
D.- Especificaciones .....	21
D.1.- De usabilidad.....	21
E.- Prototipado. ....	23
E.1. - Card Sorting .....	23
E.2.- Prototipos .....	25
E.3.- Recorrido Pluralístico.....	27
F.- Factor tecnológico.....	28
<b>6.2 Diseño.....</b>	<b>29</b>
1.- Guía de estilo. ....	29
Look & feel del proyecto. ....	29
Colores (paleta de colores) .....	29
Disposición de menús. ....	30
Formato de los listados. ....	31
Pautas de lenguaje y diccionario propio (vocabulario específico).....	32
2.- Diseño del prototipo no funcional.....	33
3.- Árbol de pantallas de nuestra aplicación.....	36
4.- Requerimientos para su uso e instalación.....	37
<b>7. VTP.....</b>	<b>38</b>
<b>7.1. Introducción.....</b>	<b>38</b>
<b>7.2. Primeras pruebas.....</b>	<b>38</b>
<b>7.3. Planificación de pruebas.....</b>	<b>38</b>
<b>7.4. Pruebas que se van a realizar para la aplicación viajero.....</b>	<b>39</b>
<b>7.5. Necesidades para realizar las pruebas.....</b>	<b>39</b>

# 1. INTRODUCCION

El presente documento contiene una Especificación de Requisitos de Software (ERS) para la aplicación informática que el viajero podrá instar en su teléfono o tableta Android para ver el tiempo que le falta a un autobús para llegar a su parada. Esta es una parte del sistema de “Tiempos de Transporte de Teruel”, a realizar durante la asignatura Proyectos Informáticos. La estructura del presente documento está inspirada en las directrices referidas en el estándar IEEE Recommended Practice for Software Requirements Specifications IEEE 830-1998.

El objetivo principal de esta aplicación es proporcionar al viajero información sobre el tiempo que tardará un autobús en llegar a una parada, así como mostrarle las paradas que tiene a su disposición.

## 1.1 Propósito

El objeto de la especificación de requisitos es definir de manera clara y precisa tanto las funcionalidades como las posibles restricciones del sistema de información a construir. Dicho documento está dirigido tanto a los miembros del equipo de desarrollo como al cliente del proyecto. En él, se establecen las bases sobre las cuales el equipo de desarrollo procederá al diseño y posterior construcción del nuevo sistema. También servirá de modelo en el cual basar las pruebas funcionales de aceptación por parte del cliente y la evaluación final del producto por parte del mismo.

## 1.2 Alcance

El producto a obtener será una aplicación dirigida a los usuarios del servicio de autobuses de Teruel. Esta aplicación proporcionará información de la situación de las paradas y los tiempos de espera para que el viajero conozca de antemano cuánto va a tardar un autobús en llegar.

Se requerirá toda la documentación de análisis y diseño, previa a la implementación del sistema.

### 1.3 Definiciones, acrónimos y abreviaturas

<b>Línea</b>	Secuencia de paradas donde un viajero puede coger un autobús para llegar a otra de las paradas de la línea
<b>Sentido</b>	Cada una de las direcciones en las que se puede recorrer una línea
<b>Ruta</b>	El trazado que une las paradas de una línea
<b>Parada</b>	Punto localizado geográficamente donde los autobuses de ciertas líneas pueden dejar o coger viajeros
<b>Vista</b>	Cada una de las pantallas que la aplicación puede mostrar
<b>Mapa</b>	Representación gráfica de las vías de circulación a partir de una vista aérea.
<b>Android</b>	Sistema operativo para dispositivos móviles compatible con aplicaciones

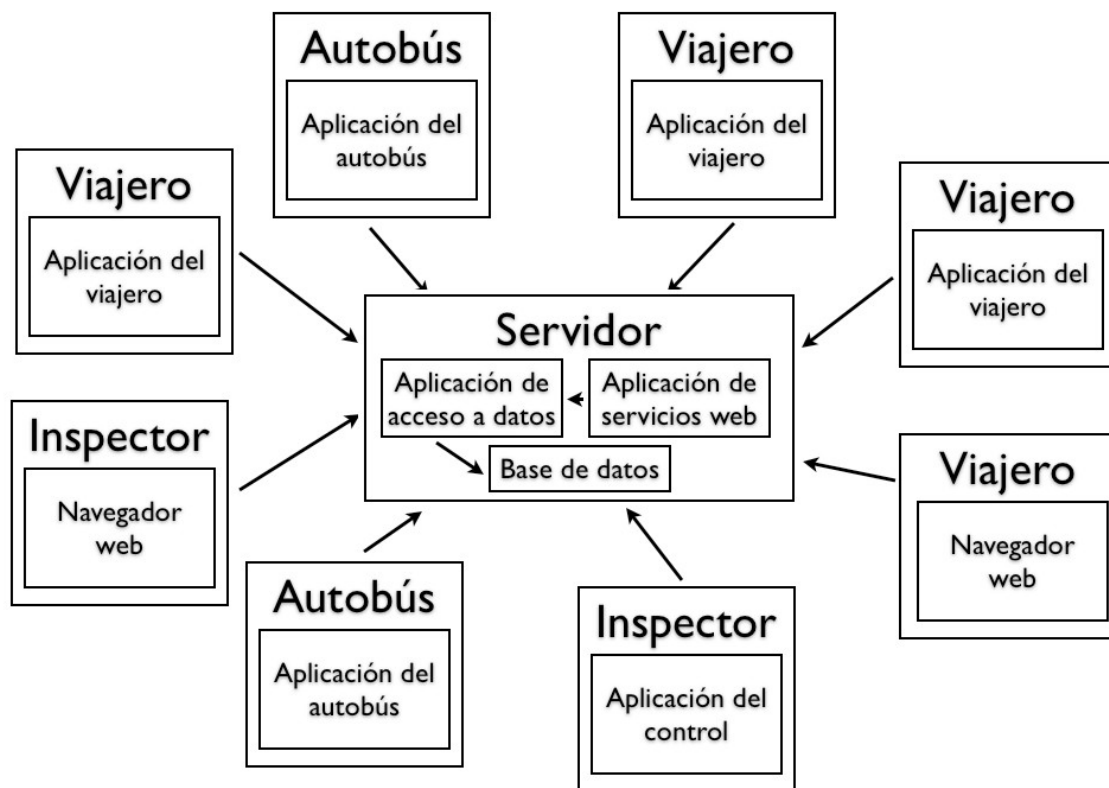
## 2. DESCRIPCIÓN GLOBAL

### 2.1 Perspectiva del producto

La aplicación obtiene del sistema la información necesaria, por lo que se necesitará conectar a una aplicación central. Para ello se debe seguir un protocolo por definir.

#### 2.1.1 Interfaz de sistema

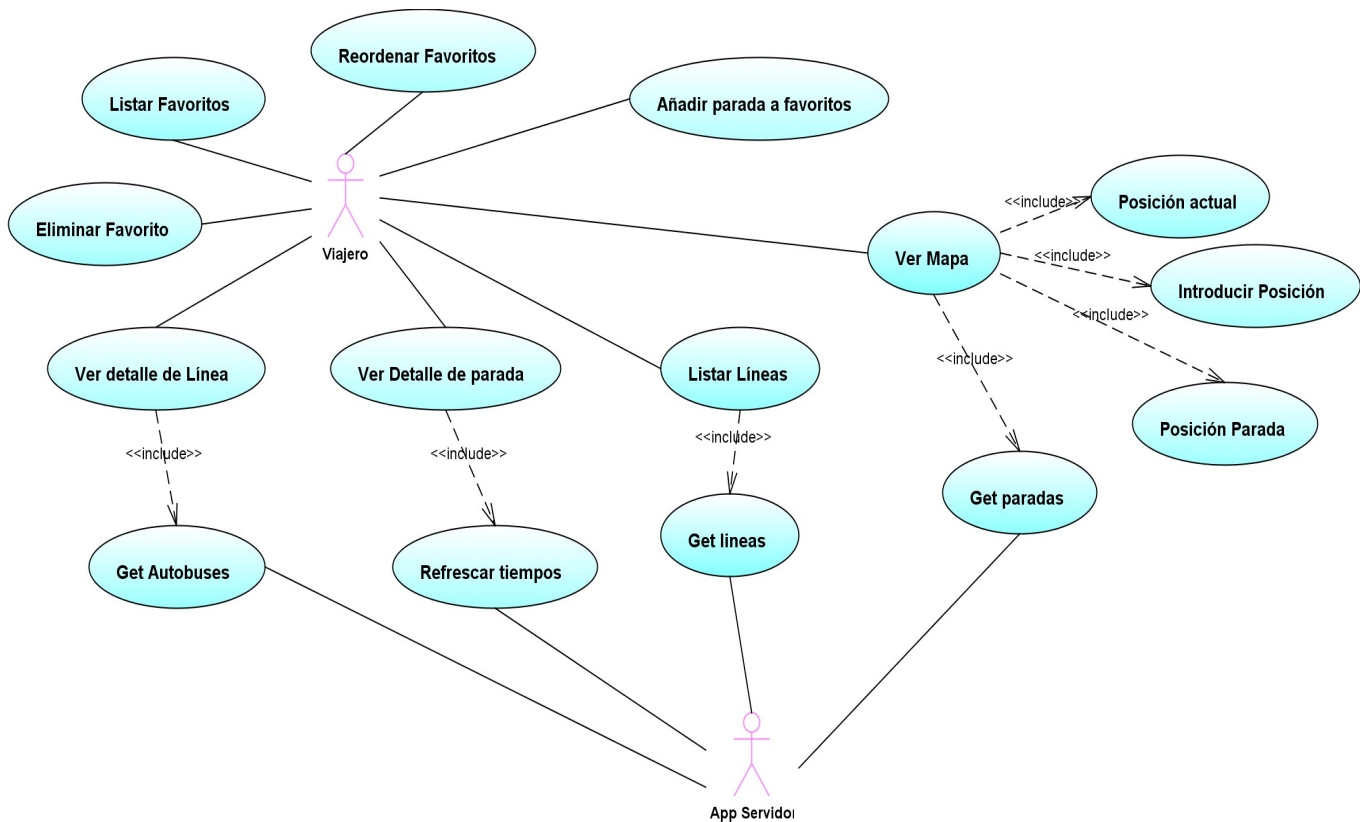
La aplicación para el viajero actúa como una aplicación cliente en el sistema, es decir, se debe conectar a otra aplicación central para su correcto funcionamiento.



#### 2.1.2 Interfaz software

Se requiere Android 2.1 o superior.

### 2.1.3 Diagramas de Casos de Uso



**Listar Favoritos:** El viajero ve un listado con las paradas que haya almacenado para un rápido acceso a éstas.

**Reordenar Favoritos:** El viajero puede modificar el orden en la pantalla de los favoritos.

**Eliminar Favorito:** Si el viajero desea eliminar una parada de los favoritos puede borrarla de la lista.

**Añadir parada a Favoritos:** Cuando el viajero está viendo los tiempos de espera de una parada tiene la opción de añadir esta parada a los favoritos para un acceso rápido a ella.

**Ver Detalle Línea:** La aplicación mostrará al controlador un mapa donde se podrá ver la ubicación geográfica de los autobuses y paradas de la línea.

**Ver detalle de Parada:** Al elegirse una parada se muestran los tiempos de espera de cada línea que pasa por esta parada.

**Listar líneas:** La aplicación mostrará al viajero una vista con un listado de todas las líneas de Teruel.

**Ver mapa:** El usuario puede ver un mapa de todas las paradas de Teruel. Este mapa se puede mostrar centrado en la **posición actual**, en una **posición introducida** o en la **posición de una parada**.

**Get autobuses:** El servidor debe responder devolviendo un listado con todos los autobuses que están cubriendo líneas en ese momento.

**Refrescar tiempos:** El servidor debe responder con un listado de las líneas, sentido, nombre de la línea y tiempo de espera de la parada solicitada.

**Get líneas:** La aplicación del servidor debe responder devolviendo un listado con todas

las líneas del sistema.

**Get paradas:** La aplicación del servidor debe responder con un listado de todas las paradas de la ciudad.



### 3. CARACTERÍSTICAS DEL PRODUCTO

Las características del producto definen las funcionalidades que tendrá la aplicación. Las características se detallan en el punto 4 “Requisitos específicos”.

#### 3.1 Características específicas

**CAR\_VIA\_01:** El viajero puede ver la lista de las líneas de la ciudad en la vista de Líneas. Se le mostrará un listado con todas las líneas que tiene la ciudad.

**CAR\_VIA\_02:** El usuario puede ver el detalle de las líneas de la ciudad con la vista Detalle de línea. En ella se le mostrará el mapa de la ciudad con las paradas pertenecientes a esa línea.

**CAR\_VIA\_03:** El viajero puede ver la información de tiempos de espera en cada parada en la vista Detalle de parada.

**CAR\_VIA\_04:** El viajero dispone de un mapa con todas las paradas de la ciudad para navegarlo o ubicarse en este mapa en la vista de Mapa.

**CAR\_VIA\_05:** El viajero puede almacenar paradas en una lista de favoritos que se podrá acceder mediante la vista Favoritos.

#### 3.2 Restricciones de sistema

**RES\_GEN\_01:** Se requiere Android 2.1 o superior.

#### 3.3 Restricciones de usabilidad

**RES\_INT\_01:** Para la visualización de mapas se debe usar la API de Google Maps para Android.

## 4. REQUISITOS ESPECÍFICOS

### 4.1 Requisitos generales

**REQ\_GEN\_01\_01:** La aplicación obtendrá la información cuando el usuario lo solicite o bien pasados unos segundos desde la anterior actualización.

### 4.2 Requisitos específicos

Los requisitos están agrupados por la vista a la que se refiere.

#### 4.2.1 General

**REQ\_VIA\_00\_01:** La aplicación presenta una pantalla de bienvenida como pantalla inicial. Después se muestra el mapa de la ciudad con todas las paradas.

**REQ\_VIA\_00\_02:** Desde cualquier pantalla el usuario dispone de la opción de saltar a la vista de Mapa, Favoritos o Líneas.

#### 4.2.2 Vista lista de líneas

**REQ\_VIA\_01\_01:** La aplicación presenta en una pantalla la lista de líneas donde se enumeran junto a sus sentidos. El viajero dispone la opción de elegir una de ellas para ver su vista Detalle de línea, en la cual se mostrarán todas las paradas por las que pasa situadas en el mapa.

#### 4.2.3 Vista detalle de línea

**REQ\_VIA\_02\_01:** La aplicación debe mostrar, para cada línea y sentido, sus paradas situadas en el mapa.

**REQ\_VIA\_02\_02:** La aplicación debe ubicar gráficamente la situación de las paradas de la línea de forma que al pulsar sobre uno de ellos se accede a su detalle.

#### 4.2.4 Vista detalle de parada

**REQ\_VIA\_03\_01:** Se debe enumerar las líneas a las que sirve esa parada y el tiempo que falta hasta el paso del siguiente autobús de esa línea.

**REQ\_VIA\_03\_02:** Se dispone de la opción de pulsar sobre una de las líneas para acceder a su vista Detalle de línea.

**REQ\_VIA\_03\_03:** El viajero puede añadir la parada que está viendo a los favoritos, introduciendo una descripción identificativa para almacenarla.

**REQ\_VIA\_03\_04:** Existe un botón atrás que nos devuelve a la vista que nos ha remitido a este detalle de parada (Favoritos, Mapa o Detalle de línea).

**REQ\_VIA\_03\_05:** El viajero dispone de la posibilidad de saltar a la vista Mapa con el plano centrado en esta parada.

**REQ\_VIA\_03\_06:** Periódicamente la aplicación actualizará la información de los tiempos. Si el usuario desea forzarlo, dispondrá de la opción de refrescar.

#### 4.2.5 Vista de mapa

**REQ\_VIA\_04\_01:** El mapa muestra todas las paradas existentes en la ciudad.

**REQ\_VIA\_04\_02:** Cuando el usuario pulse sobre una parada se salta a su vista Detalle de parada, se le muestra las líneas que pasan por la misma.

**REQ\_VIA\_04\_03:** El viajero dispone de la opción de centrar la el mapa en la posición actual.

#### **4.2.6 Favoritos**

**REQ\_VIA\_05\_01:** Los favoritos consisten en una lista almacenada de forma local.

**REQ\_VIA\_05\_02:** La aplicación muestra para cada favorito la descripción introducida por el usuario cuando se almacenó.

**REQ\_VIA\_05\_03:** Cuando el viajero pulse sobre un favorito se salta a su vista Detalle de parada.

**REQ\_VIA\_05\_04:** La misma parada no puede aparecer dos veces en los favoritos.

### **4.3 Requisitos de sistema**

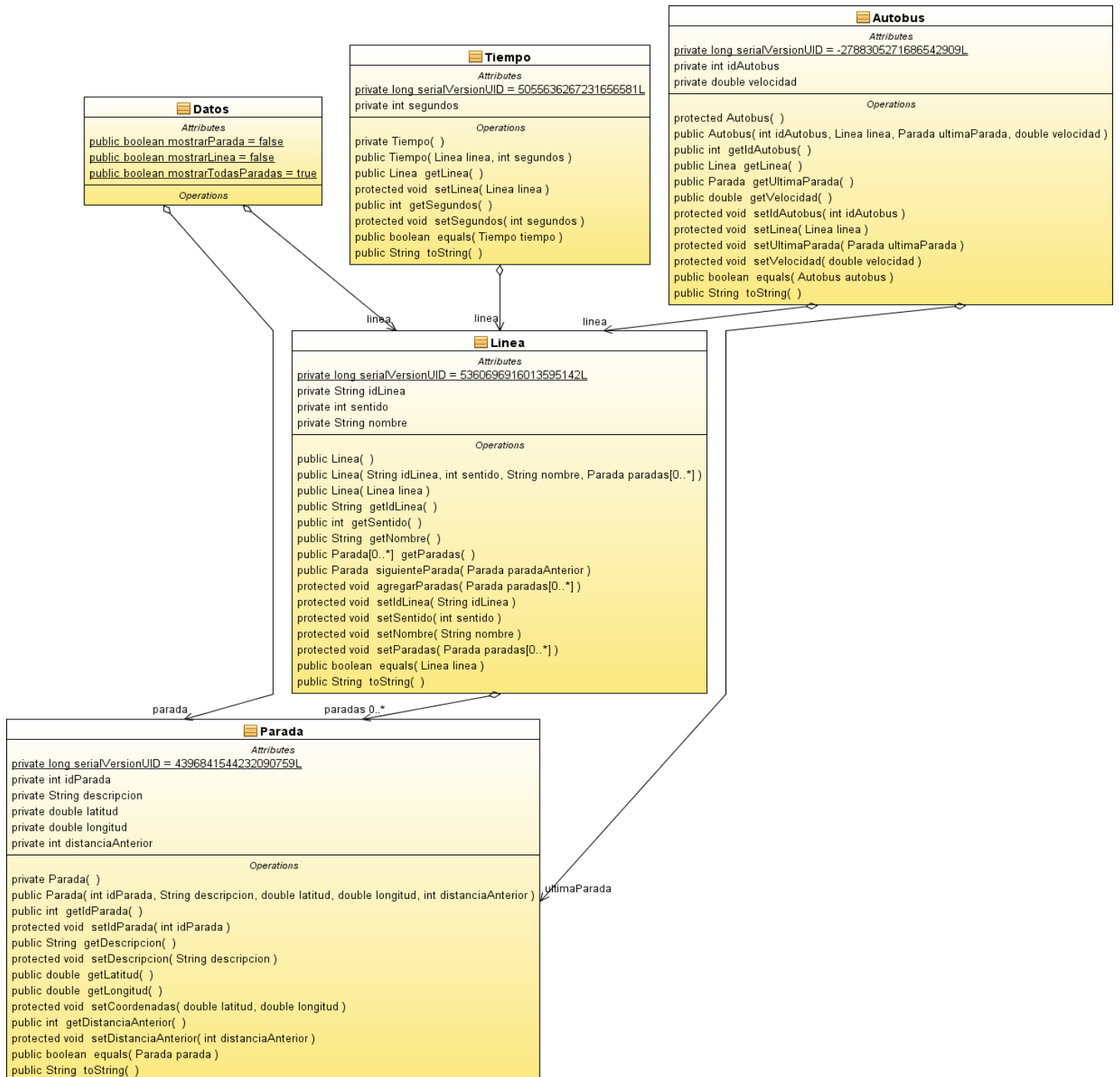
**REQ\_SIS\_01:** Se requiere Android 2.1 o superior.

**REQ\_SIS\_02:** La aplicación debe mantener la conexión con el servidor el mínimo tiempo posible.

**REQ\_SIS\_03:** La aplicación solamente puede hacer consultas a la base de datos, no modificar la información.

## 5. DIAGRAMAS UML

### 5.1 Diagramas Clases



### **Explicación diagrama de Clases:**

La clase Linea contiene la información del identificador de la línea, el sentido de la marcha en el que se circula, y el nombre de la línea. Tiene dos métodos para comprobar cual es la siguiente parada a la que se debe ir (siguienteParada()) y el otro para agregar más paradas a las pertenecientes a esa línea (agregarParada()). El resto de métodos son get y set además de los constructores.

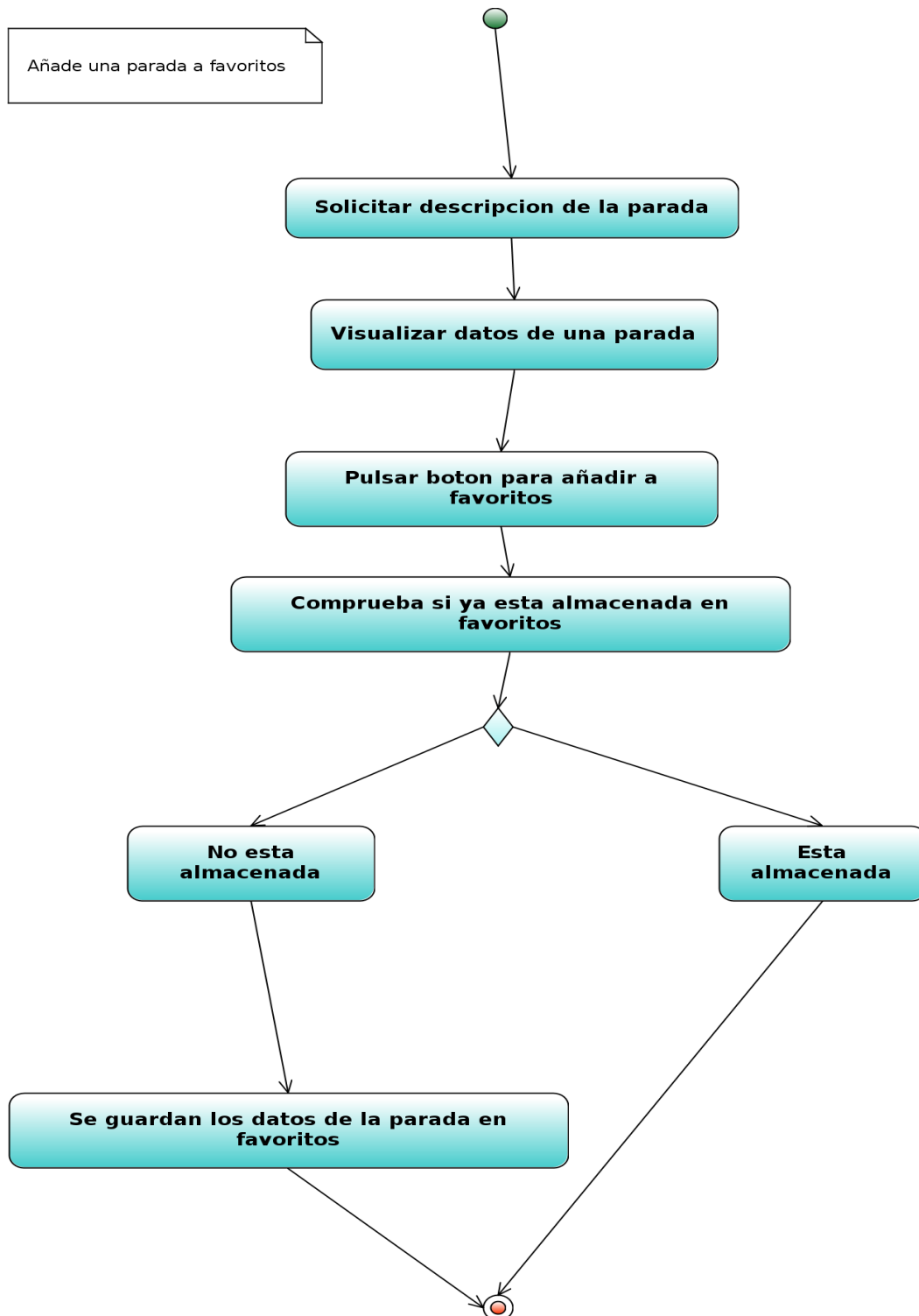
Por otra parte la clase Tiempo se encarga de almacenar el tiempo de paso de un autobús por una parada y sirve para poder procesar las peticiones del servidor. Pide las diferentes peticiones necesarias para la gestión del viajero, como obtener todas las paradas de una determinada línea. Esta clase solo contiene métodos get y set además de los constructores.

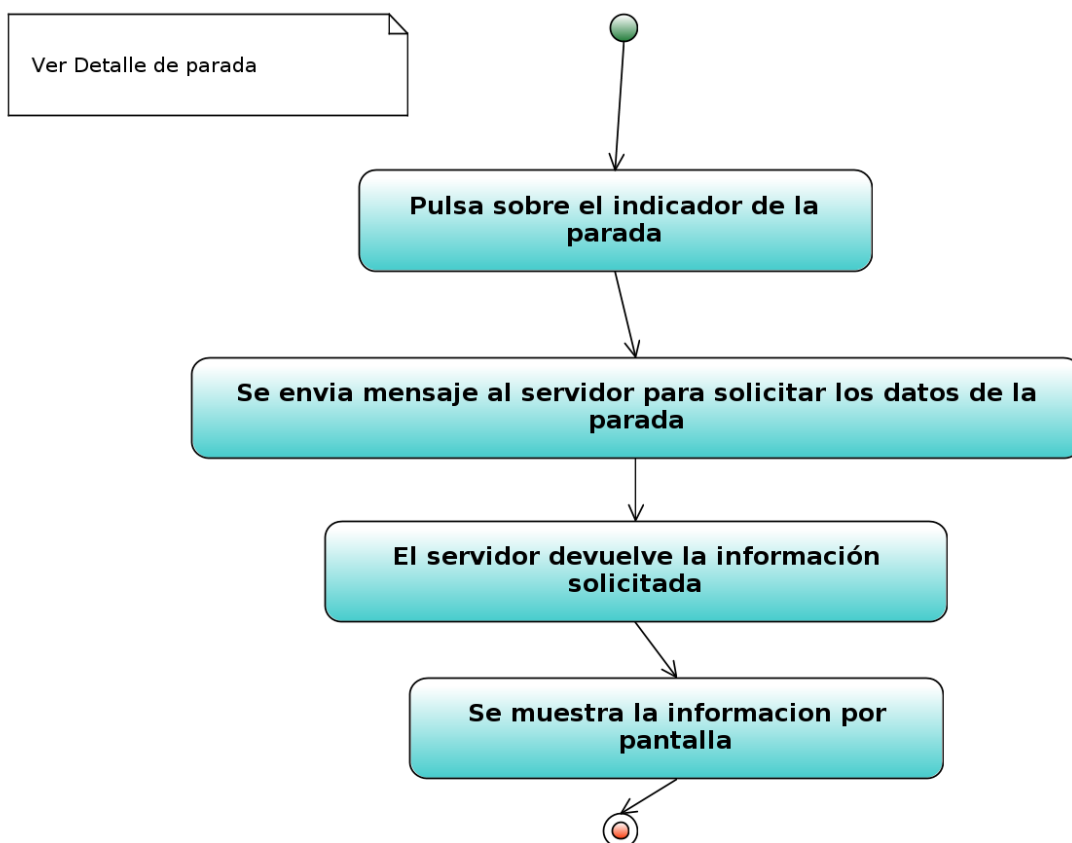
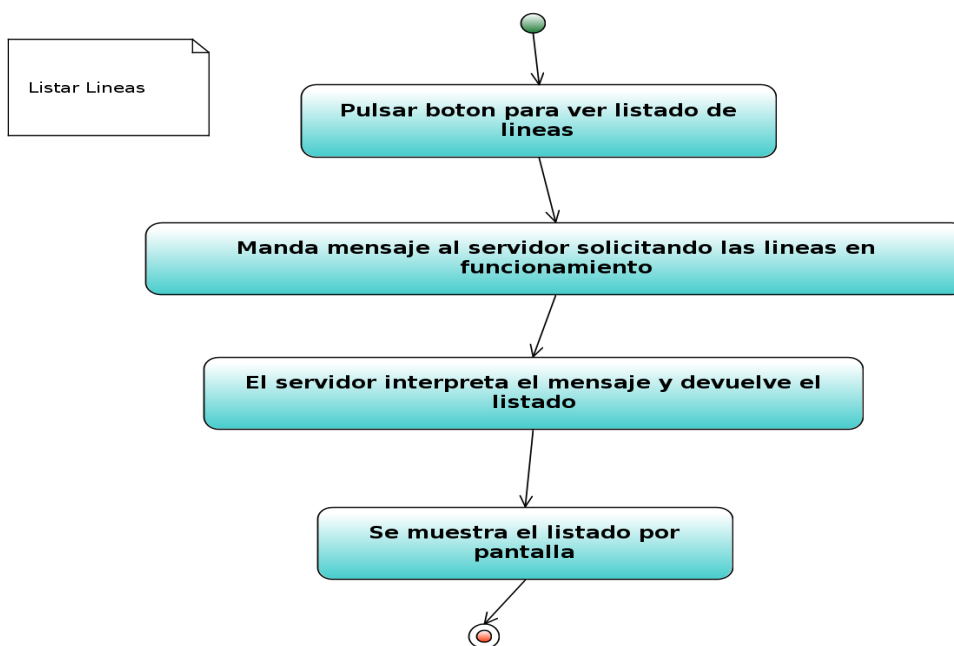
La clase Parada por su parte almacena todos los datos de una parada como su identificador, su descripción su posición y la distancia a la anterior parada. Esta clase solo contiene métodos get y set junto con sus constructores.

Para finalizar la clase Autobús almacena el identificador del autobús, la velocidad actual, la línea que está recorriendo así como la última parada por la que ha pasado.

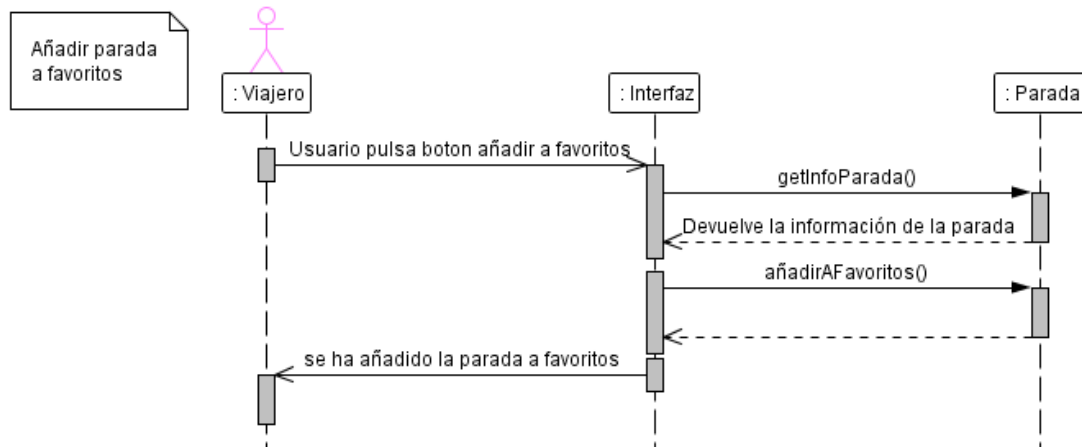
La clase datos contiene tres booleanos que son utilizados como banderas para hacer determinadas acciones, por ejemplo al arrancar la aplicación en el primer mapa deben aparecer todas las paradas existentes en Teruel, otro caso seria si el usuario a hecho clic sobre alguna línea del listado de líneas, debe cargar las paradas por las que pasan los autobuses que recorren esa línea.

## 5.2 Diagramas Actividad

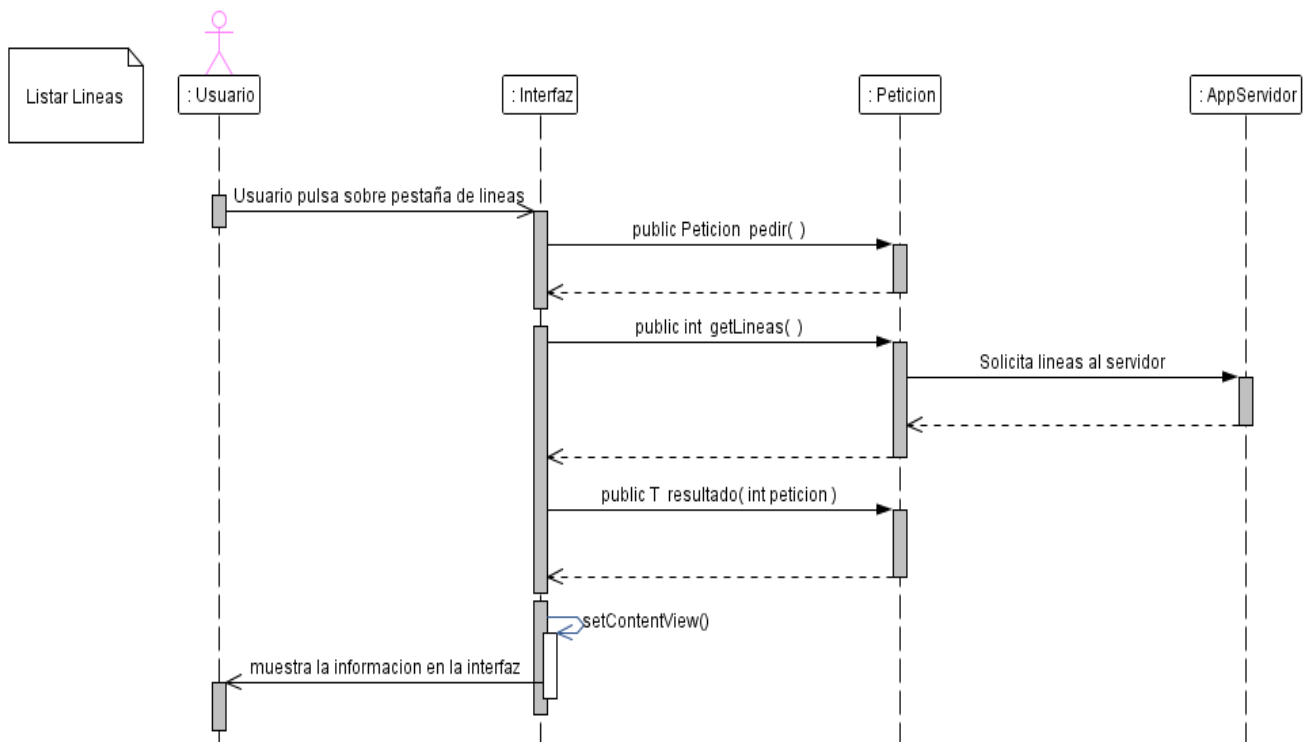




### 5.3 Diagramas Secuencia

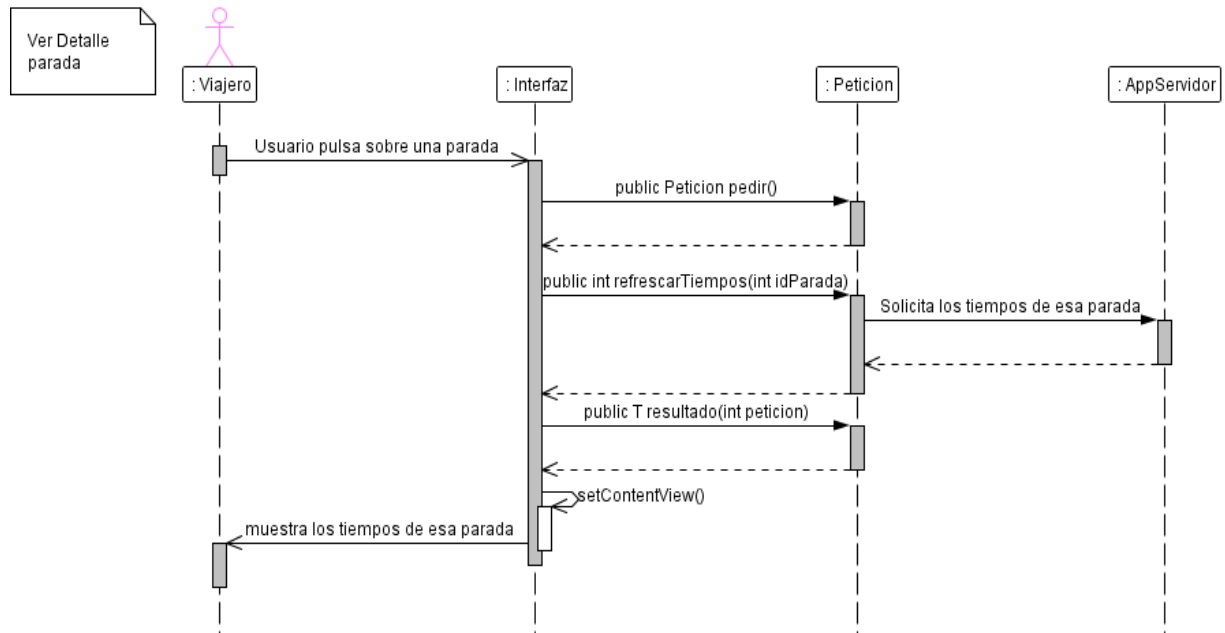


Una vez que el usuario esta en el detalle de parada tiene la opción de añadir esa parada a favoritos pulsando en el botón “Añadir a favoritos”. En primer lugar obtiene la información de esa parada y añade esa parada a la lista de favoritos.



Si el usuario pulsa la pestaña Lineas, se mostrará el listado de todas las líneas que recorren la ciudad. La interfaz hace una petición al servidor para que le de toda la información de la base de datos correspondiente a las líneas. Una vez obtiene la información las muestra en forma de listado. Si el usuario pulsa una de las líneas de la lista nos llevará al mapa de la ciudad en el que se dibujarán todas las paradas de esa línea.





Cuando un usuario pulsa una de las paradas del mapa, nos lleva a una actividad, en la que se mostrarán todas las líneas que pasan por esa parada en forma de lista. También nos informa de si esa línea está fuera de servicio o el tiempo que tardará en pasar el siguiente autobús.

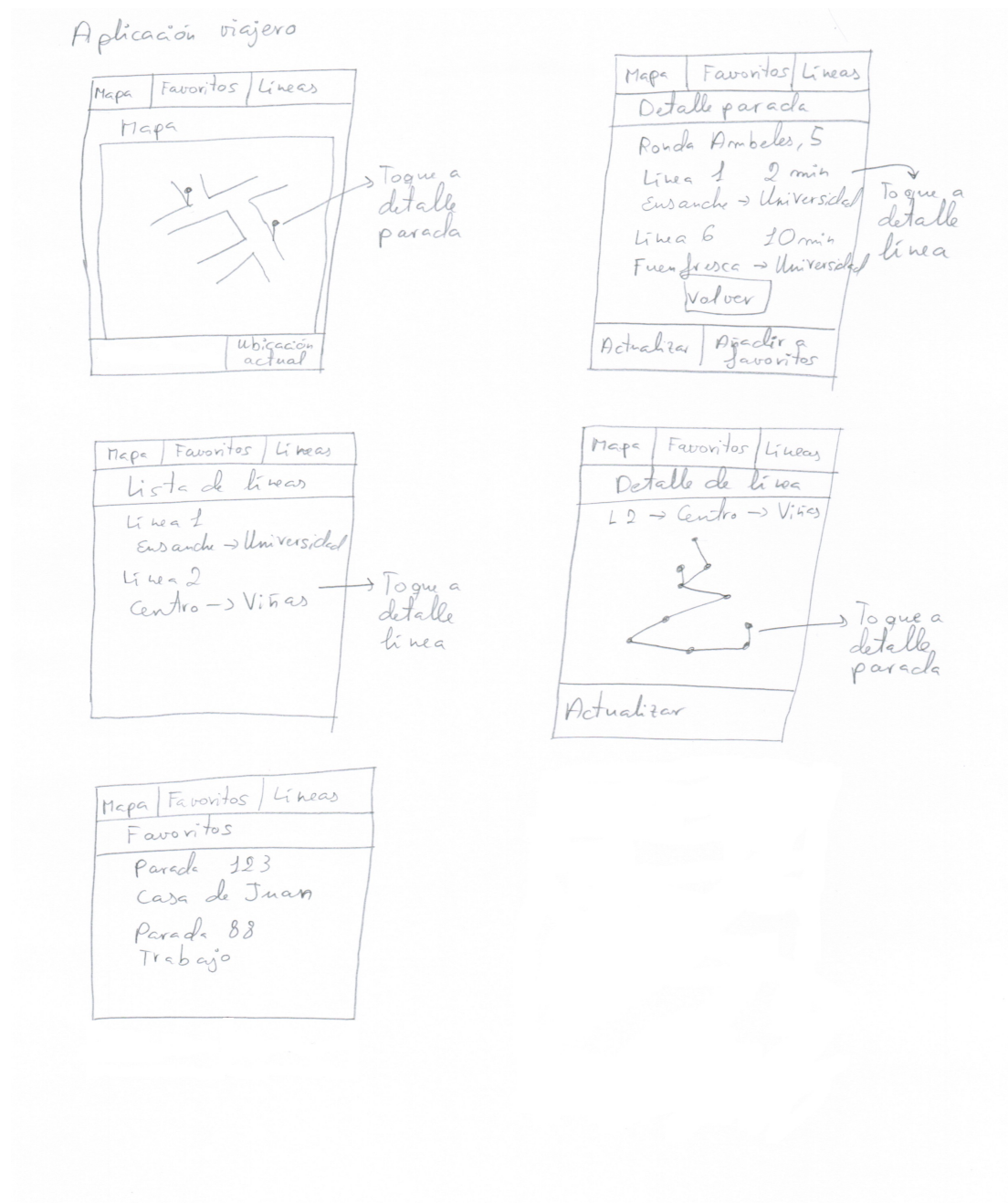
## 6. Interfaces de Usuario

### 6.1 Análisis de requisitos

#### A.- Desarrollo del concepto de producto.

##### A.1.- Escenarios / storyboards.

Se muestran en las siguientes páginas.



## **A.2.- Análisis competitivo.**

Para la realización de este apartado hemos realizado una búsqueda por Internet y hemos encontrado una aplicación similar de los autobuses de Madrid.

La interfaz de esta aplicación es bastante más simple que la nuestra, consta de 4 pestañas en su pantalla inicial, situadas en la parte superior. Estas pestañas son: urbanos, búhos, otros y favoritos.

La funcionalidad más destacable es que permite seleccionar cada una de las líneas de autobús, posteriormente el sentido y la parada, después te indica el tiempo que van a tardar en llegarán los 2 próximos autobuses.

Similitudes con nuestra aplicación:

- Permite agregar paradas de autobús a favoritos.

Diferencias con nuestra aplicación:

- No muestra mapas del lugar.
- No permite buscar posiciones dentro de los mapas.
- No informa de las paradas ni tampoco de la ubicación actual.

## **B.- Análisis de requisitos. Informe de usuarios / roles de la aplicación.**

### **B.1.1.- Posibles usuarios de la aplicación,**

El viajero podrá recibir información acerca de las líneas y las paradas, el tiempo estimado de llegada de un autobús a una parada, agregar paradas a una lista de favoritos, modificar y reordenar sus paradas favoritas.

Los viajeros tendrán que tener conceptos básicos de navegación con aplicaciones Android. Además deberá conocer la ciudad para poder localizar las paradas en un mapa o al menos la dirección de la parada de la que desea conocer la información.

### **B.1.2.- Fichas de usuarios**

En esta aplicación solo existe un rol de usuario, el conductor del autobús que como tal solo puede introducir los datos del autobús, iniciar la ruta y una vez en ruta alterar el sentido de la marcha y comunicarlo al servidor.

## C.- Tareas.

### C.1.- Especificar las tareas.

El viajero corriente podrá realizar en nuestra aplicación las siguientes tareas:

- Ver el recorrido de una línea mostrado en el mapa:
- Ver el tiempo de espera en una parada antes de que llegue el siguiente autobús:
- Añadir una parada a favoritos:
- Eliminar una parada de la lista de favoritos:
- Reordenar la lista de favoritos:
- Buscar una dirección:
- Mostrar la ubicación actual del usuario en el mapa:

### C.2.- Diagrama HTA de cada tarea.

Tareas del viajero:

- Tarea 1: Ver el recorrido de una línea mostrado en el mapa.  
Diagrama:
  1. Ver recorrido de una línea
    - 1.1. Seleccionar lista de líneas.
    - 1.2. Pulsar línea deseada.
    - 1.3. Se mostrará el recorrido de la línea.
- Tarea 2: Ver el tiempo de espera en una parada antes de que llegue el siguiente autobús.  
Diagrama:
  2. Ver el tiempo de espera
    - 2.1. Seleccionar el mapa.
    - 2.2. Pulsar la parada deseada.
    - 2.3. Se mostrará en detalle las líneas que paran en la parada anteriormente seleccionada.
- Tarea 3: Añadir una parada a favoritos.  
Diagrama:
  3. Añadir una parada a favoritos.
    - 3.1. Seleccionar el mapa.
    - 3.2. Seleccionar detalle de parada.
    - 3.3. Seleccionar Añadir a Favoritos.
- Tarea 4: Mostrar la ubicación actual del usuario en el mapa.  
Diagrama:
  4. Mostrar la ubicación actual del usuario en el mapa.
    - 4.1. Seleccionar mapa.
    - 4.2. Seleccionar ubicación actual.
    - 4.3. Se muestra en el mapa la ubicación actual del usuario.

## D.- Especificaciones

### D.1.- De usabilidad.

#### Sobre los usuarios:

- Los usuarios de aplicaciones móviles se desenvuelven en entornos donde van a estar expuestos a multitud de **estímulos simultáneos** fuera de la aplicación. La gran diferencia con respecto a entornos telemáticos tradicionales es que la atención del usuario va a estar influida en gran medida por el entorno. Por consiguiente, los productos destinados a dispositivos móviles han de tener en cuenta estas circunstancias y crear aplicaciones que sean sencillas, de navegación extremadamente simple y cuyos objetivos sean fácilmente alcanzables con la mínima carga cognitiva.
- Debido a que la mayoría de las ocasiones en las que se utiliza el dispositivo móvil van a implicar entornos cambiantes o, al menos, abiertos, un factor a tener muy en cuenta es la **luminiscencia que sufrirá la interfaz inicial**. Esta sufrirá variaciones drásticas incluso en escasos periodos de tiempo imposibilitando la fijación de la atención en tareas largas o que requieran una atención excesiva.
- En el caso de que sea necesario, aplique técnicas de **internacionalización** y localización a fin de acomodar los entornos que cree a las distintas culturas que puedan actuar potencialmente con el desarrollo que usted está llevando a cabo.
- Es crítico saber el **tipo de población** a la que el programa va ir destinado. Diversos usuarios utilizando un mismo entorno, pueden provocar grandes inconvenientes, en tanto en cuanto su manera de interactuar varía. Por ese motivo, es recomendable hacer los desarrollos de software lo más estándares posible, sin particularizar hacia poblaciones específicas.

#### Sobre la interacción Usuario – Máquina:

- Debido a que la movilidad es el aspecto que va a imperar en el desarrollo, existe un factor determinante para el terminal: su **reducido tamaño**. Desde el punto de vista de la Usabilidad ello va a generar consecuencias críticas a tener en cuenta, como puede ser el reducido tamaño de la pantalla, por ejemplo. Por ese motivo, la simplificación de la terminología, la asignación de extrema relevancia a la información icónica o la elección de iconos auto-explicativos son pilares fundamentales sobre los que planificar el diseño y estructura de la interfaz.
- Adicionalmente al punto anterior, la **interactividad** que el usuario va a tener con el terminal va a ser **reducida**, ya que, en la mayoría de los casos, lo va a sostener con una mano y con la otra va a sostener un puntero o con el propio dedo por medio del cual interactuará con la interfaz. Por ese modo, la versatilidad de movimientos va quedar reducida en gran medida.
- Otro aspecto a tener en cuenta y que también se encuentra ligado a este punto es que debido a que estamos hablando de terminales que se utilizan con una sola mano y que, generalmente, quedan alineados visualmente hacia el lado de la mano que está

sosteniendo dicho terminal, la percepción de los elementos de la interfaz será distinta si es observada por un usuario diestro o por un usuario zurdo. Especialmente, esto es crítico en usuarios **zurdos**, pues estos, asimilando la manera de coger el bastón como suelen hacerlo con los bolígrafos cuando escriben, en muchos casos adoptarán una posición de pinza, la cual, al interactuar con la interfaz y realizar desplazamientos, puede provocar frecuentes bloqueos visuales de elementos de la interfaz. La mano de este modo, taparía parte de la pantalla, dificultando el uso y la efectividad de las acciones realizadas.

- Mientras que **seleccionar opciones de la pantalla con el puntero** del ratón de un ordenador de sobremesa viene a ser algo sencillo y fácilmente adquirible como destreza, emular las mismas actuaciones en el mundo de los dispositivos móviles es algo mucho más complicado. La destreza de los usuarios a la hora de lograr un impacto con un bastón de smartphone en un determinado punto de la pantalla es ostensiblemente inferior a la que logran ejecutando la misma tarea con un puntero de ratón en un ordenador de sobremesa. Ello, queda agravado por el hecho de que muchos usuarios tienden a utilizar sustitutos del bastón si este se ha extraviado. De modo que reemplazan dicho elemento de navegación por bolígrafos, palillos, lapiceros, etc...

### **Sobre la máquina:**

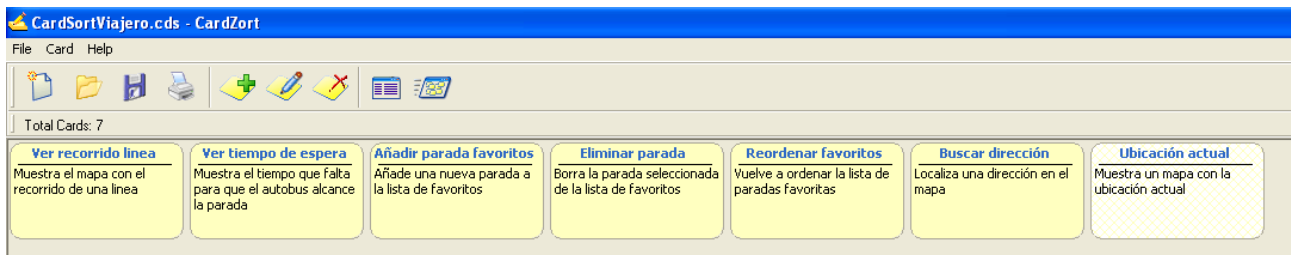
- Si bien los ordenadores personales se han convertido en herramientas cotidianas del entorno profesional y doméstico, los smartphones aún no han alcanzado ese estatus. Por ese motivo no se les ‘perdonan’ ciertos fallos que si bien están presentes en los terminales de sobremesa, pueden provocar el desuso o el abandono del smartphone. Por ello, es imprescindible desarrollar las aplicaciones y los terminales para que presenten los mínimos inconvenientes posibles.
- Minimice todo lo que pueda las interacciones que deba hacer el usuario con el terminal. Ello, se ha de hacer partiendo de la base que la pantalla es de reducidas dimensiones, que solo podrá interactuar con una mano, sosteniendo un bastón (el cual no les es familiar y, a veces, ni siquiera cómo), que lo hará en movimiento y que el entorno presentará gran cantidad de estímulos distractores. En consecuencia, las acciones, o navegaciones deben disponer de un ratio máximo de ‘clicks’ de tres o cuatro.
- Aquellas tareas no necesariamente relacionadas con la interfaz o con el producto de software en sí, han de ser igualmente minimizadas. De este modo, se ha de buscar una simplicidad casi infantil en tareas tales como establecer conexiones WIFI, sincronizar los datos e informaciones de los ordenadores de sobremesa con la información contenida en el smartphone o las actualizaciones de software requeridas por parte de los paquetes de software contenidos en los smartphones.

## E.- Prototipado.

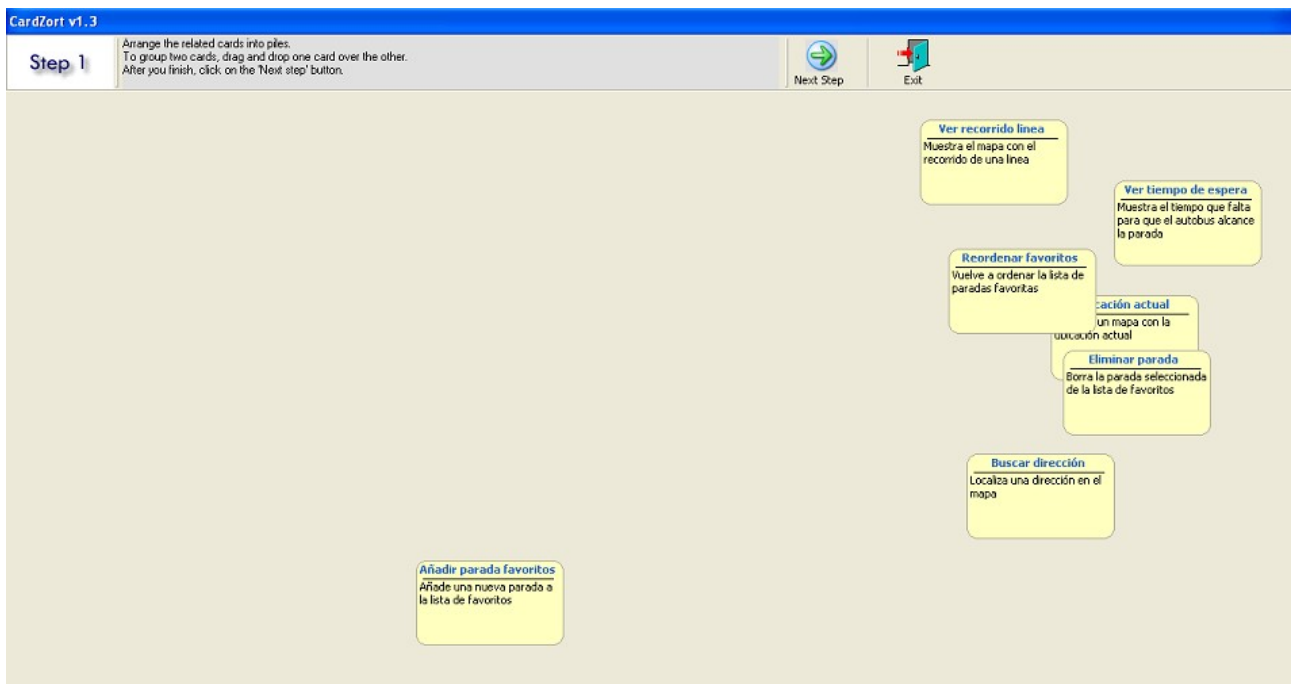
### E.1. - Card Sorting

Para la realización de este análisis hemos usado la herramienta facilitada por el profesor obteniendo las siguientes capturas de pantalla.

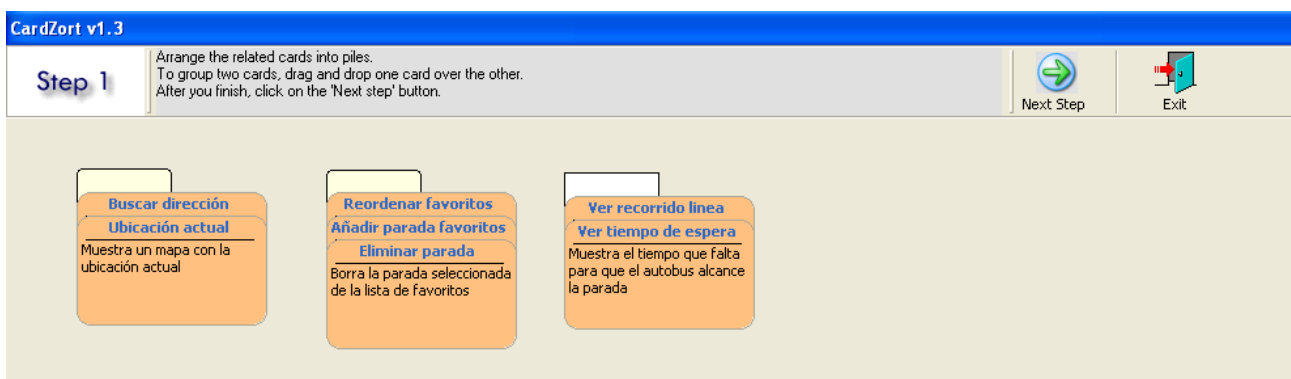
Pantalla que muestra las tarjetas creadas



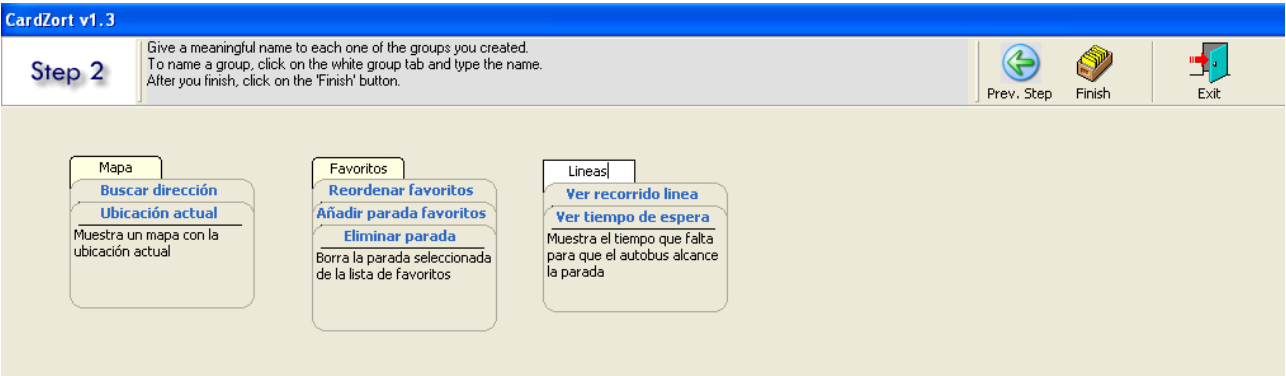
Pantalla de reordenación de tarjetas



Pantalla en la que se han agrupado ya las tarjetas



Pantalla para nombrar las agrupaciones





## E.2.- Prototipos

### Mapa



### Lista de líneas



### Detalle de línea



### Detalle de parada



## Favoritos



### *E.3.- Recorrido Pluralístico*

Tarea 1: Ver el recorrido de una línea mostrado en el mapa.

Pasos:

1. Pulsar Líneas.
2. Pulsar línea y sentido deseado.
3. Se mostrará el detalle del recorrido de la línea.

Tarea 2: Ver el tiempo de espera en una parada antes de que llegue el siguiente autobús.

Pasos:

1. Pulsar Mapa.
2. Pulsar la parada deseada.
3. Se mostrará en detalle las líneas que paran en la parada seleccionada y el tiempo que falta hasta que llegue el siguiente autobús.

Tarea 3: Añadir una parada a favoritos.

Pasos:

1. Pulsar Mapa.
2. Pulsar parada deseada para ver detalle de parada.
3. Pulsar Añadir a Favoritos.

Tarea 4: Mostrar la ubicación actual del usuario en el mapa.

Pasos:

1. Pulsar mapa.
2. Pulsar ubicación actual.
3. Se muestra en el mapa la ubicación actual del usuario.

#### **F.- Factor tecnológico.**

La aplicación será diseñada para que funcione en terminales Android, con una versión 2.1 ó superior, para ello bastará con lanzar un ejecutable .apk en cada terminal, además, los desarrolladores deberán obtener una APIKey para GoogleMaps.

## 6.2 Diseño

### 1.- Guía de estilo.

#### *Look & feel del proyecto.*

Las guías de estilo tenidas en cuenta en el desarrollo de esta aplicación se pueden encontrar en la siguiente página:

[http://developer.android.com/guide/practices/ui\\_guidelines/index.html](http://developer.android.com/guide/practices/ui_guidelines/index.html)

Dado que en las clases no se especificaban modelos de look and feel orientados a aplicaciones móviles y aún menos para aplicaciones Android hemos tenido que buscar por nuestra cuenta una serie de guías de estilo para Android que propusieron los inventores de la tecnología para que las aplicaciones de esta tecnología tuvieran unos estándares en los que basarse, es por esto que hemos elegido usar estas guías de estilo y no basarnos en las expuestas en clase.



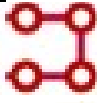




#### *Colores (paleta de colores)*

En cuanto a la paleta de colores hemos basado nuestros diseños en el negro de fondo para facilitar la visión de las pantallas incluso en ambientes con alta luminosidad dado que si el fondo es blanco el brillo impide la lectura y comprensión de los elementos de la interfaz en estos dispositivos que están diseñados para su uso en movimiento. Por otro lado para garantizar el contraste de los textos hemos usado color blanco para los textos y así garantizar que se verán bien y se distinguirán del fondo.

Además de estos dos colores para los Dialogs que describiremos posteriormente utilizamos un color gris. También usamos un color gris claro para los botones. Unas muestras de los colores son las que siguen:



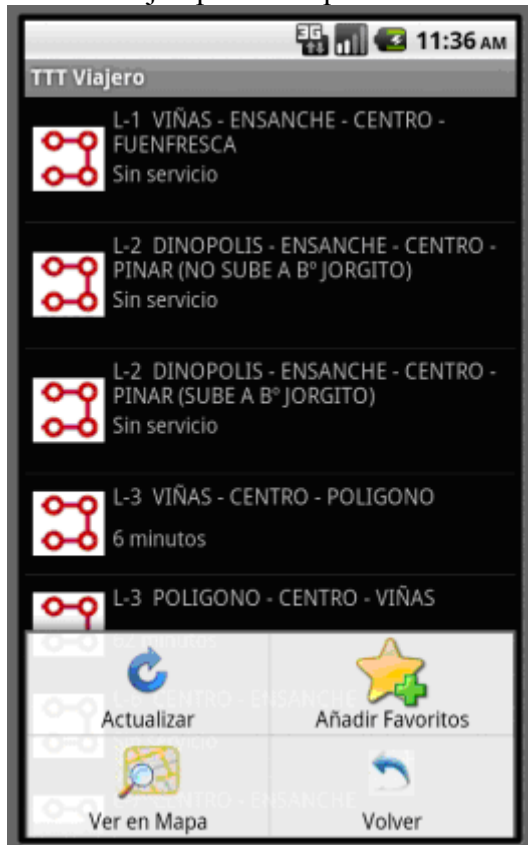
En cuanto a las metáforas graficas usadas en el caso de esta aplicación tenemos las siguientes:

	Actualizar
	Añadir a favoritos
	Linea
	Parada
	Ubicación actual
	Ver mapa
	Volver

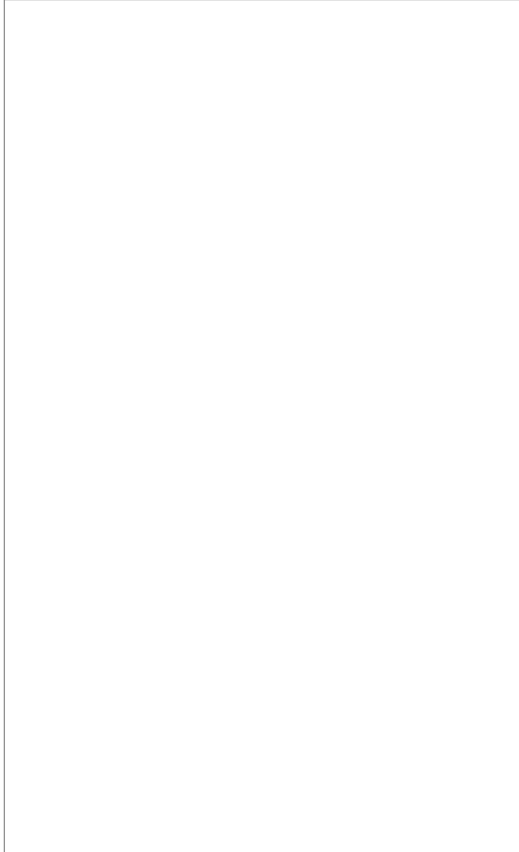
### *Disposición de menús.*

En el caso de la aplicación para el viajero tendremos en cuenta varios menús porque cada pantalla de la aplicación tiene distintas funcionalidades, la aplicación está pensada para que el usuario pueda navegar entre las diferentes pantallas con diferentes botones y que éstos estén puestos en el menú para que no resten visibilidad en la pantalla, ya que, las pantallas de los dispositivos móviles son de pequeño tamaño.

Como un ejemplo en el que se muestre un menú tenemos la siguiente pantalla:



También tenemos Dialogs en Android que son unas pantallas que indican al usuario que la aplicación está a la espera de recibir algún evento y permiten generar diálogos de progreso con el usuario. En nuestro caso tenemos ejemplos de Dialogs de progreso que se muestran al cargar la aplicación, las líneas de autobuses que recorren la ciudad o las paradas que componen una línea de autobús. A continuación se muestra un ejemplo:



### ***Formato de los listados.***

El formato de los listados que hemos utilizado se basa en un formato típico de Android que consiste en mostrar la información en elementos unos debajo de otros ocupando todo el ancho de la pantalla y permitiendo interactuar con estos elementos pulsando sobre uno de ellos de forma que toda la pantalla esté ocupada y podamos interactuar con los elementos de la lista de forma fácil y efectiva.

En nuestra aplicación hemos colocado un listado que se corresponde con la pantalla que muestra las líneas y el sentido que componen el servicio de autobuses y este listado con las líneas de transporte urbano disponibles en Teruel para que el viajero seleccione la que línea que desee y pueda ver su recorrido en el mapa pulsando en una línea, ésta se seleccionará y le dirigirá a la pantalla que muestra el recorrido de la línea.

El aspecto de esta pantalla de listado es el siguiente:



En ella podemos ver las diferentes líneas con su número, su descripción y su icono y simplemente pulsando sobre uno de ellos mostraríamos esa línea.

### *Pautas de lenguaje y diccionario propio (vocabulario específico).*

Por su parte el vocabulario empleado incluye ciertos términos técnicos como Autobús, Favoritos, Línea, Mapa, Parada, Sentido, pero que a su vez se han hecho muy comunes con el paso del tiempo entre la gente normal puesto que el autobús es un medio de transporte comúnmente utilizado por la ciudadanía.

A pesar de todo ello hemos decidido explicar brevemente todos los términos específicos usados en esta aplicación:

- **Autobús:** Se refiere al medio de transporte en si a la máquina que traslada a los pasajeros de un lugar a otro. Cada autobús tiene un número identificador así como una línea asignada durante el trayecto.
- **Línea:** Es el conjunto de paradas por las que debe pasar el autobús para recoger viajeros así como el trayecto entre estas paradas Tiene un número y dos sentidos posibles.
- **Mapa:** Es una representación gráfica de la ciudad en la que se muestran las diferentes calles de la ciudad.
- **Parada:** Se trata de cada uno de los lugares en los que hay situados una marquesina de autobús para que los viajeros lo esperen.
- **Sentido:** Se trata de cada uno de los dos trayectos que puede realizar una línea, habitualmente se corresponden con ida y vuelta.

Así pues quedan definidos los términos más específicos de esta aplicación en cuanto a su lenguaje técnico o vocablos específicos.



## 2.- Diseño del prototipo no funcional.

La siguiente pantalla muestra el mapa con las paradas a seleccionar por el viajero.



Al pulsar en el menú se mostrará un botón que nos indicará la ubicación actual en la que nos encontramos:



Al pulsar en la pestaña de favoritos mostrará las paradas que se hayan agregado al listado de paradas favoritas:



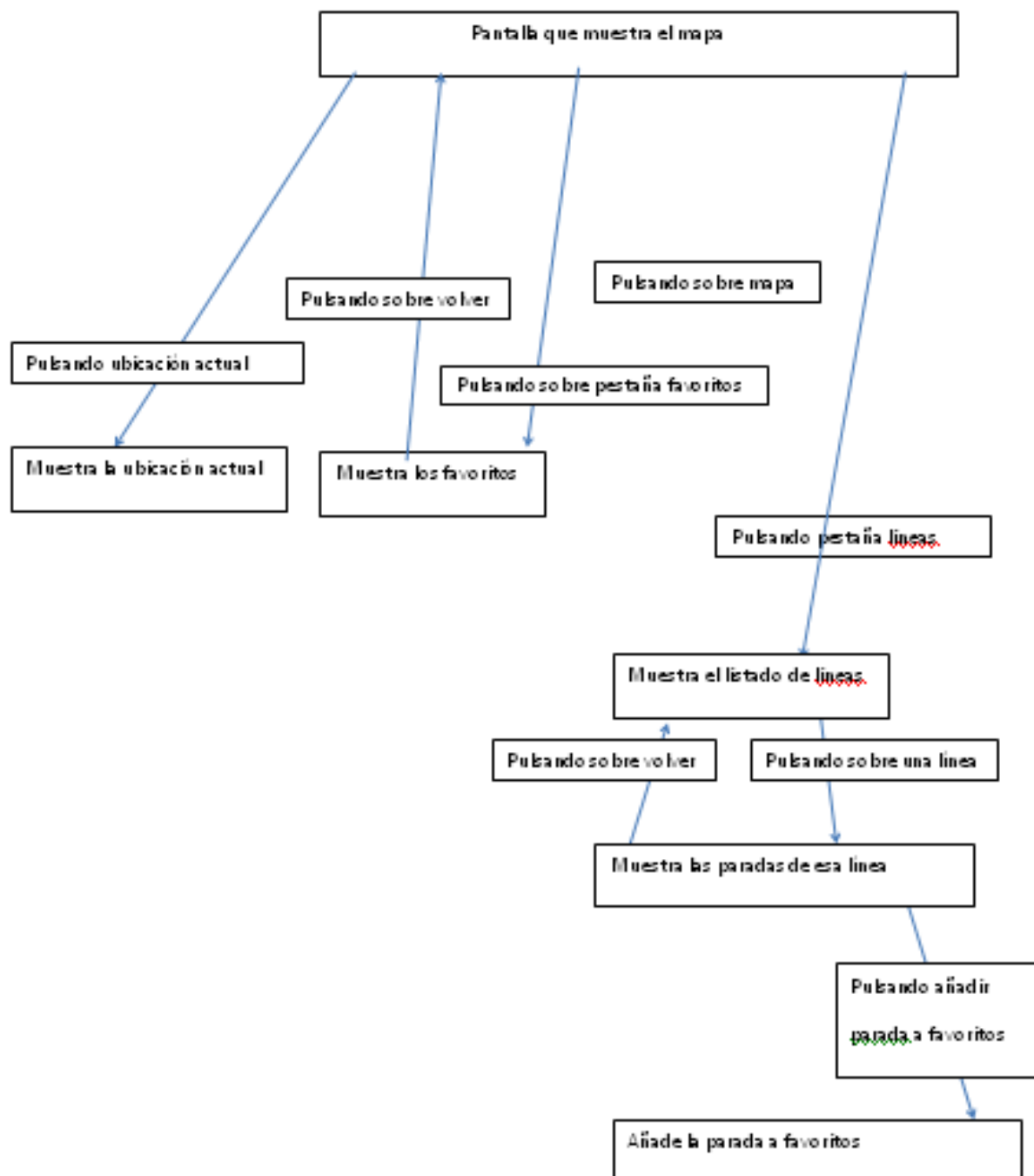
Al pulsar en la pestaña de líneas nos mostrará el listado de las líneas de autobús existentes:



Al pulsar en el menú nos aparecerán las distintas opciones como son: actualizar, añadir a favoritos, ver en el mapa y volver:



### 3.- Árbol de pantallas de nuestra aplicación.



#### 4.- Requerimientos para su uso e instalación.

Para la instalación y posterior ejecución de la aplicación se adjunta un fichero ejecutable Android con extensión .apk que contiene la aplicación además de los códigos fuentes necesarios para la aplicación.

Existen dos variantes para su ejecución:

Si se posee un terminal Android: solo se necesita descargar el archivo .apk en el terminal e instalarlo y ejecutarlo normalmente.

Si no se posee un terminal Android:

Se necesitara descargar e instalar el entorno de desarrollo eclipse de la siguiente url:

<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/heliossr2>

Se necesitara descargar e instalar el plugin de Android de la siguiente url:

<http://developer.android.com/sdk/installing.html>

Se deberá acoplar el plugin al eclipse como se enseña en el siguiente tutorial:

<http://developer.android.com/sdk/eclipse-adt.html#installing>

Se deberá abrir el proyecto en eclipse y proceder a su ejecución para lo cual habrá que definir un emulador con una versión de Android 2.1 o superior con Google APIs. (Es posible que surjan problemas a la hora de mostrar los mapas porque es necesario obtener una clave de Google para usar mapas que se encuentra en la siguiente url:  
<http://wiki.androidstartup.com/proyectos/androidnerddinner/gui/api-key> )

A la hora de la instalación solo se puede tener instalada una y no las tres debido a que hemos considerado que es mejor no tener todas las aplicaciones el mismo móvil para mayor seguridad.

## **7. VTP**

### **7.1. Introducción**

El proyecto TTT es una implementación de una aplicación para usuarios que desean conocer el tiempo que falta para la llegada de su autobús urbano y para consulta de las diferentes líneas que circulan por la ciudad. La aplicación será ejecutada en la plataforma electrónica web y Android para dispositivos móviles.

Existe una comunicación entre el usuario y el servidor a través de una conexión TCP/IP, la cual será transparente al usuario. Los autobuses serán los encargados de ir mandando al servidor su localización y toda la información necesaria para que los usuarios puedan obtener sus consultas reales. El usuario desconocerá como es el funcionamiento interno del proyecto, él únicamente accederá a nuestra aplicación y realizara diferentes consultas a un servidor sobre líneas, horarios... y el servidor será el encargado de enviar la información a los usuarios que la soliciten.

### **7.2. Primeras pruebas**

Las primeras pruebas a realizar serán las relacionadas en la conexión entre el cliente y el servidor. Comunicación entre ambos, envío de peticiones y solicitudes, gestión de las mismas y envío de la información solicitada.

Una vez se compruebe que estas primeras pruebas son correctas, se harán pruebas para comprobar el correcto funcionamiento de todos y cada uno de los casos de uso de nuestras diferentes aplicaciones.

### **7.3. Planificación de pruebas**

El desarrollo de las diferentes pruebas que se llevarán a cabo para nuestra proyecto comenzarán a finales de Marzo de 2011, y se irán desarrollando a medida que haya código funcional. Finalmente se hará un conjunto de pruebas finales las cuales comenzarán según la planificación la segunda semana de Abril. Las pruebas se realizarán en la máquina virtual de Android ya que ninguno de los integrantes del grupo disponemos de un dispositivo con dicho sistema operativo.

## 7.4. Pruebas que se van a realizar para la aplicación viajero

Las diferentes pruebas que se van a llevar a cabo para la aplicación viajero se han obtenido realizando un estudio de todos los requisitos de la aplicación viajero. En este documento se refleja para un determinado requisito las pruebas que se deberán realizar para comprobar el correcto funcionamiento de cada uno de los diferentes prototipos.

### 4.2.1 General

**REQ\_VIA\_01\_01:** Cargar las líneas en pantalla.

**REQ\_VIA\_01\_01:** Si el número de líneas es superior a la pantalla se debe obtener una barra de desplazamiento.

**REQ\_VIA\_01\_01:** Selección de una línea.

### 4.2.3 Vista detalle de línea

**REQ\_VIA\_02\_02:** Colocar parada en el mapa.

**REQ\_VIA\_02\_02:** Pulsar en parada para ver detalle de la parada.

### 4.2.4 Vista detalle de parada

**REQ\_VIA\_03\_01:** Actualizar la información de las líneas con su tiempo.

**REQ\_VIA\_03\_02:** Pulsar en una línea y ver detalle de la línea.

**REQ\_VIA\_03\_03:** Introducir favoritos a lista.

### 4.2.5 Vista de mapa

**REQ\_VIA\_04\_01:** Mostrar paradas en mapa

**REQ\_VIA\_04\_03:** Centrar mapa en ubicación actual.

### 4.2.6 Favoritos

**REQ\_VIA\_05\_04:** Al pulsar en una parada de la lista de favoritos se ve su detalle de parada.

## 7.5. Necesidades para realizar las pruebas

Para realizar las diferentes pruebas se necesitara hacer un hipotético entorno como el que sería usado por los usuarios de nuestro sistema. Para ello usaremos el SDK de la máquina virtual de Android sobre un entorno de desarrollo Eclipse.