

# APP TrailSSport



**Alejandro Aguilar Baena.**

Entrega Final.

## **Consultores**

Helena Boltà Torrell.

Jordi Almirall López.

## Índice.

<b>1.</b>	<b>INTRODUCCIÓN.....</b>	<b>3</b>
1.1.	OBJETIVOS.....	3
1.2.	LISTADO DE FUNCIONALIDADES.....	4
1.3.	RIESGOS DEL PROYECTO. ....	4
1.4.	¿POR QUÉ ESTA ELECCIÓN?.....	5
1.5.	CALENDARIO DEL PROYECTO. ....	6
1.6.	TECNOLOGÍA Y HERRAMIENTAS ESCOGIDAS.....	7
<b>2.</b>	<b>DESARROLLO CENTRADO EN EL USUARIO.....</b>	<b>8</b>
2.1.	USUARIOS Y CONTEXTO DE USO. ETAPAS DE ANÁLISIS.....	8
2.1.1.	<b>Métodos de Indagación.</b> .....	8
2.1.1.1.	<i>Desarrollo, resultados y conclusiones obtenidas de los métodos utilizados.</i> .....	9
a)	<i>Observación e investigación contextual.</i> .....	9
b)	<i>Encuestas.</i> .....	9
	<i>Gráfica 1 - Porcentaje de running.</i> .....	10
	<i>Gráfica 2 - Porcentaje de duatlón.</i> .....	11
	<i>Gráfica 3 - Porcentaje de triatlón.</i> .....	11
	<i>Gráfica 4 - Porcentaje de bike.</i> .....	11
	<i>Gráfica 5 - Porcentaje de Asistencia</i> .....	12
	<i>Gráfica 6 - Porcentaje consideran útil APP.</i> .....	12
c)	<i>Análisis competitivo o comparativo.</i> .....	13
	<i>Ilustración -1 APP CarrerasPopulares</i> .....	13
	<i>Ilustración -2 APP Madrid Running</i> .....	14
	<i>Ilustración -3 APP Todo Carreras.</i> .....	15
	<i>Ilustración -4 APP CarrerasPorMontaña</i> .....	15
d)	<i>Método de diario.</i> .....	16
2.2.	<b>DISEÑO CONCEPTUAL. ETAPA DE DISEÑO.</b> .....	17
2.2.1.	<b>Escenarios de uso.</b> .....	17
2.2.1.1.	<b>Los perfiles de Usuarios.</b> .....	17
2.2.1.2.	<b>Personajes o personas.</b> .....	18
	<i>Tabla 1 - Primera Persona Ficticia(Deportista por afición).</i> .....	19
	<i>Tabla 2 - Segunda Persona Ficticia (Deportista competitivo).</i> .....	19
	<i>Tabla 3 - Tercera Persona Ficticia (Deportista muy competitivo).</i> .....	19
2.2.1.3.	<b>Escenarios</b> .....	20
	<i>Tabla 4 - Primer Escenario.</i> .....	20
	<i>Tabla 5 - Segundo Escenario.</i> .....	20
	<i>Tabla 6 - Tercer Escenario.</i> .....	20
2.2.2.	<b>Flujos de interacción.</b> .....	21
2.3.	<b>DISEÑO DE PROTOTIPOS. ETAPA DE TRANSICIÓN ENTRE LA DE DISEÑO Y LA DE EVALUACIÓN.</b> .....	23
2.3.1.	<b>SKETCHES.</b> .....	23
2.3.2.	<b>Prototipado.</b> .....	23
	<i>Pantalla 5 - Pantalla inicial.</i> .....	23
	<i>Pantalla 6 - Selección de deportes.</i> .....	24
	<i>Pantalla 7 - Subcategorías de running y bike.</i> .....	25
	<i>Pantalla -8 - Ubicación y provincias.</i> .....	25
	<i>Pantalla 9 - Selección de fechas y tipo de días.</i> .....	26
	<i>Pantalla 10 - Configurar Notificaciones.</i> .....	26

Pantalla 11 – Ver listado de competiciones.....	27
Pantalla 12 – Ver competiciones en mapa.....	28
Pantalla 13 –Ver detalle de una competición.....	29
Pantalla 14 – Acceso normal.....	30
Pantalla 15 - Prueba de imagen como fondo.....	31
<b>2.4.    EVALUACIÓN. ETAPA DE EVALUACIÓN.....</b>	<b>31</b>
2.4.1.    Opiniones/Test de Usuarios en cuanto al diseño.....	32
<b>3.    DESARROLLO DEL PROYECTO.....</b>	<b>33</b>
<b>3.1.    IMPLEMENTACIÓN.....</b>	<b>33</b>
3.1.1.    Activity y/o clases.....	34
3.1.1.1.    Clase ActivityInicio.....	34
3.1.1.2.    Clase ActivitySelectFechas.....	34
3.1.1.3.    Clase ActivitySelectSport.....	35
3.1.1.4.    Clase ActivitySelectUbi.....	35
3.1.1.5.    Clase ActivitySelectNotif.....	36
3.1.1.6.    Clase ActivityLista.....	36
3.1.1.7.    Clase ActivityActividad.....	37
3.1.1.8.    Clase ActivityMapa.....	39
3.1.1.9.    Clase CustomArrayAdapter y Clase ListaCompe.....	40
3.1.1.10.    Clase TrailsSportCursorAdapter.....	41
3.1.1.11.    Clase TrailsSportDbAdapter.....	41
3.1.1.12.    Clase TrailsSportDbHelper.....	42
3.1.1.13.    Clase actividad.....	42
3.1.1.14.    Clase MisFiltros.....	43
3.1.1.15.    Clase selectSport.....	44
3.1.1.16.    Clase utilidades.....	44
3.1.2.    SharedPreferences.....	44
3.1.3.    Animaciones o transiciones entre pantallas.....	46
<b>3.2.    PROTOTIPOS FINALES DE INTERFAZ DE USUARIO.....</b>	<b>48</b>
3.2.1.    Pantalla de Inicio.....	48
3.2.2.    Pantalla de selección de deportes(o tipos de competiciones).....	48
3.2.3.    Pantalla de selección de Provincia.....	49
3.2.4.    Pantalla de selección de fechas.....	50
3.2.5.    Pantalla de selección de notificaciones.....	50
3.2.6.    Pantalla lista competiciones.....	51
3.2.7.    Pantalla detalle competición.....	51
<b>4.    CONCLUSIONES Y FUTURO DE LA APP.....</b>	<b>52</b>
<b>5.    ANEXOS.....</b>	<b>53</b>
<b>6.    BIBLIOGRAFÍA.....</b>	<b>53</b>
LIBROS.....	53
RECURSOS WEB.....	53

## 1. Introducción.

Esta entrega se corresponde con la entrega final del proyecto final de carrera, con este documento pretendo dejar constancia de todo lo que he aprendido hasta llegar a esta entrega, todos los problemas con los que me he encontrado, como los he superado, además de tener claro las mejoras futuras y los aspectos que me gustaría poder aprender sobre este campo de desarrollo de aplicaciones móviles con android, ya que ha despertado en mi un gran interés.

### 1.1. Objetivos.

El objetivo del Trabajo fin de Carrera (TFC) es poner en práctica todos los conocimientos adquiridos en la Ingeniería Técnica Informática de Gestión y adentrarme en el desarrollo de aplicaciones móviles con Android, ya que es un sector que está en estos momentos en auge, e incluso dentro de la empresa que trabajo podría ser bastante útil ampliar mis conocimientos en estos lenguajes.

El objetivo principal de esta App que vamos a desarrollar, es centralizar en una aplicación todas las competiciones deportivas, como carreras de running, medias maratones, maratones, Cross, duatlones, triatlones, carreras de Bicis,... esto permitirá a los deportistas estar informados en todo momento de las competiciones(o eventos) disponibles, y además a organizadores de eventos dar a conocer los mismos.

Para ello, usaremos las características que ofrecen hoy en días los dispositivos móviles, localización por GPS, visualización de mapas y rutas y recepción de notificaciones instantáneas.

Creo que además de lo descrito anteriormente, algunos objetivos que pretendo conseguir con este proyecto son los siguientes:

- Aprovechar los servicios externos que ofrece Google Maps.
- Cumplir los plazos estimados.
- Adquirir nuevos conocimientos de Java y de Android.
- Poder tener en mi móvil una aplicación desarrollada por mí.

Quiero añadir en este apartado que debido a mis conocimientos actuales sobre este tema, puede darse el caso, una vez entrado en materia, conozca o aprenda ciertos aspectos dentro del desarrollo de aplicaciones móviles que me lleven a realizar algún cambio en el proyecto, siempre y cuando sea justificado y explicando evidentemente el porqué se ha realizado dicho cambio.

**Tengo que añadir en este punto, que todos estos objetivos mencionados se han alcanzado a excepción de aprovechar los servicios externos que ofrece Google Maps, ya que debido a mis conocimientos y al poco tiempo que disponíamos para realizar la APP, ha sido imposible terminar de abordar las funciones de google Maps dentro de la aplicación. Es un tema que queda pendiente para este verano.**

## 1.2. Listado de funcionalidades.

Las funcionalidades principales de las que consta la aplicación son las siguientes:

- Búsqueda de competiciones por categorías:
  - o Deportes.
    - Running.
    - Duatlón.
    - Triatlón.
    - Mountan Bike.
    - Natación
  - o Fechas.
    - Selección de rango de fecha: Desde dd/mm/aaaa hasta dd/mm/aaaa.
  - o Provincias.
    - Selección de provincia.

La principal función de la App es poder realizar búsqueda por tipo de deportes (también llamado categorías) de distintas competiciones, por provincia y por fechas.

- Visualización de las competiciones en un listado y acceso a cada una de ellas con sus correspondientes datos.

## 1.3. Riesgos del proyecto.

En este apartado vamos a recordar los riesgos del proyecto mencionados en la las entregas anteriores y que ha ocurrido con ellos.

- Uno de los riesgos principales es la inexperiencia y el desconocimiento en desarrollo de aplicaciones móviles. Soy analista programador Oracle, llevo trabajando con esta tecnología unos siete años, en ocasiones he trabajado con algo de Java, pero en menor medida, por lo que este proyecto es todo un reto para mí. → En cuanto a este riesgo creo que he aprendido bastante de Java y Android, estoy bastante contento por todos los conocimientos que he adquirido, aunque me hubiera gustado poder aprender más de lo que he aprendido.
- Otro riesgo importante, y el que más me preocupa es el mantenimiento de eventos por parte de las organizaciones de este tipo de eventos. En principio los datos los voy a introducir yo, si el desarrollo cumple las fechas y considero que pueda abordar este punto, intentaré realizar la mejora o evolución del proyecto para que los organizadores puedan crear los eventos dentro de la aplicación, hablando en términos más técnicos, en principio los datos estarán en local cuando se instale la aplicación, no habrá servidor en el que actualizar los eventos en principio. → Como el desarrollo ha ocupado más tiempo del que creía ha sido imposible abordar algunos puntos, por lo que me he visto obligado a utilizar una base de datos que se crea dentro de una clase y los datos se insertan dentro de la misma. Tengo pensado este verano hacer esto bien, crear una base de datos y alojarla en algún sitio e intentar crear un web servid que ataque a dicha base de datos.

Sigo pensando como dije en las entregas anteriores que la idea en cuanto a datos, es que una vez que la aplicación se dé a conocer, los mismos organizadores de eventos sean los que informen de estos, es más una posible mejora para el proyecto sería dar acceso a estos organizadores para que puedan introducir ellos mismos los eventos, y por tanto contar con un servidor para ello.

- Al no comenzar constando de servidor, el objetivo de las notificaciones, puede ser que no sea factible, ya que al estar los datos de los eventos en local no fuese del todo factible. Aunque en principio pretendo y espero poder abordar este objetivo dentro de la planificación y del desarrollo. Con ello añadiría a la aplicación la posibilidad de disponer de un sistema de notificaciones, gestionar un servicio de notificaciones que realizara peticiones PULL cada cierto tiempo y comprobará si existen nuevos eventos que encajen con el perfil del usuario. Estas notificaciones de eventos se realizarán según el usuario tenga configurado su perfil por provincia, deporte, distancia.... → En cuanto a este tema he creído conveniente no abordarlo porque al tener los datos en local no tenía mucho sentido, he preferido que lo que he desarrollado este correcto y sin errores a intentar seguir avanzo.

#### **1.4. ¿Por qué esta elección?**

He elegido este tema para desarrollar una aplicación, porque desde hace unos años me adentre en el mundo de la bicicleta, comencé con carreras de bicicleta, a los dos años aproximadamente empecé a apuntarme a carreras de running, y este año he comenzado a realizar duatlones. En varias ocasiones he comentado con algún amigo “que raro que no exista una APP para realizar estas búsquedas de competiciones”, se de antemano que a veces es complicado encontrar las competiciones de este tipo centralizadas en una web o un una APP.

Como este tema es un tema que me apasiona, pensé que me motivaría lo suficiente para poder adentrarme en este mundo. De hecho aunque a la aplicación desarrollada le faltan algunas cosas para poder subirla al Market de android, más que nada por ser una base de datos en local, yo personalmente la estoy usando, ya que he insertado las competiciones actuales de la provincia en la que resido.

El nombre elegido es TrailSSport, que traducido al castellano viene a significar algo así como pruebas deportivas, me pareció un nombre atractivo.

### 1.5. Calendario del Proyecto.

El desarrollo del proyecto se ha hecho utilizando el ciclo en cascada. Dentro del marco del TFC, una vez establecido el plan de trabajo, se realizó una planificación con objetivos y requisitos, la cual ayudo a forjar su especificación. Después se ha hecho el diseño, tratamiento de excepciones, la implementación, y el testing o pruebas. Y por último se ha realizado la memoria y la presentación virtual.

En la siguiente tabla presentamos el calendario de trabajo del proyecto que se planifico en la primera entrega.

<u>Tarea</u>	<u>Nombre de tarea</u>	<u>Comienzo</u>	<u>Fin</u>	<u>Días</u>
1	Plan de Trabajo	01/03/2014	10/03/2014	9 días
2	Planificación	10/03/2014	15/03/2014	5 días
3	Especificación	15/03/2014	22/03/2014	7 días
4	Diseño técnico	22/03/2014	08/04/2014	17 días
5	Tratamiento de excepciones	08/04/2014	<b>16/04/2014</b>	8 días
6	Implementación	16/04/2014	<b>24/05/2014</b>	38 días
7	Testing	24/05/2014	28/05/2014	4 días
8	Informe Final	28/05/2014	<b>11/06/2014</b>	14 días

Esta es la planificación del proyecto, en la que tuvimos en cuenta las fechas clave de cada fase, e intentamos hacer coincidir con las entregas de las prácticas.

En la siguiente página podremos observar que este calendario a sufrido algunas modificaciones, ya que en alguna de las primeras etapas, se ha ido ganando algo de tiempo, siendo aprovechado para abordar otras etapas de la planificación del proyecto.

En la siguiente tabla presentamos el calendario de trabajo resultante del trabajo que hemos llevado a cabo:

<b>Tarea</b>	<b>Nombre de tarea</b>	<b>Comienzo</b>	<b>Fin</b>	<b>Días</b>
1	Plan de Trabajo	02/03/2014	10/03/2014	8 días
2	Planificación	10/03/2014	15/03/2014	5 días
3	Especificación	15/03/2014	22/03/2014	7 días
4	Diseño técnico	22/03/2014	01/04/2014	10 días
5	Tratamiento de excepciones	01/04/2014	08/04/2014	7 días
6	Implementación	08/04/2014	20/05/2014	42 días
7	Testing	20/05/2014	24/05/2014	4 días
8	Informe Final	24/05/2014	06/06/2014	13 días

### 1.6. Tecnología y herramientas escogidas.

- Lenguaje de programación Java y Android SDK Manager.
- Como entorno de desarrollo, Android SDK que cuenta ya con las APIs y librerías necesarias, e incluye una versión de Eclipse con Android Developers Tools V22.3.0 (web de descarga <http://developer.android.com/sdk/index.html#download>).
- Java SE Development Kit 7 u 54 (web de descarga <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html?ssSourceSiteId=otnes>)
- Para diseñar los prototipos hemos utilizado una aplicación online, <https://www.fluidui.com/>, aunque después los he tenido que hacer de nuevo en eclipse porque la versión online gratuita no permitía exportar/importar.

- Para el diseño de diagramas hemos utilizado EDGE Diagrammer Diagnostic.
- No he llegado a utilizar la API de Google, porque no se ha abordado esta mejora.
- Para la gestión de la base de datos hemos trabajado con SQLite, creando una base de datos desde una clase Java, e insertando los registros desde la misma clase.
- Microsoft Office para creación de documentos, presentación y diagramas.
- Portátil Toshiba Satélite L655D con Windows 7 64 bits.
- Móvil para realizar pruebas, Samsung Galaxy SIII.
- Para la grabación de video he utilizado una GoPro Hero 3 Black Edition.

## 2. Desarrollo Centrado en el Usuario.

### 2.1. Usuarios y contexto de uso. Etapas de análisis.

No podemos diseñar o analizar un producto sin antes analizar a los usuarios que van a realizar el uso del mismo, es importante que los perfiles este acorde con el uso que se les va dar. Por este motivo esta primera etapa es muy importante ya que vamos a usar los métodos de indagación convenientes para realizar este análisis al detalle.

#### 2.1.1. Métodos de Indagación.

Gracias a estos métodos conoceremos a los usuarios, entornos, intereses, manejo... En esta fase hemos contado con los siguientes métodos de investigación:

- **Observación e investigación contextual:** consiste en la observación de los usuarios en su entorno habitual. Es una de las técnicas más útiles para conocer de manera objetiva qué hacen los usuarios, y en qué condiciones lo hacen. La información que se obtiene puede servir tanto para determinar sus comportamientos como para evaluar su usabilidad.
- **Entrevistas en profundidad:** El diálogo con usuarios potenciales puede proporcionar información de las necesidades que debe cubrir la aplicación y de la experiencia con otros. No hay un guión establecido y se harán con usuarios que hayan usado otras guías o no.
- **Análisis competitivo o comparativo:** El análisis competitivo o análisis comparativo (benchmarking) consiste en analizar productos que son similares y/o que compiten con el sistema interactivo que se está diseñando. Para conocer las expectativas de los usuarios (dado que puede que ya conozcan estos productos), entender las tendencias del mercado, aprender de los errores pero también de lo que funciona, conocer las funcionalidades básicas o comunes, estudiar las interfaces, etc.
- **Método de diario:** el método de diario hace que sea el mismo participante quien recopila información sobre las actividades que se pretenden investigar. Durante un plazo de tiempo, es quien registra sus acciones, comportamientos y sensaciones.

### **2.1.1.1. Desarrollo, resultados y conclusiones obtenidas de los métodos utilizados.**

#### **a) Observación e investigación contextual.**

Como actualmente soy aficionado a este tipo de eventos, suelo observar en las competiciones y en algunos entrenamientos con amigos, que suele ser difícil encontrar información sobre eventos o actividades de este tipo. Creo que este método es bastante acertado debido a este motivo.

En muchas ocasiones se suele preguntar entre los asistentes de estas competiciones, por futuras competiciones, si asistirán unos u otros a las mismas, y siempre se suele terminar con un “Ya nos veremos en la próxima”.

Yo mismo, la mayoría de las veces suelo preguntar al resto de corredores si conocen próximos acontecimientos y muchas veces es así como nos vamos enterando de otras carreras.

Otro método de informar de nuevas competiciones por parte de los organizadores es informar de ellas a través de folletos informativos en competiciones celebradas con anterioridad.

Y claro está, como hoy en día es típico en muchos aspectos, buscar por internet desde una tableta, un móvil o un ordenador es lo más común, aunque muchas veces es lo más pesado y lo que más tiempo suele quitar. Debido a que hay pocas páginas en las que esta información sea precisa, concreta y certera.

#### **b) Encuestas.**

Para encontrar los perfiles de usuario de este tipo de aplicaciones podemos recurrir a efectuar un estudio propio (encuesta) o bien basarse en estudios previos ya realizados. Aunque en nuestro caso creo que los usuarios están más que claros, ya que solo vamos a disponer de deportistas y en un segundo desarrollo tendremos informadores, pero vamos a usar la encuesta para determinar si hay usuarios que se correspondan con una APP de este tipo, si les sería útil, si la llegarían a usar y claro esta nos vendrá muy bien para determinar que deportes son los más destacados en las competiciones en las que suele participar los usuarios.

Para realizar las encuestas, debido a su facilidad de uso y naturaleza multiplataforma, me he decidido por utilizar *Formularios de Google Drive*, el formulario utilizado para la encuesta se puede encontrar en:

[https://docs.google.com/forms/d/1g\\_d0IMqSOI2RpL9ANRJDalwq2cOS4-SoVC2De3w5M6M/viewform](https://docs.google.com/forms/d/1g_d0IMqSOI2RpL9ANRJDalwq2cOS4-SoVC2De3w5M6M/viewform)

La mayoría de los que han contestado esta encuesta son conocidos míos o amigos de amigos, conocidos de conocidos que suelen asistir a eventos de este tipo. Hay algunos que no han asistido nunca a competiciones, realmente estos no nos interesan mucho en este sentido.

**Las preguntas planteadas en la encuesta han sido las siguientes:**

- ¿Te gusta el deporte?
- ¿Practicas deporte?
- ¿Con qué frecuencia?
- ¿Sueles asistir a carreras running, duatlones, triatlones, bicicletas, otros?
- ¿Con qué frecuencia?
- ¿De dónde obtienes la información de estos eventos?
- ¿Sería útil una aplicación móvil que te informe de las carreras que te interesan?
- Nombre y apellidos.
- Correo electrónico.
- Edad.

Algunas preguntas como ¿te gusta el deporte?¿practicas deporte?¿con que frecuencia? Y ¿si suelen asistir a carreras y cuáles? Han sido meramente informativas para saber si son personas a las que me interesaba estudiar o no porque son personas a las que pudiera interesarle la aplicación, me refiero a que practiquen deporte y suelen asistir a competiciones.

En realidad esta última pregunta me interesaba, ¿Sueles asistir a carreras running duatlones, triatlones, bicicletas, otros? Ya que con estas respuestas puedo hacer una estimación de los deportes que tengo que tener disponibles en la APP para buscar competiciones.

Antes de nada quiero añadir que no he hecho participe a muchas personas de la encuesta, porque al descubrir que es una idea nueva, ya que no existe ninguna APP en el Market dedicado a esto, he preferido no airear mucho la idea por miedo a que alguien se adelante y cree una APP de este tipo, robándome la idea.

Gracias a esta pregunta, he llegado a la conclusión de que como mínimo vamos a disponer de los deportes inicialmente pensados, ya que ninguno de los encuestados ha contestado que asista a competiciones de otros deportes.

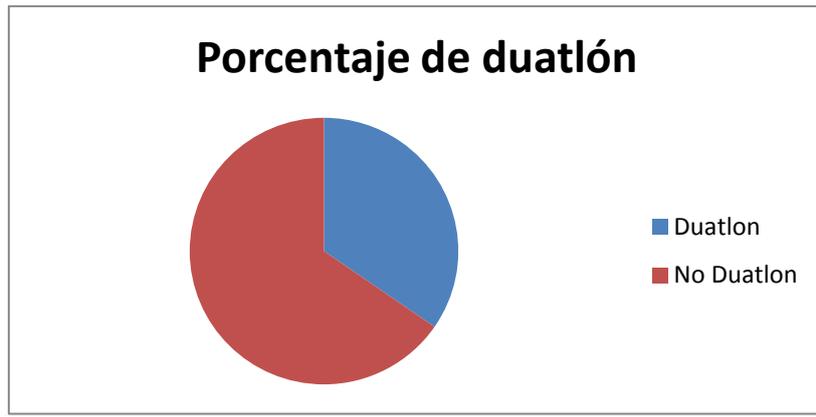
La estadística de cada deporte como competición a la que asisten los encuestados es la siguiente.

- Running: Veinte de los veintiséis encuestados practican running.



Gráfica 1 - Porcentaje de running.

- Duatlón: Nueve de los veintiséis encuestados practican duatlón.



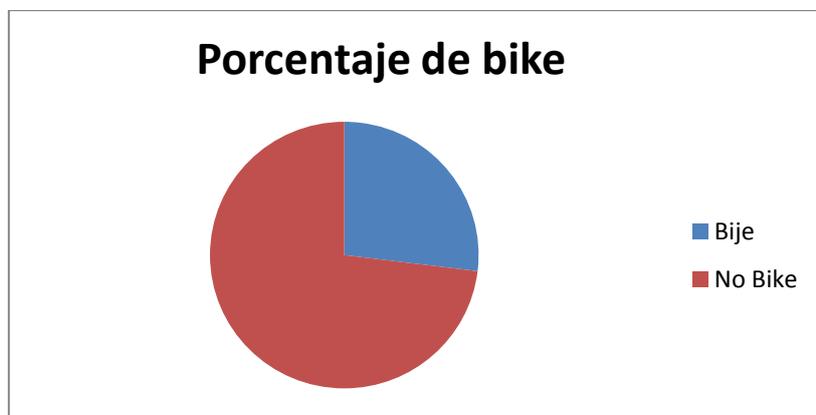
Gráfica 2 - Porcentaje de duatlón.

- Triatlón: Tres de los veintiséis encuestados practican triatlón.



Gráfica 3 - Porcentaje de triatlón.

- Bicicleta: Siete de los veintiséis encuestados practican ciclismo.



Gráfica 4 - Porcentaje de bike.

Por lo que debido a esto, se van a considerar estos deportes inicialmente, o por lo menos son los indispensables.

La pregunta de con qué frecuencia asisten a carreras es interesante para saber si realmente son personas que usarían mucho o no la aplicación.



Gráfica 5 - Porcentaje de Asistencia

Como vemos la mayoría de ellos señalan que suelen asistir a una cada dos meses, siguiéndoles los que asisten a una al mes. Claro está que hay que tener en cuenta que asistir a una competición cada mes o cada dos meses implica estar pendiente de dichas competiciones todos los meses, por lo que creemos que puede ser una APP útil y muy utilizada.

La pregunta de “¿De dónde obtienes la información de estos eventos?”, nos confirma que nadie utiliza una aplicación de este tipo, realmente porque no existe ya que en la siguiente pregunta de “¿Sería útil una aplicación móvil que te informe de las carreras que te interesan?” todos contestan que “Si”. Además un pequeño porcentaje indica que es difícil de encontrar esta información y el resto se limita a hacer búsquedas bastante tediosas por internet.

La pregunta de “¿Sería útil una aplicación móvil que te informe de las carreras que te interesan?” es para dejar claro que no hay ninguna APP con este objetivo.



Gráfica 6 - Porcentaje consideran útil APP.

La pregunta de sería útil que una aplicación que te informe de las carreras que te interesen, ha sido decisiva para conocer que al 100% le sería útil.

La edad la hemos solicitado, por tenerla, ya que en mi caso no es determinante para nuestro análisis. El nombre, apellidos y correo electrónico lo requerimos por si necesitará realizar encuestas nuevas a estos usuarios, e incluso para informarles de que la APP esté disponible cuando lo esté.

### c) Análisis competitivo o comparativo.

En este sentido creo que jugamos un papel importante en este sector, ya que no he conseguido encontrar alguna APP que cumpla con todos estos objetivos.

- Hay infinidad de aplicaciones para deportistas, por ejemplo para medir distancias y tiempos de los entrenamientos (tracks), mejorar entrenamientos y poder estudiarlos.
  - o **Endomondo Sport**, cuenta con versión gratuita y versión de pago ampliando funcionalidades.
  - o **RunKeeper** es totalmente gratuita.
  - o **Runtastic** también cuenta con versión gratuita que está limitada y versión de pago ampliando funcionalidades.
  - o **Strava** cuenta con dos aplicaciones una para runners "**Strava Run**" y otra para ciclista "**Strava Cycling**", ambas aplicaciones totalmente gratuitas.

La mayoría de estas APP disponen de información del clima que hace en el momento de realizar el ejercicio, modalidades deportivas, compartir entrenamiento en redes sociales y otras muchas más opciones de uso.

Todas estas aplicaciones tienen mucha competitividad y los usuarios suelen cambiar de unas a otras con normalidad.

- Pero en cambio si hacemos un estudio de aplicaciones para buscar competiciones hay pocas que cumplan con todo los requisitos que queremos que cumpla la nuestra, todas las estudiadas son gratis, las más destacadas son las siguientes:
  - o **Carreras Populares**: Esta aplicación es la versión para móvil de [www.carreraspopulares.com](http://www.carreraspopulares.com), el mayor calendario de carreras populares de running, montaña y MTB en internet.



[Ilustración -1 APP CarrerasPopulares](#)

Esta aplicación recopila carreras de running, montaña y BTT, a diferencia de la que estoy diseñando la nuestra cuenta con distintas categorías para running y bicicleta, además de incluir duatlones y triatlones.

Las opiniones que vienen en el Market indican que no tiene botón salir, en algún comentario se indica que es fea y poco intuitiva, que faltan carreras.

Me he instalado la APP en el móvil para probarla y solo permite seleccionar provincias, y te aparece un listado con todas las competiciones existentes, no se permite ordenar por fecha, provincia o actividad como en la que estamos desarrollando.

A raíz de analizar esta APP, tengo algunos objetivos que tiene que cumplir mi APP, para poder superarla con creces:

- Ser bonita y atractiva a la vista, fácil de usar e intuitiva.
  - Contar con la mayor recopilación de competiciones, actualizarlas constantemente, y si fuese posible contactar con los organizadores para que nuestra APP sea la primera en tener la información.
  - Esta APP tiene que tener un bonito y simple botón salir.
  - Para duatletas y triatletas está claro que es fundamental la APP que tengo entre manos.
- **Madrid running:** Esta APP, solo consta de carreras running y además pertenecientes a la Comunidad de Madrid. En ella se pueden buscar las carreras populares que se organizan en la Comunidad de Madrid solamente.



[Ilustración -2 APP Madrid Running](#)

Esta aplicación es atractiva visualmente, está organizada por meses, por lo que para buscar una competición en concreto hay que acceder a cada mes y buscar en el interior de la lista. No se puede ordenar ni buscar por km, ni nada semejante.

Por lo que a pesar de su funcionamiento muy correcto e intuitivo, creo que con nuestra APP podemos superarla con creces, ya que en nuestro diseño vamos a poder realizar una búsqueda más concreta y exhaustiva tanto de running como

de otro deporte y dentro y fuera de la Comunidad de Madrid. Además de mostrar en modo lista y modo mapa, añadiendo la funcionalidad de ordenar.

- **Todo carreras:** Esta APP es una estupenda agenda de carreras de todo tipo y cuenta con las de toda España.



Ilustración -3 APP Todo Carreras.

Puedes buscar carreras por tipo: carreras populares/ carreras de bomberos/ carreras de montaña/trails/raids/ carreras de orientación/carreras canicross/ carreras cross, además de también poder buscar carreras por provincia o por distancia (menos de 10km, 10km, media maratón, maratón, ultra maratón...)

Después de instalarla y hacer uso de esta APP he detectado que no se puede ordenar por fecha y que esta algo desactualizada. Hay bastantes carreras que no aparecen.

Por lo que gracias a esta APP se me ocurren bastantes modalidades de running que se podrían añadir a nuestra APP, ya que todas las mencionadas son de correr, por lo que no consta de bicicleta, duatlón y triatlón como la nuestra, aunque tengo que decir que es rápida e intuitiva.

- **Carreras por montaña:** esta es una aplicación para aficionados a las carreras por montaña.



Ilustración -4 APP CarrerasPorMontaña

Ofrece toda la información relevante sobre el mundo de las carreras por montaña: los datos detallados de cada carrera (fecha, lugar de celebración, distancia, desnivel, el recorrido en el mapa, modalidad, los lugares de interés

de la carrera); el calendario de las carreras en España y las más importantes del calendario mundial en todas sus modalidades; la ficha de los mejores corredores del circuito internacional con imágenes y videos de sus actuaciones; noticias de actualidad sobre el mundo de las carreras de montaña; permite localizar las carreras más cercanas a nuestra posición geográfica.

En cuanto a comentarios sobre la aplicación hay varios sobre error de conexión y de login, aunque realmente yo me registre a la primera, sin ningún tipo de problema.

Permite buscar por carreras por los alrededores y ver todas las carreras, aunque realmente no es muy intuitiva, es correcto que al buscar por ubicación el GPS no me llegaba a conectar, y además cuenta con un buscador por palabras, no sé si es muy útil para este tipo de competiciones.

De todas formas seguimos al igual que antes, esta aplicación solo está orientada al mundo de los corredores.

Por mucho que he investigado no he encontrado ninguna APP que incluya todas las características con las que contará la nuestra, que sea tan atractiva e intuitiva y que permita manejar los listados de competiciones en cuanto a ordenación y visualización en mapa. Por eso creemos que podemos llamar la atención y ser los primeros en destacar con esta APP para los running, duatletas, triatletas y bikers.

#### **d) Método de diario.**

**Este método será tenido en cuenta si el desarrollo permite llegar a este punto, en caso contrario supondría una segunda fase del proyecto.**

Los diarios pueden tomar diferentes formatos según las necesidades. De este modo, se pueden plantear a los participantes un diario abierto, en el que se expresen libremente con sus palabras, o uno cerrado, donde tendrán que contestar cuestionarios con varias opciones de respuesta.

Para su efectividad es necesario que los usuarios tengan claras las instrucciones sobre cómo realizar la tarea de registro. Qué acciones han documentado, cuál es la información que se desea sobre la acción (dónde, cuándo, por qué, con quién, durante cuánto tiempo, etc.), con qué frecuencia lo tienen que registrar, durante cuánto tiempo, etc.

## 2.2. Diseño Conceptual. Etapa de Diseño.

### 2.2.1. Escenarios de uso.

Antes de nada vamos a ver algunos conceptos importantes:

- **Los perfiles de usuarios** son agrupaciones de usuarios según sus características. Normalmente, son el resultado de un estudio cuantitativo que ha permitido definir los porcentajes de cada perfil y agruparlos por elementos comunes. Estas características pueden basarse en aspectos socio demográficos, actitudinales, de expectativas, etc.
- **Un personaje** es la descripción de un usuario arquetípico que puede servir como guía en el proceso de diseño.
- **Un escenario** es la descripción de un personaje en una situación de uso del sistema o del producto interactivo con unos objetivos concretos. Esta descripción incluye el contexto en el que tiene lugar la acción y la secuencia de acciones que se realizan. Las situaciones concretas que se generan a partir de estos elementos son de gran ayuda para explorar ideas y considerar aspectos del diseño.

#### 2.2.1.1. Los perfiles de Usuarios.

Realmente en esta APP vamos a disponer inicialmente de un usuario, que será el deportista, “**Usuario deportista**” este puede ser de mayor o menor edad, pero dispone de un móvil para acceder a la APP y al ser tan fácil, intuitiva, sencilla de manejar, no vamos a diferenciar como en otras aplicaciones entre usuarios apasionados por la tecnología y usuarios con desconocimiento de la tecnología.

Las tareas que podrán realizar estos usuarios serán las siguientes

- Tener acceso a todas las competiciones que deseen.
- Poder activar notificaciones sobre ciertos deportes y/o localidades.
- Consultar el mapa en modo ge localización para ver las carreras.

Si quiero mencionar que la idea es que en una segunda fase del proyecto se llegue a poder crear otro perfil de usuario, para los organizadores de eventos, para que ellos mismo puedan dar de alta las carreras “**Usuario Informador**”.

Las tareas que podrán realizar estos usuarios, una vez se desarrolle esta segunda fase, serán las siguientes:

- Dar de alta una nueva carrera.
- Modificar alguna carrera dada de alta por ellos mismos.

Como de momento solo vamos a trabajar con deportistas, vamos hacer una distinción de usuarios dentro de estos:

- **Deportistas no competitivos:** Son aquellos deportistas que entrenan para mantenerse en forma, pero que no asisten a competiciones, estos usuarios realmente no utilizaran la APP.
- **Deportistas por afición:** Son aquellos deportistas adictos al deporte y que de vez en cuando asisten a alguna competición.
- **Deportistas competitivos:** Son aquellos deportistas que tienen interés en asistir a competiciones habitualmente.
- **Deportistas muy competitivos:** Son aquellos deportistas que asisten a todas las competiciones posibles, y que intentan superarse en cada una de ellas.

### 2.2.1.2. Personajes o personas.

La técnica de los personajes se desarrolló originalmente como una técnica de ayuda al diseño, un personaje se construye con precisión y rigor a partir de la información cuantitativa y cualitativa de la investigación de usuarios: segmentación, perfil de usuarios, observación, encuestas, entrevistas y dinámicas de grupo. Por lo tanto, primero hace falta investigar a los usuarios, analizar los datos obtenidos y, finalmente, modelar los usuarios en personajes. Se considera, de esta manera, que un personaje es un modelo de usuario.

Los personajes se utilizan para que los diseñadores y los desarrolladores tengan en cuenta a los usuarios en las diferentes etapas del proceso de diseño y desarrollo y, por lo tanto, sirven de recordatorio para quién se diseña y ayudan a tener en cuenta a los usuarios en todo el proceso de diseño y desarrollo.

Analizando las encuestas realizadas, podemos destacar que disponemos en principio de los siguientes personajes:

- Running → Hemos visto que un porcentaje de los encuestados solo se dedica a correr.
- Bicicleta → Debido a que otro menor porcentaje de los encuestados solo se dedica a hacer carreras de bicicleta.
- Duetletas → Con este personaje vamos a cubrir a aquellos que corren, asisten a carreras de bicicletas y a duatlones.
- Triatleta → Y este último personaje consta de lo mencionado anteriormente, además de asistir a triatlones.

**Observaciones:** No necesitamos personaje alguno, para determinar únicamente a nadadores, ya que ninguno de los encuestados asiste a competiciones de nadar, solo nadan en los triatlones.

Con la información obtenida en la fase de análisis, se han podido identificar dos perfiles de usuarios, "Usuario Deportista" y "Usuario Informador" como ya mencionamos anteriormente, en la primera fase solo vamos a tratar el usuario deportista, en la segunda fase crearemos el usuario informador. Como recalco en el punto anterior, como solo vamos a tener deportistas, vamos hacer distinción de usuarios dentro de estos.

Para ayudar a construir los escenarios, crearemos tres personas ficticias que representarán los perfiles de usuarios explicados en el punto anterior.

**Carlos Gutiérrez (Deportista por afición).**

Es un corredor nato, entrena entre tres y cinco veces a la semana, pero solamente por mantenerse en forma y porque disfruta con ello, en alguna ocasión ha asistido a alguna carrera motivado por algún amigo.

Objetivos:

- Mantenerse en forma y disfrutar de correr.

Comportamientos:

- Entrenar semanalmente.

Necesidades

- Salir a correr y en alguna ocasión apuntarse a alguna competición.

[Tabla 1 - Primera Persona Ficticia\(Deportista por afición\).](#)

**Manuel García (Deportista competitivo).**

Es un duatleta desde hace tiempo, suele asistir a un duatlón como mínimo al mes.

Objetivos:

- Disfrutar con sus amigos en competiciones.

Comportamientos:

- Entrenar semanalmente running y bicicleta.

Necesidades

- Tener conocimiento de los posibles duatlones a realizar.

[Tabla 2 - Segunda Persona Ficticia \(Deportista competitivo\).](#)

**Juan Carlos Gómez (Deportista muy competitivo).**

Es un deportista desde hace años, suele asistir a carreras running, carreras de bicis, duatlones, triatlones,...

Objetivos:

- Asistir al máximo de competiciones al año.

Comportamientos:

- Entrenar semanalmente running, bicicleta y natación.

Necesidades

- Tener conocimiento de las posibles competiciones a realizar.

[Tabla 3 - Tercera Persona Ficticia \(Deportista muy competitivo\).](#)

### 2.2.1.3. Escenarios

**Carlos Gutiérrez** está entrenando con sus amigos runners, uno de ellos comenta que se ha enterado de que el próximo mes hay un maratón nuevo, que es el primero que se hace en esa localidad, que por qué no se apuntan, cuando terminan el entrenamiento, alguno de ellos buscan en internet y no encuentran mucha información.

Si alguno de ellos tuviera instalada “AppTrailSSport”, podría hacer una rápida búsqueda de competiciones y obtener información sobre la misma instantáneamente e incluso acceder a través de la APP a la web de la organización.

[Tabla 4 - Primer Escenario.](#)

**Manuel García** tiene que planificar sus próximas competiciones, porque tiene que planificar viajes de trabajo y las vacaciones. Pero no tiene claro las competiciones que están publicadas, está buscando en internet y muchas de ellas no tienen fecha aún o no tienen bien informada la web de inscripción.

Si tuviera instalada “AppTrailSSport”, podría hacer una rápida búsqueda de competiciones y planificar sus actividades, viajes y vacaciones.

[Tabla 5 - Segundo Escenario.](#)

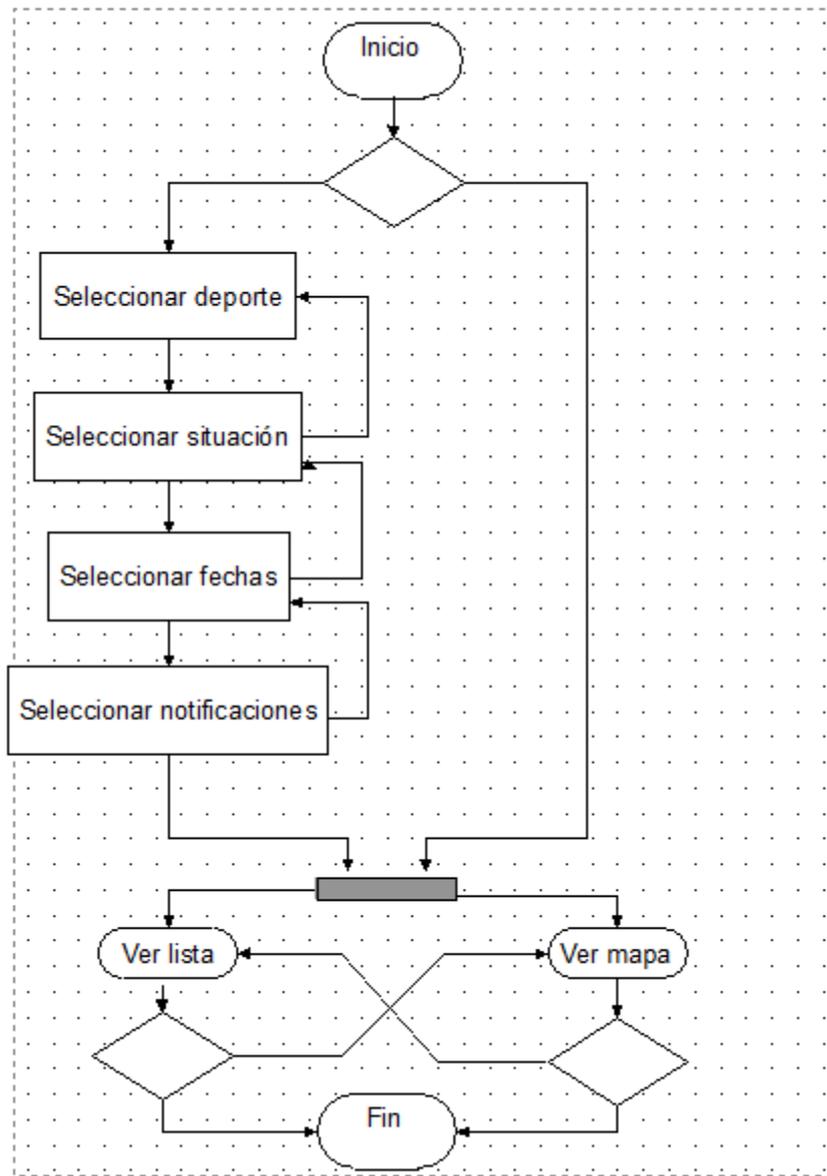
**Juan Carlos Gómez** está cansado de buscar carreras de running, bicicletas, duatlones y triatlones, para cuadrar fechas y entrenamiento, ya que muchas de ellas no están fijadas, pueden cambiar.

Si tuviera instalada “AppTrailSSport”, podría hacer una rápida búsqueda de competiciones y planificar sus actividades rápidamente, ya que podría verlas todas ordenadas por fechas.

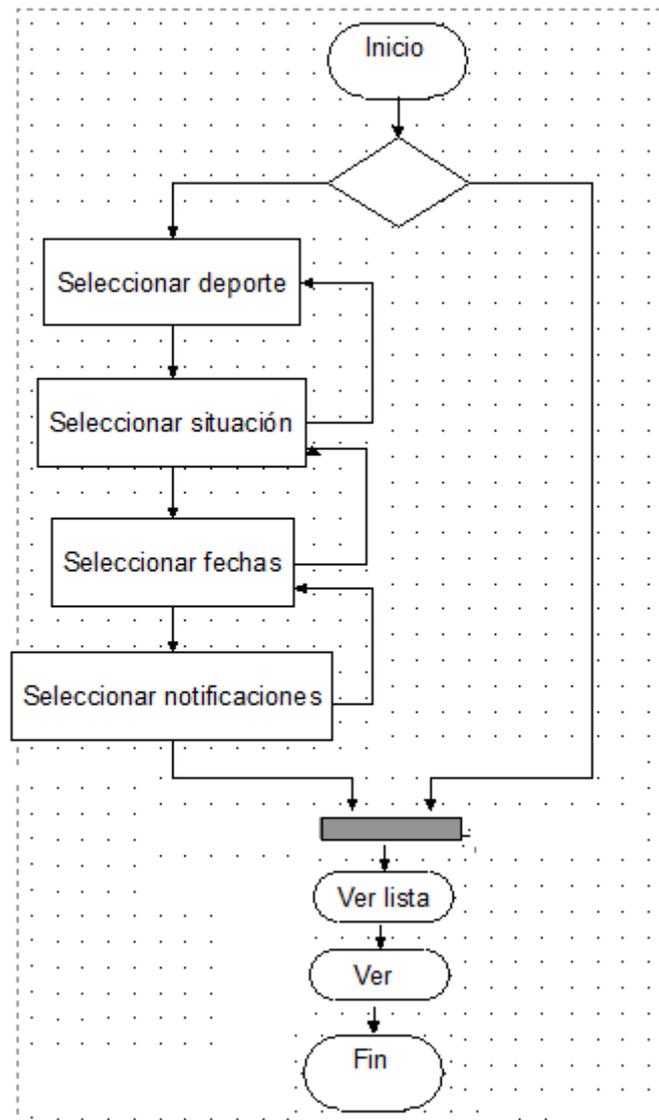
[Tabla 6 - Tercer Escenario.](#)

**2.2.2. Flujos de interacción.**

Para realizar el diagrama he utilizado para el diseño “EDGE Diagrammer Diagnostic”.



Este es el diagrama inicial, pero a continuación os mostraré las modificaciones que ha sufrido el diagrama UML, se ha eliminado de este, la parte del mapa, ya que no se ha podido abordar dicha implementación. En el caso de la pantalla de notificaciones, la hemos dejado porque aunque no estén disponibles los check, la pantalla está disponible en la APP y pasamos por ella hasta llegar al listado.



La idea principal siempre ha sido la misma, una aplicación sencilla de manejar e intuitiva, por lo que la pantalla de inicio dispone de tres botones, “Buscar Competiciones” (configurar búsqueda), y si ya se ha realizado la búsqueda, estará habilitado el botón de “Ver Lista Competiciones”, vuelvo a comentar que el botón “Ver Mapa Competiciones” de mapa no estará disponible porque no se ha realizado esa mejora.

El primer botón “Buscar Competiciones”, nos permitirá acceder a las pantallas de configuración de búsqueda de competiciones, comenzando por la pantalla de seleccionar deporte, seleccionar provincia, seleccionar fechas y activar notificaciones.

Una vez llegados a este punto ya se podrá consultar el listado a través del botón “Ver Lista Competiciones”. Cuando accedamos de nuevo a la aplicación, si se quiere consultar la lista de competiciones para la búsqueda ya configurada, solo tendremos que acceder a “Ver Lista Competiciones”, ya que la configuración queda almacenada.

Creo que el funcionamiento y el desplazamiento a través de las pantallas, es muy simple, no puede llevar a equívocos y el usuario no perderá prácticamente tiempo en entender su funcionamiento, es uno de los objetivos claves en este tipo de aplicaciones.

### **2.3. Diseño de prototipos. Etapa de transición entre la de diseño y la de evaluación.**

En este punto vamos a presentar los diseños de pantallas con las que contará nuestra aplicación, un primer punto con un diseño a mano alzada y en otro segundo punto un diseño más exhaustivo y concreto.

#### **2.3.1. Sketches.**

Con la obtención de los resultados de los análisis realizados en los puntos anteriores, vamos a presentar un diseño inicial a mano alzada.

#### **2.3.2. Prototipado.**

Para diseñar los prototipos he utilizado la una aplicación online, <https://www.fluidui.com/> y a pesar de que la versión de pruebas tiene límite de pantallas y he tenido que borrar algunas para crear las últimas, es bastante fácil de manejar y tiene muchísimas opciones y características.

Tengo que añadir que estas pantallas son el prototipo inicial, las definitiva que aparecen en la APP, las encontraremos en el punto 3.2.

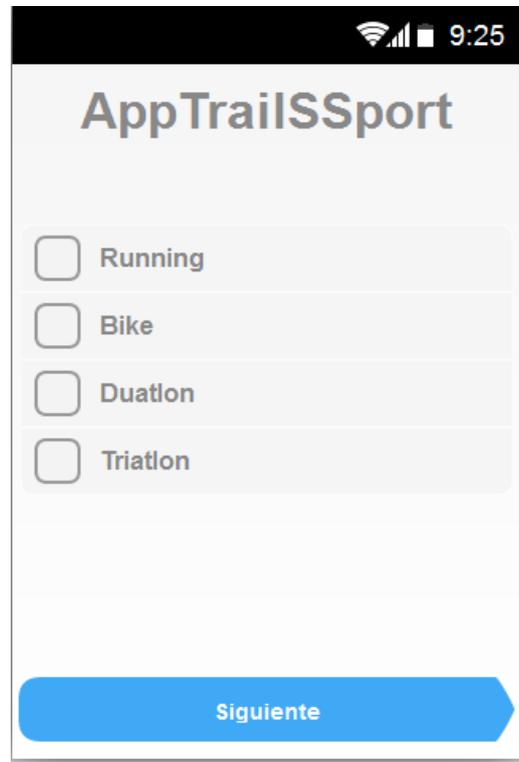
La pantalla que se abre al acceder por primera vez a la APP es la siguiente:



[Pantalla 5 - Pantalla inicial.](#)

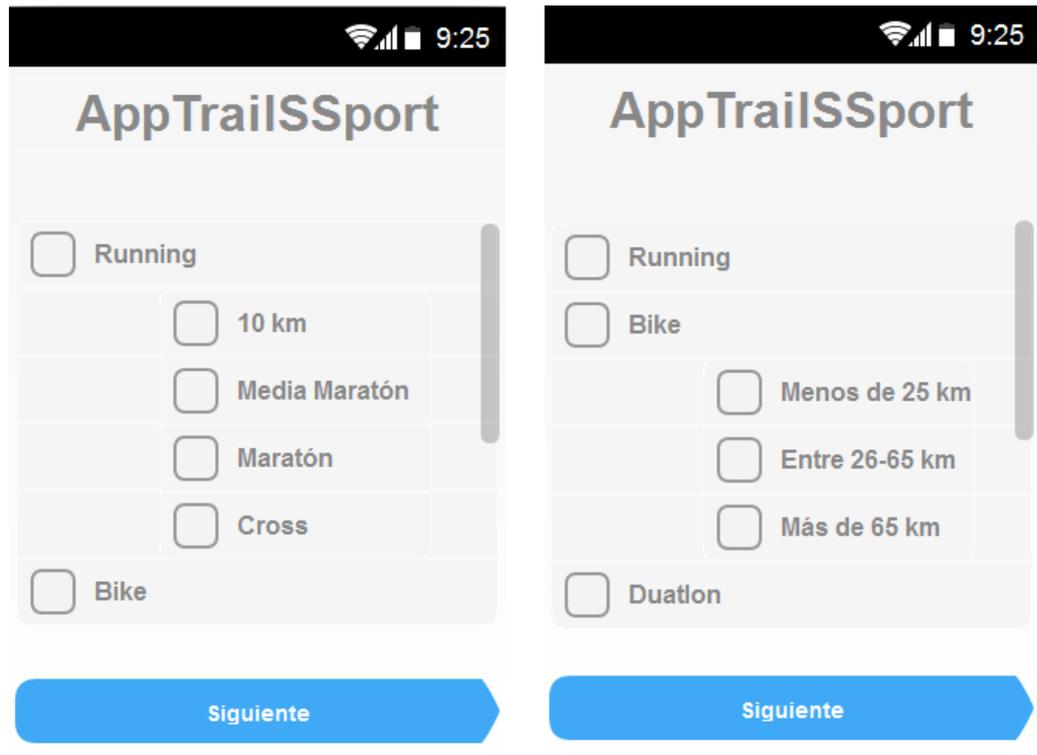
Desde esta pantalla al pulsar sobre el botón “Buscar Competiciones” se dará paso a la configuración de los filtros para realizar la búsqueda de las competiciones.

La primera pantalla de configuración es la de selección de deportes.



Pantalla 6 - Selección de deportes.

Esta pantalla si seleccionas “Running” o “Bike” se darán opciones de subcategorías, como se pueden ver en las siguientes imágenes:



[Pantalla 7 - Subcategorías de running y bike.](#)

La configuración continúa con la selección de ubicación de las competiciones.



[Pantalla -8 - Ubicación y provincias.](#)

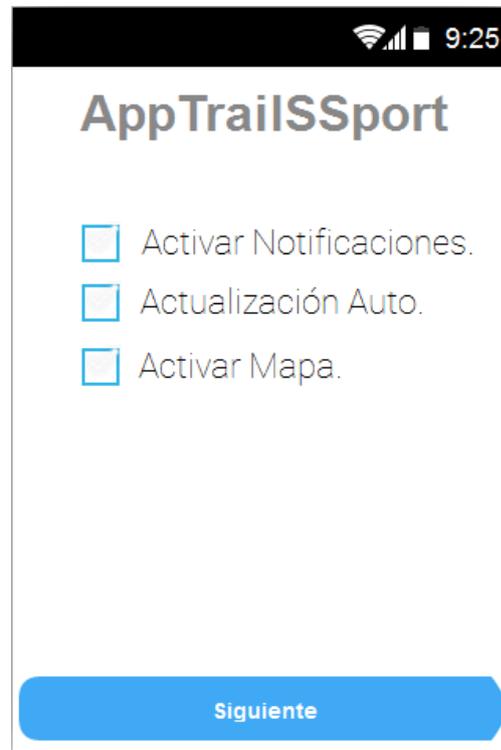
Si continuamos con la configuración de las competiciones que nos interesé, pasaremos a la selección de fechas, y días.



The screenshot shows the AppTrailSSport app interface. At the top, there is a black status bar with a Wi-Fi icon, a battery icon, and the time 9:25. Below the status bar, the app title "AppTrailSSport" is displayed in a large, bold, grey font. Underneath the title, there are two date selection fields: "Desde:" followed by a date input field containing "12 - 5 - 2014" and a calendar icon, and "Hasta:" followed by another date input field containing "12 - 5 - 2014" and a calendar icon. Below these fields is a list of four options, each with a checked checkbox: "Sábado.", "Domingo.", "Festivos.", and "Todos.". At the bottom of the screen, there is a blue button with a white arrow pointing right, labeled "Siguiente".

[Pantalla 9 - Selección de fechas y tipo de días.](#)

Y para finalizar configuraremos las notificaciones y actualizaciones de la mismas.



The screenshot shows the AppTrailSSport app interface. At the top, there is a black status bar with a Wi-Fi icon, a battery icon, and the time 9:25. Below the status bar, the app title "AppTrailSSport" is displayed in a large, bold, grey font. Underneath the title, there are three notification configuration options, each with an unchecked checkbox: "Activar Notificaciones.", "Actualización Auto.", and "Activar Mapa.". At the bottom of the screen, there is a blue button with a white arrow pointing right, labeled "Siguiente".

[Pantalla 10 - Configurar Notificaciones.](#)

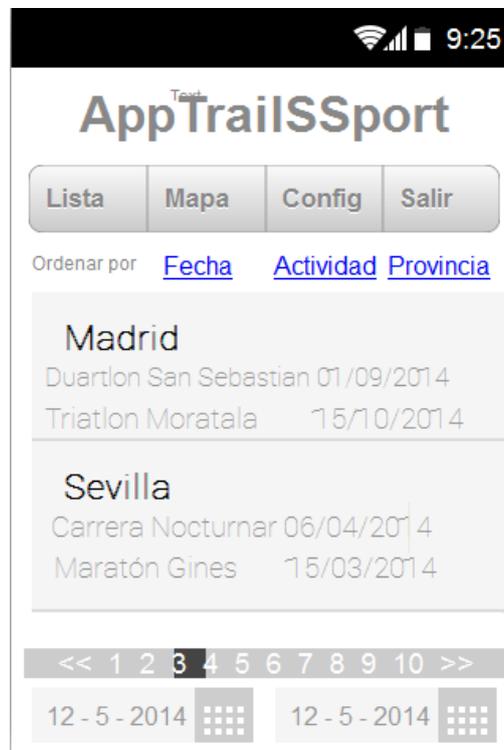
De esta pantalla tenemos que destacar los tres check presentados:

- Activar Notificaciones: con este check el usuario activará que quiere ser informado de las carreras que se den de alta en la APP correspondientes a su configuración de búsqueda.
- Actualización Auto.: Nos referimos a que si hay nuevas versiones de la aplicación, esta se actualice sin avisar al usuario de ello.
- Activar mapa: Necesario para poder ver la lista de competiciones en modo mapa.

Otro punto importante es que llegados a este momento, al pulsar el botón de siguiente se presentará el listado de competiciones con la casuística configurada en las pantallas anteriores.

Esta configuración puede ser modificada por el usuario en cualquier momento. Pero la configuración que queda en la aplicación, se almacena en memoria para poder consultar competiciones en todo momento sin tener que informar de ello.

Llegados a este punto si pulsamos "Siguiente" nos aparecerá la pantalla con las competiciones que concuerden con nuestra configuración de búsqueda.

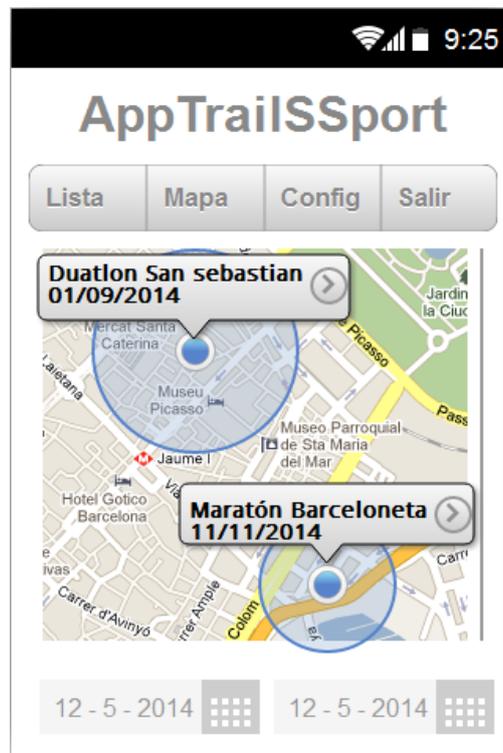


[Pantalla 11 – Ver listado de competiciones.](#)

Desde esta pantalla se pueden realizar varias acciones:

- Botón “Lista”: No estará disponible porque nos encontramos en este modo.
- Botón “Mapa”: cambiar de modo lista a modo mapa.
- Botón “Config”: Volveremos a la primera pantalla de configuración, la de seleccionar deportes, para cambiar la configuración.
- Botón “Salir”: Salir de la aplicación.
- Botones de Ordenar por “Fecha”, “Actividad” o “Provincia”.
- Pulsar sobre una competición: Aparecerá la pantalla para ver detalle de la competición.
- Botones de fecha: Para modificar el rango de fechas de las competiciones que estamos visualizando.

Vamos a comenzar con el botón “Mapa”, a través de este botón accedemos a la siguiente pantalla donde se muestran las competiciones en mapa.



[Pantalla 12 – Ver competiciones en mapa.](#)

Al igual que en la pantalla anterior, desde esta pantalla se pueden realizar varias acciones:

- Botón “Lista”: cambiar de modo mapa a modo lista.
- Botón “Mapa”: No estará disponible porque nos encontramos en este modo.
- Botón “Config”: Volveremos a la primera pantalla de configuración, la de seleccionar deportes, para cambiar la configuración.
- Botón “Salir”: Salir de la aplicación.
- Botones de Ordenar por “Fecha”, “Actividad” o “Provincia”.
- Pulsar sobre una competición: Aparecerá la pantalla para ver detalle de la competición.
- Botones de fecha: Para modificar el rango de fechas de las competiciones que estamos visualizando.

Una vez que estamos situados en la pantalla ver competiciones en lista o en mapa, si pulsamos sobre una de las competiciones accederemos a su información, como podemos ver en la siguiente pantalla:



[Pantalla 13 –Ver detalle de una competición.](#)

En principio la información que se mostrará de cada competición, será:

- Título: Compuesto de nombre y fecha
- El mapa: Por si se quiere acceder a él para calcular la distancia.
- Localidad y provincia.
- Modalidad y distancias.
- Breve descripción sobre el evento.
- Un link con el acceso directo a su página web.

De momento esta es la información con la que constará cada competición, a no ser que creamos necesario añadir alguna otra.

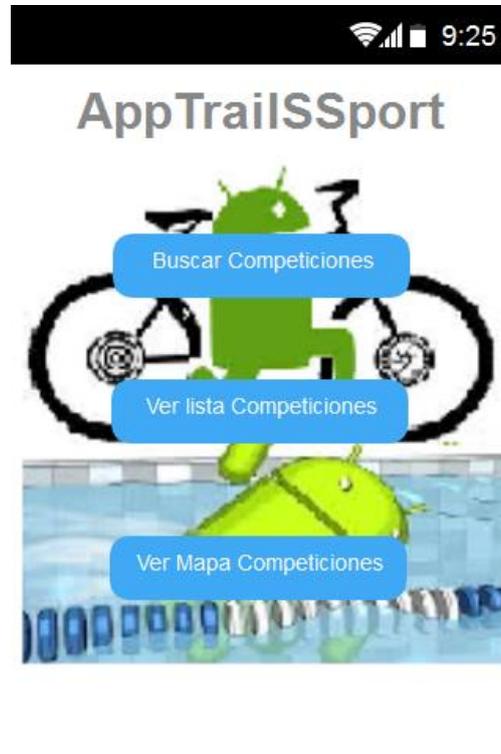
A partir de ahora si volvemos a acceder a la aplicación, como ya están configurados los filtros de competiciones, la pantalla inicial habrá cambiado y nos permitirá acceder directamente a la lista o mapa de competiciones.



Pantalla 14 – Acceso normal.

Estoy estudiando si queda más atractivo añadir en cada pantalla imágenes como fondo de pantalla, tengo pensado hacer algunas pruebas con imágenes de carreras, duatlones, triatlones u otras competiciones.

De momento hemos hecho alguna prueba con la pantalla inicial añadiéndole el logo de la aplicación en el fondo de la pantalla, adjuntamos a continuación una prueba, aunque vamos a descartar el tema de poner fotografías o dibujos como fondo de pantalla porque al sobreponer textos encima de imágenes, la mayoría de las veces se dificulta la lectura de las pantallas.



[Pantalla 15 - Prueba de imagen como fondo.](#)

#### 2.4. Evaluación. Etapa de evaluación.

Los métodos de evaluación permiten obtener información sobre los diseños y de este modo mejorarlos y/o validarlos. La iteración en los procesos de diseño es un aspecto clave en el DCU y es en esta combinación diseño-evaluación donde es más importante. Así, desde los primeros prototipos, deben incluirse puntos de evaluación para trabajar siempre teniendo en cuenta las necesidades, los deseos y las limitaciones de los usuarios.

En este apartado se describen los procesos que se llevarán a cabo para realizar la etapa de evaluación.

Esto servirá para situar al usuario en un entorno más cercano, y parecido al que encontraría cuando utilizará la aplicación.

Esto nos dará información de mucho valor para nuestro estudio, saber qué puntos se pueden mejorar o cambiar.

### 2.4.1. Opiniones/Test de Usuarios en cuanto al diseño.

En general, el objetivo de un test con usuarios es proporcionar información de carácter cualitativo, que no cuantitativo o estadístico. Así pues, normalmente hay que dar más importancia a la interpretación de los resultados que a la elaboración de métricas o estadísticas. Mediante el test con usuarios se comprueba si las interfaces y los procesos están bien diseñados y si los usuarios encuentran obstáculos y dificultades para lograr sus objetivos.

Los test de usuarios se han realizado con un grupo de personas de variada edad y que realizan distintas actividades deportivas, sus sugerencias han sido las siguientes:

- Juan de Dios López Baena ha sugerido que en la parte de running debería haber una opción de Trail, lo estudiaremos antes de implementarlo.
- Antonio Jesús Castaño Camino aunque no suele asistir a competiciones, practica ciclismo, trabaja como programador Android, y como tal sugiere que debería tener más diseño la interfaz.
- Jordi Almirall López en una de las consultas realizadas por mail, me comento que era mejor en cuanto a manejabilidad que las pantallas tuvieran el botón de siguiente en lugar de pasar de unas a otras deslizando el dedo simplemente, ya que era menos intuitivo. Esto ya se ha tenido en cuenta en el diseño.
- Quiero resaltar los comentarios de Rafael Baena Cobol, que comentaba que la aplicación era muy simple y fácil de manejar, muy intuitiva y que no puede llevar a errores ni equívocos a la mayoría de usuarios.
- También destacar a Jairo García que me comentaba que entrena con un grupo de vecinos y que suelen asistir a medias maratones y maratones, nos da la opinión de él y de sus compañeros de entrenamiento como que en principio ven la aplicación dentro de la normalidad, me han insistido en que en cuanto esté funcionando que están muy interesados en utilizarla, que les parece muy útil.
- Fran Baena Castaño también comenta que el diseño es claro e intuitivo.
- Por parte de unos amigos que realizan escalada he recibido la sugerencia de añadir competiciones de escalada, por lo que lo estudiare para añadirlo más adelante.

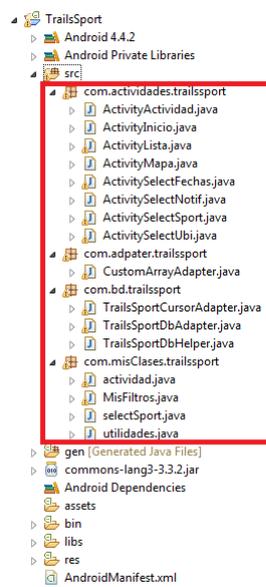
### 3. Desarrollo del Proyecto.

#### 3.1. Implementación.

Para organizar la implementación he estructurado las clases en los siguientes paquetes, dentro del directorio src:

- com.actividades.trailssport:
  - o **class** ActivityActividad
  - o ActivityInicio
  - o ActivityLista
  - o ActivityMapa
  - o ActivitySelectFechas
  - o ActivitySelectNotif
  - o ActivitySelectSport
  - o ActivitySelectUbi
- com.adapter.trailssport
  - o CustomArrayAdapter
- com.bd.trailssport
  - o TrailsSportCursorAdapter
  - o TrailsSportDbAdapter
  - o TrailsSportDbHelper
- com.misClases.trailssport
  - o actividad
  - o MisFiltros
  - o selectSport
  - o utilidades

Adjunto una captura con la estructura aquí explicada:



### 3.1.1. Activity y/o clases.

#### 3.1.1.1. Clase ActivityInicio.

Esta es la actividad principal de nuestra aplicación, muestra los tres botones que permiten realizar las funciones principales de la aplicación. Esto lo hace invocando al layout activity\_inicio.



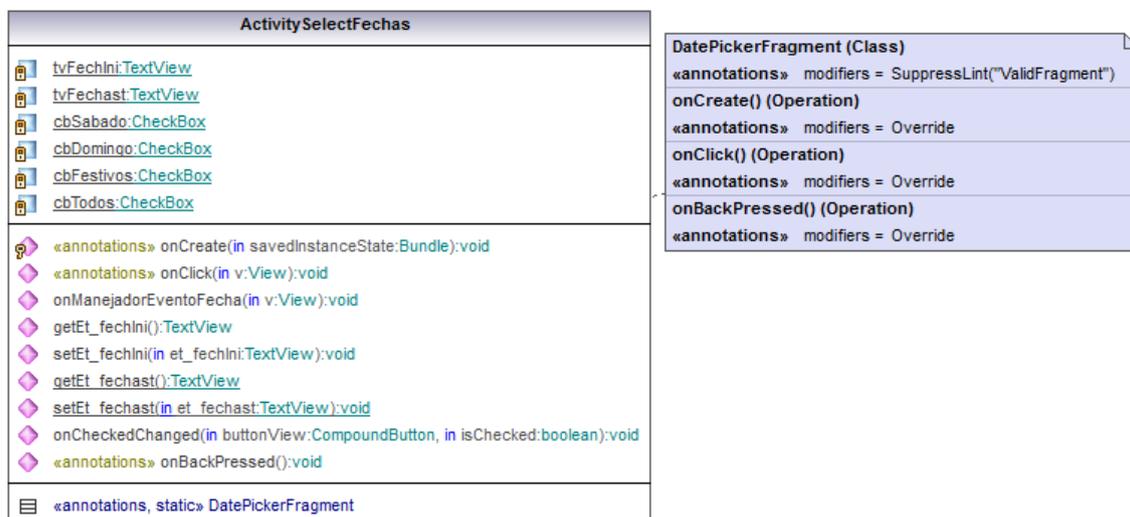
Comentar de esta clase y que sirva de precedente, que extiende de activity porque se trata de una actividad e implementa a OnClickListener para poder controlar cuando se pulsan los botones.

```
public class ActivityInicio extends Activity implements OnClickListener {
```

En las siguientes clases que extiendan de Activity o implementen a OnClickListener, no se volverán a mencionar para no duplicar información de explicación de las clases.

#### 3.1.1.2. Clase ActivitySelectFechas.

Esta es la actividad a través de la cual informar el rango de fechas, para el que se quieren buscar las competiciones. Esto lo hace invocando al layout actividad\_fechas.



Destacar de esta clase que implementa CompoundButton.OnCheckedChangeListener, debido a que en esta clase hemos creado un popup para la selección de fecha desde y fecha hasta. Para ello hemos utilizado la siguiente clase, se podría usar desde fuera del activity, pero en nuestro caso la hemos añadido dentro del activity porque no se usará en más sitios.

```
public static class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {
    private View view;
    public DatePickerFragment(View view) {
        this.view = view;
    }
}
```

```

    }

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        final Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);
        // Create a new instance of DatePickerDialog and return it
        return new DatePickerDialog(getActivity(), this, year, month, day);
    }
    public void onDateSet(DatePicker view, int year, int month, int day) {
        if (this.view.getId() == R.id.bt_fehcini){
            tvFechlni.setText(day+"/"+(month+1) + "/" + year);
        }else if (this.view.getId() == R.id.bt_fechhas){
            tvFechast.setText(day+"/"+(month+1) + "/" + year);
        }
    }
}
}

```

El método `onDateSet` es el que recuperará la fecha seleccionada por el usuario, teniendo en cuenta que enero es el mes cero, tendremos que sumar uno para que la fecha seleccionada por el popap sea la misma que recuperamos. Con el método `onManejadorEventoFecha` capturo el evento del botón que se pulsa y llamamos a la clase que hemos creado.

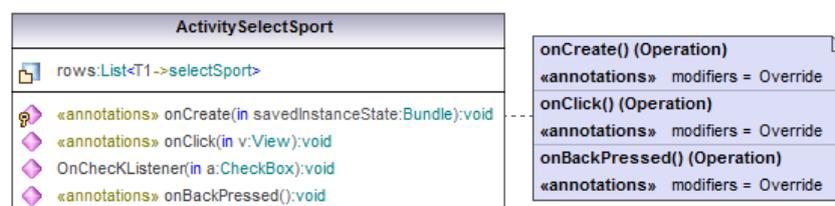
```

public void onManejadorEventoFecha(View v) {
    DialogFragment newFragment = new DatePickerFragment(v);
    Bundle args = new Bundle();
    int vista = v.getBottom();
    args.putLong("vista",vista);
    newFragment.setArguments(args);
    newFragment.show(getFragmentManager(), "datePicker");
}

```

### 3.1.1.3. Clase ActivitySelectSport.

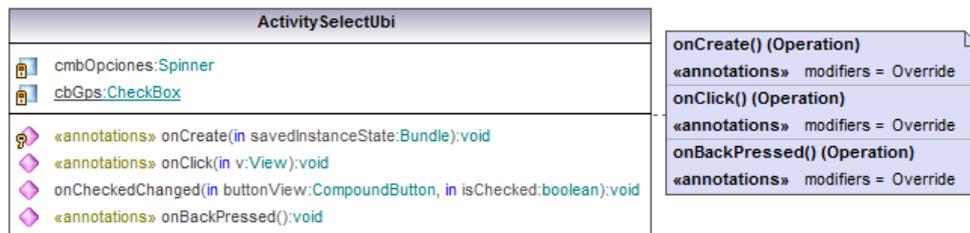
Esta es la actividad a través de la cual informar de los tipos de deportes o competiciones que interesan para realizar la búsqueda de competiciones. Esto lo hace invocando al layout `actividad_sport`.



En esta clase, destacar que para crear los check, hemos creado un array con los check y su información. Aunque realmente usamos una clase adaptadora, "CustomArrayAdapter", un layout con un listview "listview.xml", y para devolver las filas del array usamos un Relative Layout que contiene un Checkbox, una imagen y dos TextView, es el layout `actividad_sport` (`actividad_sport`).

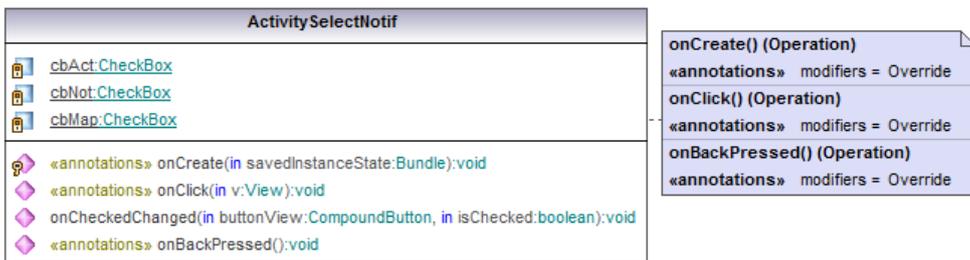
### 3.1.1.4. Clase ActivitySelectUbi.

Esta es la actividad a través de la cual informar de la provincia que interesa para realizar la búsqueda de competiciones. El check GPS no está implementado, pero lo hemos dejado en el diseño porque es una de las mejores futuras. Este activity invoca al layout `actividad_ubicacion`.



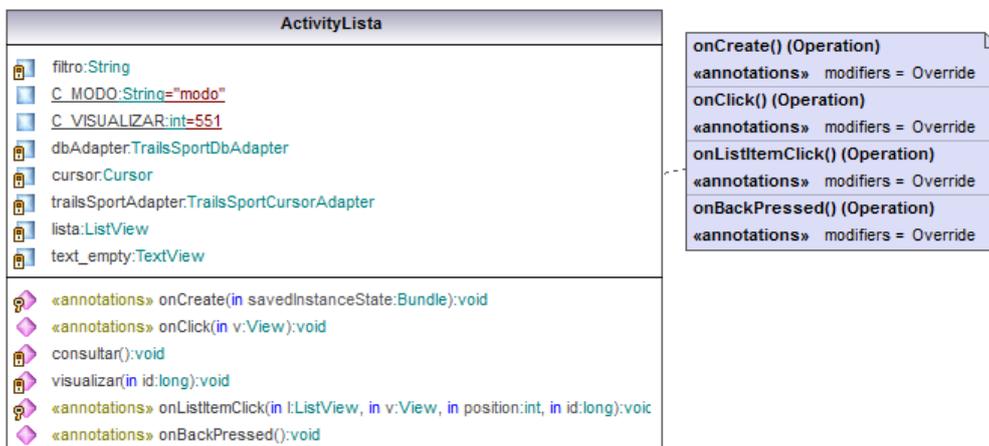
### 3.1.1.5. Clase ActivitySelectNotif.

Esta es la actividad a través de la cual informar de las notificaciones a activar, no vamos a comentar nada más de esta activity, ya que no se ha incluido la funcionalidad de los check, solo el diseño. Es una de las mejoras futuras. Esta activity invoca al layout actividad\_notif.



### 3.1.1.6. Clase ActivityLista.

Esta es la actividad encargada de mostrar la lista de competiciones, después de haber pasado por las pantallas de configuración de búsqueda. Esto lo hace invocando al actividad\_lista.



De esta activity, comentar varias cosas; con el método onCreate obtenemos la referencia al ListView incluida en el layout, creamos y preparamos el adaptador dbAdapter para gestionar las operaciones sobre la entidad, en la base de datos SQLite.

Por otro lado, el método **consultar()** se encargará de vincular el CursorAdapter TrailssportAdapter a la ListView lista.

Además utilizaremos el evento **onListItemClick** de la **ListActivity** ActivityLista, que llama al método **visualizar**. Este método declara el **Intent** necesario para iniciar la nueva actividad. En este **Intent** incluimos los parámetros “modo de comportamiento” e “identificador del registro seleccionado”. Esto lo realizamos de este modo por si implementábamos el modo inserción.

Esta clase es bastante importante, ya que es la encargada de crear la base de datos, realizar la consulta y mostrar las competiciones devueltas por la consulta en un listview creado en otro layout.

Para realizar estas funcionalidades hemos implementado tres clases adaptadoras, una de ellas, TrailsSportDbHelper, para crear la base de datos, otra clase TrailsSportDbAdapter, para realizar las consultas a la base de datos y otra clase TrailsSportCursorAdapter, para añadir los datos recuperados al listview. Estas tres clases las veremos más adelante.

### 3.1.1.7. Clase ActivityActividad.

Esta es la actividad encargada de mostrar todos los datos de la actividad seleccionada en la pantalla (o activity) anterior. Esto lo hace invocando al layout actividad\_actividad.

Una vez tenemos preparado el adaptador (clase TrailsSportDbAdapter explicada más adelante) para obtener el registro de la base de datos SQLite, realizamos las siguientes operaciones:

- El método onCreate obtiene los datos que hemos pasado desde la Actividad anterior, ActiviyLista.
- Con el método consultar prepara el adaptador de base de datos que utilizamos para consultar el registro, por su id.
- Obtenemos el registro de la base de datos y rellenamos los campos que vamos a mostrar.



Destacar que el fichero actividad\_actividad.xml es donde se visualizará toda la información de la competición, lo podemos ver a continuación:

```

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".HipotecaFormulario" >

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="10dp" >

        <TextView
            android:id="@+id/label_nombre"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/label_nombre_comp"
    
```

```

        android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView
    android:id="@+id/nombre"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/label_nombre" />

<TextView
    android:id="@+id/label_provincias"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/label_provincias"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:layout_below="@+id/nombre"
    android:layout_marginTop="20dp" />

<TextView
    android:id="@+id/provincia"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/label_provincias"
    android:ems="10"
    android:inputType="textMultiLine">
</TextView>

<TextView
    android:id="@+id/label_fecha"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/label_fecha"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:layout_below="@+id/provincia"
    android:ems="10"
    android:layout_marginTop="20dp" />

<TextView
    android:id="@+id/fecha"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/label_fecha"
    android:ems="10">
</TextView>

<TextView
    android:id="@+id/label_descripcion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/label_descripcion"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:layout_below="@+id/fecha"
    android:ems="10"
    android:layout_marginTop="20dp" />

<TextView
    android:id="@+id/descripcion"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/label_descripcion"
    android:ems="10"
    android:inputType="textMultiLine">
</TextView>

<TextView
    android:id="@+id/label_web"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/descripcion"
        android:layout_marginTop="20dp"
        android:text="@string/label_web"
        android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView
    android:id="@+id/web"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:clickable="true"
    android:layout_alignRight="@+id/descripcion"
    android:layout_below="@+id/label_web"
    android:textColor="#0099CC"

/>

<TextView
    android:id="@+id/label_tipo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/label_tipo"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:layout_below="@+id/web"
    android:layout_marginTop="20dp" />

<TextView
    android:id="@+id/tipo"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/label_tipo"
    android:layout_toLeftOf="@+id/imageView1"
    android:ems="10"
    android:inputType="textMultiLine" >

</TextView>

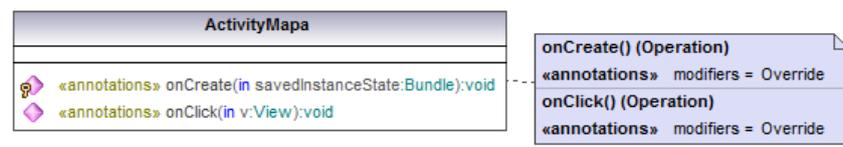
<ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/web"
    android:layout_alignTop="@+id/label_tipo"
    android:layout_marginRight="14dp"
    android:src="@drawable/bike" />

</RelativeLayout>
</ScrollView>

```

### 3.1.1.8. Clase ActivityMapa.

Esta es la actividad que en caso de estar implementada mostraría la lista de competiciones, pero posicionando todas estas competiciones en un mapa de Google Maps. Es una mejora futura. Esto lo haría invocando al layout activity\_inicio.



### 3.1.1.9. Clase CustomArrayAdapter y Clase ListaCompe.

La clase CustomArrayAdapter es la encargada de mostrar los datos de un array, del tipo de la clase row ListaCompe en el listview. Donde para cada registro mostrado en el listview, se mostrará una fila del array a través del Relative Layout que contiene el Checkbox, la imagen y los TextView con el titulo y el subtítulo (actividad\_sport).



Dentro de este .java, hemos creado también la clase ListaCompe con los atributos que vamos a mostrar en la lista, título, subtítulo y el checkbox.

EL listview, listview.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<ListView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/list"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:cacheColorHint="@android:color/transparent"
    android:divider="@android:color/transparent"
    android:dividerHeight="8dp"
    android:scrollbarStyle="outsideOverlay"
    android:paddingBottom="8dp"
    android:paddingLeft="5dp"
    android:paddingRight="5dp"
    android:paddingTop="15dp" >

</ListView>
    
```

EL RelativeLayout, listview\_sport.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >

    <!-- android:background="@drawable/listview_selector" -->

    <CheckBox
        android:id="@+id/checkBox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_centerVertical="true"
        android:paddingBottom="10dp"
    
```

```

        android:paddingTop="10dp" />

<ImageView
    android:id="@+id/icono"
    android:layout_width="55dp"
    android:layout_height="70dp"
    android:layout_toRightOf="@id/checkbox"
    android:adjustViewBounds="true"
    android:scaleType="centerInside"
    android:src="@drawable/swim" />

<RelativeLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_toRightOf="@+id/icono"
    android:paddingLeft="10dp" >

<TextView
    android:id="@+id/textViewTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="rgerhg"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView
    android:id="@+id/textViewSubtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textViewTitle"
    android:text="ewgweg"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textSize="10sp" />
</RelativeLayout>
</RelativeLayout>
    
```

### 3.1.1.10. Clase TrailsSportCursorAdapter.

Hemos definido esta clase para utilizarla desde la Activity ActivityLista, alimentado con ella el ListView con los datos obtenidos a través de la clase TrailsSportDbAdapter.

The screenshot shows the class `TrailsSportCursorAdapter` with the following methods and annotations:

- `dbAdapter.TrailsSportDbAdapter=null`
- `«constructor» TrailsSportCursorAdapter(in context:Context, in c:Cursor)`
- `«annotations» bindView(in view:View, in context:Context, in cursor:Cursor):void`
- `«annotations» newView(in context:Context, in cursor:Cursor, in parent:ViewGroup):View`

Annotations for `bindView()` and `newView()` are shown as:

- `«annotations» modifiers = Override`

El método **bindView**, carga la información de cada competición en la lista.

```

public void bindView(View view, Context context, Cursor cursor)
{
    TextView tv = (TextView) view ;
    setText(utilidades.formatDate(
        cursor.getString(cursor.getColumnIndex(TrailsSportDbAdapter.C_COLUMNA_FECHA)))
    +" (" +cursor.getString(cursor.getColumnIndex(TrailsSportDbAdapter.C_COLUMNA_TIPO))
    +" ) - "+
        cursor.getString(cursor.getColumnIndex(TrailsSportDbAdapter.C_COLUMNA_NOMBRE)));
}
    
```

### 3.1.1.11. Clase TrailsSportDbAdapter.

Hemos definido la clase TrailsSportDbAdapter para gestionar las operaciones en la base de datos SQLite. Esta es la clase que utilizamos para realizar la consulta de un registro en base de datos, necesaria para devolver la información de una actividad en concreto desde el activity ActivityActividad.



Desde la actividad ListaActivity recibimos el identificador del registro que debe mostrarse, para obtener el registro de la base de datos SQLite, hemos incluido el método **getRegistro** en la clase **TrailsSportCursorAdapter.java**.

### 3.1.1.12. Clase TrailsSportDbHelper.

La creación y conexión de la base de datos, esto es lo que se realiza a través de esta clase que hereda de SQLiteOpenHelper. Esta clase gestiona la creación de la base de datos la primera vez que se accede a la misma.



En esta clase hemos implementado el método onCreate con la creación de la tabla COMPETICIONES que consta de los siguientes campos:

- \_id: Identificador del registro.
- comp\_nom: Nombre de la competición.
- comp\_fec: Fecha de la competición.
- comp\_tipo: Tipo de competición.
- comp\_desc: Descripción de la competición.
- comp\_url: Página web de la organización de la competición.
- comp\_prov: Provincia donde se realiza la competición.
- comp\_x: Este campo era para guardar las coordenadas, pero como no se ha implementado la parte de mapas, no lo utilizamos, inserto un 1 para evitar errores.
- comp\_pobl: Población donde se realiza la competición.

### 3.1.1.13. Clase actividad.

Esta es la clase creada para poder crear un objeto actividad con sus atributos nombre, fecha y seleccionado. De esta forma la lista de actividades se puede implementar con dicho objeto. Aunque realmente no la estamos usando en estos momentos, pero no ha sido eliminada porque forma parte de las mejoras futuras.

actividad	
<pre> nombre:String fecha:String descrip:String seleccionado:boolean checkListener:OnClickListener=new OnClickListener() { @Override public void onClick(View v) { actividad a = (actividad) v.getTag(); a.setChecked(!a.isChecked()); } } </pre>	
<pre> «constructor» actividad(in nombre:String, in fecha:String, in seleccionado:boolean) getNombre():String setNombre(in nombre:String):void getFecha():String setFecha(in fecha:String):void isChecked():boolean setChecked(in checked:boolean):void getDescrip():String setDescrip(in descrip:String):void getCheckListener():OnClickListener setCheckListener(in checkListener:OnClickListener):void </pre>	

### 3.1.1.14. Clase MisFiltros.

Esta clase es utilizada para llamar al adaptador que realiza la consulta en base de datos, pasándole un objeto con los atributos de dicha clase.

De esta forma una vez que recuperamos los datos de las SharedPreferences, que son los datos que se configuran a través de las pantallas para realizar la consulta de competiciones, se crea un objeto de tipo Misfiltros con sus atributos correspondientes y se llama al método adaptador, que realiza la consulta a base de datos, pasándole este objeto con los filtros.

MisFiltros	
<pre> deportes:String provincia:String fechaDesde:long fechaHasta:long seleccionado:boolean total_sport:int=0 </pre>	
<pre> «constructor» MisFiltros(in deportes:String, in provincia:String, in fechaDesde:long, in fechaHasta:long) getDeportes():String setDeportes(in deportes:String):void getProvincia():String setProvincia(in provincia:String):void getFechaDesde():long setFechaDesde(in fechaDesde:long):void getFechaHasta():long setFechaHasta(in fechaHasta:long):void filtrosQuery():String[] getTotal_sport():int setTotal_sport(in total:int):void </pre>	

Esto podemos verlo en el método Consultar, de la clase ActivityList:

```

private void consultar() {

    SharedPreferences settings = getSharedPreferences("PREFERENCES", Context.MODE_PRIVATE);
    String conf_ini = settings.getString("CONFI", "NO");
    String filtro_sport = settings.getString("sport", "");
    String filtro_prov = settings.getString("provincia", "NO");
    long filtro_fecDes = settings.getLong("fechaDesde", 00000000);
    long filtro_fecHas = settings.getLong("fechaHasta", 00000000);

    Editor editor = settings.edit();
    editor.putString("CONFI", "OK");
    editor.commit();

    MisFiltros miFiltro = new MisFiltros(filtro_sport, filtro_prov, filtro_fecDes, filtro_fecHas);

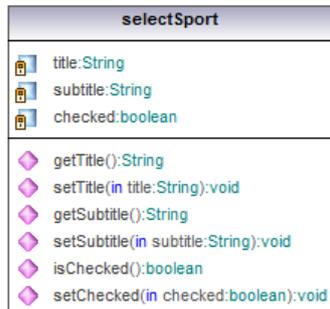
```

```

cursor = dbAdapter.getCursor(miFiltro);
startManagingCursor(cursor);
trailsSportAdapter = new TrailsSportCursorAdapter(this, cursor);
lista.setAdapter(trailsSportAdapter);
lista.setEmptyView(text_empty);
}
    
```

**3.1.1.15. Clase selectSport.**

Esta clase se ha creado con los atributos título, subtítulo y check, utilizando un objeto de este tipo para crear el array de deportes del activity selección de deportes.



**3.1.1.16. Clase utilidades.**

Esta clase se ha creado para añadir los métodos necesarios para realizar tratamientos de fechas, o cadenas.

- formatDate: Método que transforma la fecha guardada en base de datos como long a string con formato “dd/mm/yyyy”.
- convertDate: Método que transforma la fecha a long para almacenarla en base de datos
- getDescSport: este método se ha implementado para poder recuperar el nombre del deporte al pasarle como parámetro el id del tipo de deporte almacenado en base de datos.
- validarFecha: valida una fecha, la utilizamos cuando seleccionamos la fecha en el activity de selección de fechas.



**3.1.2. SharedPreferences.**

En android se puede almacenar información de diversas maneras, aunque para guardar pequeñas configuraciones o datos únicos, la plataforma pone a nuestra disposición la clase SharedPreferences.

SharedPreferences almacena información accesible desde cualquier actividad de la aplicación, y permanece almacenada a pesar de que salgamos de la misma.

En nuestra aplicación hemos optado por utilizar SharedPreferences para ir almacenando los datos de búsqueda de competiciones.

En las clases ActivityInicio se comprueba si existe instanciada como SharedPreferences la variable "CONFI", para saber si se han configurado o no los filtros de búsqueda de competiciones, y permitir así pulsar o no el botón de "Consultar Lista Competiciones".

```
// Creamos la instancia de "SharedPreferences" en MODE_PRIVATE
// RECUPERAMOS la variable CONFIG que nos indica si ya se ha configurado la APP.
SharedPreferences settings = getSharedPreferences("PREFERENCES", Context.MODE_PRIVATE);
String conf_ini = settings.getString("CONFI", "NO");
```

En la clase ActivitySelectSport se añade a la instancia SharedPreferences la variable "sport" que contiene los deportes seleccionados en esta actividad, para realizar la búsqueda de competiciones.

```
//save the task list to preference
SharedPreferences settings = getSharedPreferences("PREFERENCES", Context.MODE_PRIVATE);
Editor editor = settings.edit();
editor.putString("sport", paramSport);
editor.commit();
```

En la clase ActivitySelectUbi se añade a la instancia SharedPreferences la variable "provincia" que contiene la provincia seleccionada para realizar la búsqueda de competiciones, provincia que ha sido seleccionada a través la lista desplegable de este actividad.

```
String selec=cmbOpciones.getSelectedItem().toString();
//save the task list to preference
SharedPreferences settings = getSharedPreferences("PREFERENCES", Context.MODE_PRIVATE);
Editor editor = settings.edit();
editor.putString("provincia", selec);
editor.commit();
```

En la clase ActivitySelectFechas se añade a la instancia SharedPreferences las variables "fechaDesde" y "fechaHasta" con las fechas seleccionadas en esta actividad para realizar la búsqueda de competiciones.

```
//Almacenamos en PREFERENCES las fechas desde y hasta
SharedPreferences settings = getSharedPreferences("PREFERENCES", Context.MODE_PRIVATE);
Editor editor = settings.edit();
editor.putLong("fechaDesde", inicio_Timelong);
editor.putLong("fechaHasta", hasta_Timelong);
editor.commit();
```

En la clase ActivityLista se recuperan las instancias de SharedPreferences y además se modifica la variable "Confi", para que la aplicación quede informada de que ya se han configurado los datos de búsqueda, y así cuando se acceda de nuevo a la APP no haya que volver a configurar la búsqueda de competiciones, simplemente se acceda a la consulta de lista de competiciones.

```
SharedPreferences settings = getSharedPreferences("PREFERENCES", Context.MODE_PRIVATE);
String conf_ini = settings.getString("CONFI", "NO");
String filtro_sport = settings.getString("sport", "");
String filtro_prov = settings.getString("provincia", "NO");
long filtro_fecDes = settings.getLong("fechaDesde", 00000000);
long filtro_fecHas = settings.getLong("fechaHasta", 00000000);
```

```
Editor editor = settings.edit();
editor.putString("CONFIG", "OK");
editor.commit();
```

Con estos datos crearemos un objeto filtro de la clase Misfiltros para llamar al método que realiza la consulta en la base de datos, pero esto ya se ha explicado en la clase Misfiltros.

En esta clase ActivityLista también se definen otras SharedPreferences para poder hacer llegar a la pantalla activityActividad el modo en que accedemos a ella (el modo lo hemos implementado para realizar pruebas de altas de competiciones) y el id de la competición a la que accedemos.

```
private void visualizar(long id){
// Llamamos a la Actividad indicando el modo visualización y el identificador del registro
Intent i = new Intent(ActivityLista.this, ActivityActividad.class);
i.putExtra(C_MODO, C_VISUALIZAR);
i.putExtra(TrailsSportDbAdapter.C_COLUMNNA_ID, id);
startActivityForResult(i, C_VISUALIZAR);
overridePendingTransition(R.anim.in_animation_right, R.anim.out_animation_left);
}
```

Desde la pantalla ActivitiActividad se recupera el id de la competición seleccionado para realizar la consulta del registro que se va a mostrar.

```
// Obtenemos el identificador del registro si viene indicado
if (extra.containsKey(TrailsSportDbAdapter.C_COLUMNNA_ID)) {
id = extra.getLong(TrailsSportDbAdapter.C_COLUMNNA_ID);
consultar(id);
}
```

### 3.1.3. Animaciones o transiciones entre pantallas.

Cuando estamos desarrollando y creando una aplicación, el diseño es algo que tiene que ser atractivo a la vez que intuitivo, esto cuenta bastante, ya que es lo que entra por los ojos del usuario. Una de las cosas que hace que nuestra aplicación sea más atractiva son las transiciones entre pantallas o Activity.

En nuestro caso hemos optado por una transición al pulsar alguno de los botones que nos llevan a las siguientes pantallas, haciendo que la nueva pantalla entra por la derecha al mismo tiempo que la actual se marcha por la izquierda. Y en los casos de pulsar el botón de retorno (o atrás) del móvil, hemos implementado justo lo contrario, para que el movimiento vaya acorde con la transición de pantallas.

La verdad que es una cosa que queda muy bien en la aplicación y es muy sencillo de hacer.

Para hacer esto basta con añadir justo después del StartActivity, en nuestro caso lo hemos llamados StartActivityResult, la siguiente línea cuando pulsamos el botón siguiente:

```
overridePendingTransition(R.anim.in_animation_right, R.anim.out_animation_left);
```

Y cuando pulsamos el botón de volver o atrás del móvil lo añadimos en el método **onBackPressed** que es el evento de dicho botón:

```
public void onBackPressed(){
    super.onBackPressed();
    overridePendingTransition(R.anim.in_animation_left, R.anim.out_animation_right);
}
```

En ambos casos lo que se hace es llamar con la función **overridePendingTransition** a dos XML alojados en la carpeta **“res/anim”**.

**Estos XML, son los siguientes:**

El fichero XML **“in\_animation\_left”**:

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="300"
    android:fromXDelta="-100%"
    android:toXDelta="0%" />
```

El fichero XML **“out\_animation\_right”**:

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="300"
    android:fromXDelta="0%"
    android:toXDelta="100%" />
```

El fichero XML **“in\_animation\_right”**:

```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="300"
    android:fromXDelta="100%"
    android:toXDelta="0%" />
```

El fichero XML **“out\_animation\_left”**:

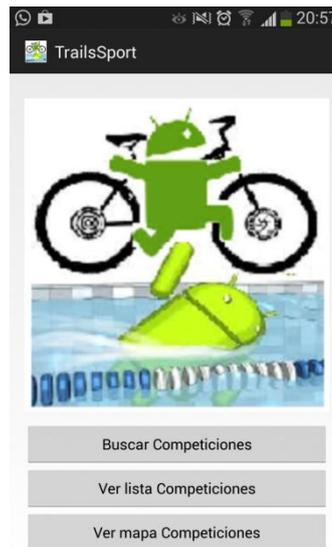
```
<?xml version="1.0" encoding="utf-8"?>
<translate xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="300"
    android:fromXDelta="0%"
    android:toXDelta="-100%" />
```

Solo mencionar de estos archivos que el atributo **duration** es el que indica el tiempo que tarda en pasar de una transición a otra en milisegundos.

## 3.2. Prototipos finales de interfaz de Usuario.

### 3.2.1. Pantalla de Inicio.

Al acceder a la aplicación aparece esta pantalla, la primera vez solo se podrá pulsar el primer botón para configurar la búsqueda de competiciones, al pulsar algún otro botón se informa mediante un mensaje toast, que primero hay que acceder a configurar las opciones de búsqueda.



Esta pantalla consta de tres botones:

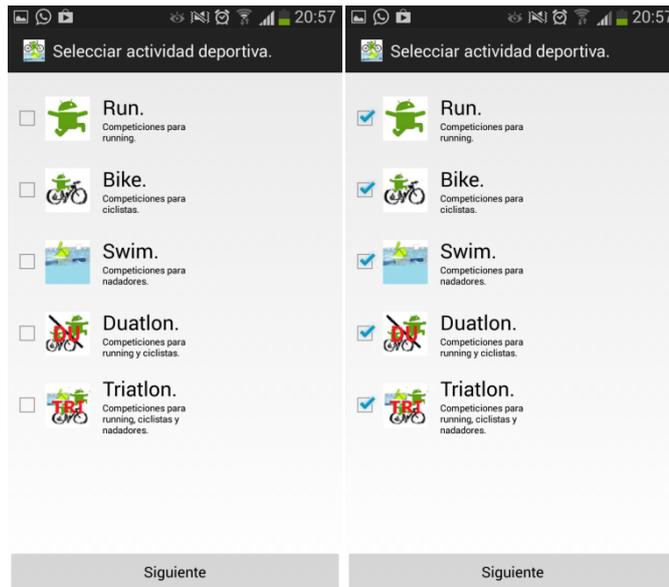
Buscar competiciones: El primero de ellos es el que da acceso a la configuración de búsqueda de competiciones.

Ver lista competiciones: El segundo de ellos da acceso al listado de búsqueda, si ya se ha configurado previamente, si no es así se informa de ello mediante un mensaje toast.

Ver mapa competiciones: El último botón daría acceso al listado de competiciones, pero en modo mapa, aunque esta opción no está disponible, ya que no hemos podido implementarla, por lo que la aplicación nos informa de ello mediante un mensaje toast.

### 3.2.2. Pantalla de selección de deportes(o tipos de competiciones).

Esta pantalla es la que permite realizar la selección de deportes o tipos de competiciones, como podemos ver el formato final es muy parecido al prototipo, solo que hemos eliminado las subcategorizas de distancias, ya que no nos ha parecido muy práctico de momento.



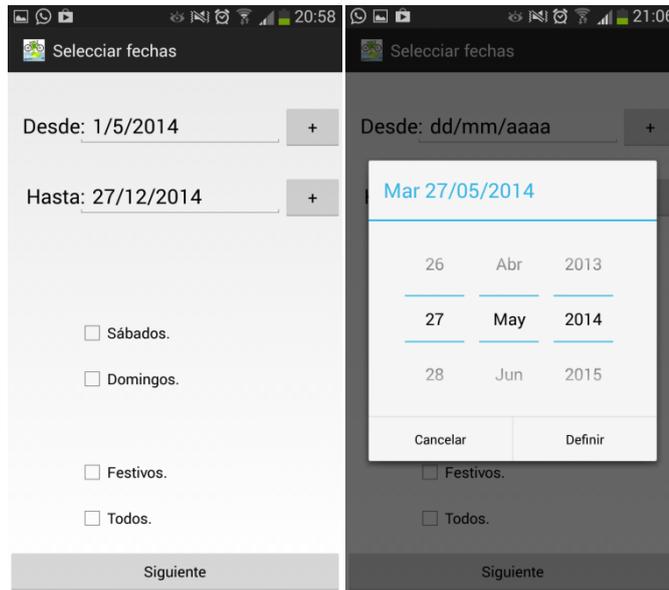
Esta pantalla no permite seguir adelante si no se ha seleccionado ninguna opción. Informará de ello mediante un mensaje toast.

**3.2.3. Pantalla de selección de Provincia.**



Esta pantalla permite seleccionar la provincia, para la que se quieren buscar las competiciones. El check “Activar localización por GPS” no está disponible en esta versión porque no se ha implementado.

**3.2.4. Pantalla de selección de fechas.**



Esta pantalla permite seleccionar el rango de fechas, para el cual se buscarán las competiciones. Los check no están disponibles en esta versión porque no se han implementado.

**3.2.5. Pantalla de selección de notificaciones.**



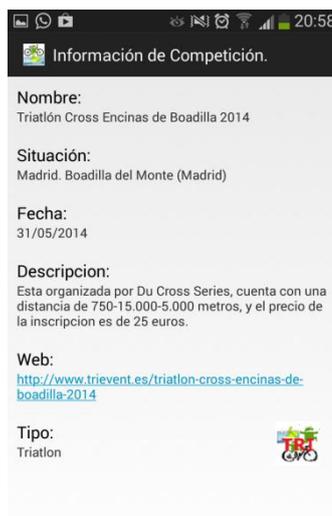
Esta pantalla es meramente de transición, ya que no se ha implementado la funcionalidad de los check.

### 3.2.6. Pantalla lista competiciones.



Esta pantalla es la encargada de mostrar el resultado de la búsqueda realizada de competiciones, informa a modo resumen de las mismas, con la fecha de realización, entre paréntesis una abreviatura del tipo de competición y su nombre. Esta lista esta ordenada por fecha de realización.

### 3.2.7. Pantalla detalle competición.



Esta pantalla es la encargada de mostrar la información de la competición seleccionada. En ella se muestra el nombre, situación que incluye provincia y municipio, fecha de realización, breve descripción, enlace a su página web y tipo de competición con su icono correspondiente.

Cuando se pulsa sobre la Url de la página web, de la competición, esta se abrirá en el explorador que tenga el dispositivo android configurado por defecto.



#### 4. Conclusiones y futuro de la APP.

En el presente proyecto se ha implementado con éxito una aplicación para móviles Android que permite realizar consultas sobre competiciones deportivas, carreras running, de bicicleta, de natación, duatlones y/o triatlones. El proyecto es escalable y el código reutilizable, aunque necesita urgentemente alojar los datos de las competiciones de forma externa y poder atacarlos a través de un web servid.

Se ha utilizado una base de datos sencilla y eficaz implementada en SQLite3, al estar en local no ha influido mucho que las consultas de la provincia se hagan por el nombre de la provincia y no por un id, aunque esto es algo que tendría que cambiarse una vez se implemente la base de datos y está se aloje en red.

Desde el punto de vista personal ha sido muy satisfactorio realizar este proyecto, por una parte me ha permitido poner en práctica innumerables conceptos aprendidos durante la carrera y por otra la realización del TFC para mi, pretendía ser el inicio de un proyecto más grande que me ofreciera no sólo lecciones sino también ejercicios, contacto con profesores y el contacto con distintos usuarios, y creo que he conseguido todo esto y con creces.

Además para mí personalmente ha sido increíble aprender cómo funciona una aplicación móvil y todo lo que ello conlleva, estoy deseando poder dedicarle tiempo a mi aplicación y hacer de ella una buena APP para que sea usada por todos los deportistas como yo.

En el análisis DCU de la APP la mayoría de las personas que se dedican a realizar competiciones de este tipo, han comentado que está bastante bien, solo falta implementar una base de datos y su alimentación para que se pueda actualizar automáticamente, y entonces será todo un éxito.

Para el futuro de la App espero terminar la carrera en junio y este verano poder dedicarle tiempo para que en septiembre esté disponible para todos los usuarios que la quieran. Las mejoras que tengo en mente son las siguientes:

- La primera de ellas y las más importantes es la de crear una base de datos alojada en la red y acceder a ella a través de un web servid.
- Otra mejora bastante importante para mi es conseguir mostrar las competiciones en un mapa.
- Implementar el check de GPS en la pantalla de selección de provincia para que busque además por localización.
- Implementar los check de la pantalla de selección de fechas, de sábado, domingo, festivos y todos.
- Implementar todos los check de la pantalla de notificaciones y por tanto las notificaciones automática y las actualizaciones automáticas.
- Contactar con organizadores de eventos (competiciones) de este tipo y presentarles la APP, para estudiar si es factible crear un acceso para que ellos mismos puedan dar de alta los eventos o actividades.

## 5. Anexos.

En este punto voy aprovechar para hacer mención de algunas cosas que he dejado en el código de la entrega de la pec3 que deberían haberse suprimido porque al final no se han implementado o solo se han usado para pruebas:

- Activity\_main.xml. → Es el principal que se había creado por defecto con el típico hola mundo.
- La clase actividad → Ya lo he comentado en su explicación que esta clase se utilizará en futuras mejoras, por ello no la he suprimido.

## 6. Bibliografía.

### Libros

La Biblia de Java 2, Steven Holzner (2000), Ediciones Anaya Multimedia (Grupo Anaya S.A.).

Java 2 J2SE 1.4, F. Javier Moldes Teo. (2003), Ediciones Anaya Multimedia (Grupo Anaya S.A.).

### Recursos web

Guías de desarrollo oficiales de Android: <http://developer.android.com/develop/index.html>

Blogs de interés (ejemplos):

<http://www.aprendeandroid.com/menu.htm>

<http://elbauldeandroid.blogspot.com.es/>

Preguntas y respuestas de Stackoverflow:

<http://stackoverflow.com/>