



Título del Proyecto:

Carrendario

Autor:

Vidal Salvador,
Jose

Director:

Tomás Gironés,
Jesús

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

**Diploma de Especialización en
Desarrollo de Aplicaciones para
Android**

Septiembre del 2015



Contenido

Título del Proyecto:	1
Autor:	1
Director:	1
Diploma de Especialización en Desarrollo de Aplicaciones para Android	1
Introducción	3
Descripción del problema	3
Objetivos	3
Motivación	3
Situación de... / Tecnologías utilizadas.....	3
Arquitectura de la aplicación	5
Esquema del diseño	5
Modelo de datos	5
Vistas	9
Conclusiones	9
Anexos.....	10
Listado de fuentes entregadas / Código fuente en GitHub	10
Manual de usuario	Error! Bookmark not defined.

Introducción

Descripción del problema

Es un hecho que las competiciones de running y otros deportes que se practican al aire libre están de moda. En los últimos años, se ha multiplicado tanto el número de eventos de esta índole como el número de usuarios que asisten a ellos. Por este motivo, resulta de gran utilidad poder en todo momento disponer de una fuente de información que nos suministre detalles acerca de las próximas competiciones en las que poder participar.

En la actualidad, ya existen en el mercado aplicaciones móviles que se encargan de esta tarea. No obstante, estas aplicaciones son, o complicadas de utilizar, o con un diseño poco amigable.

Objetivos

Partiendo de la problemática expuesta en el punto anterior, el objetivo principal que se pretende acometer es el de llegar a desarrollar una aplicación Android que sea capaz de mostrar información relativa a competiciones deportivas de forma sencilla e intuitiva.

Motivación

La principal motivación que ha impulsado la realización de este proyecto ha sido la de poder llegar a desarrollar una aplicación que permita, a la gran cantidad de deportistas que participan en competiciones deportivas, visualizar de forma rápida y sencilla las próximas competiciones a las que pueden asistir. Pudiendo ‘agendar’ éstas a su Google Calendar si lo desean.

Del mismo modo, por medio de esta aplicación, se intentará plasmar la mayor parte de los conocimientos adquiridos a lo largo del curso.

Situación de... / Tecnologías utilizadas

En cuanto a las tecnologías utilizadas, estas han sido:

- Capa frontend:
 - Android. Sistema operativo para dispositivos móviles
 - Caldroid (<https://github.com/roomorama/Caldroid>). Calendario de android con más opciones del que ofrece Android por defecto.



- Android-floating-action-button (<https://github.com/futuresimple/android-floating-action-button>). Librería para mostrar un menu compuest de FAB buttons.
- GSON. Librería de transformación JSON-Objetos java.
- Java. Lenguaje de programación de propósito general orientado a objetos desarrollado por Sun Microsystems. Sus características principales son:
 - Simple
 - Orientado a Objetos
 - Distribuido
 - Interpretado
 - Robusto
 - Seguro
 - Multihilo
 - Portable
 - Dinámico
- Capa backend:
 - Java
 - MyBatis(<https://mybatis.github.io/mybatis-3/es/>). Framework de persistencia
 - Spring (<https://spring.io/>). Spring es un marco de trabajo cuya finalidad es facilitar el desarrollo de aplicaciones con Java
 - JSoup (<http://jsoup.org/>). Librería java para realizar tareas de extracción de datos de páginas web.
 - Jackson. Librería utilizada para permitir que los servicios REST retornen los resultados en formato JSON.
 - Openshift (<https://openshift.redhat.com>). Servidor web gratuito en la nube.
 - Sistema de gestión de bases de datos MYSQL y phpMyAdmin. MySQL ha sido el sistema elegido para gestionar la base de datos de este proyecto y la herramienta phpMyAdmin para manejar la administración de MySQL vía web.
 - Maven. Herramienta open-source, que se creó en 2001 con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente).
- Entorno de programación Android Studio. Entorno de desarrollo integrado para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

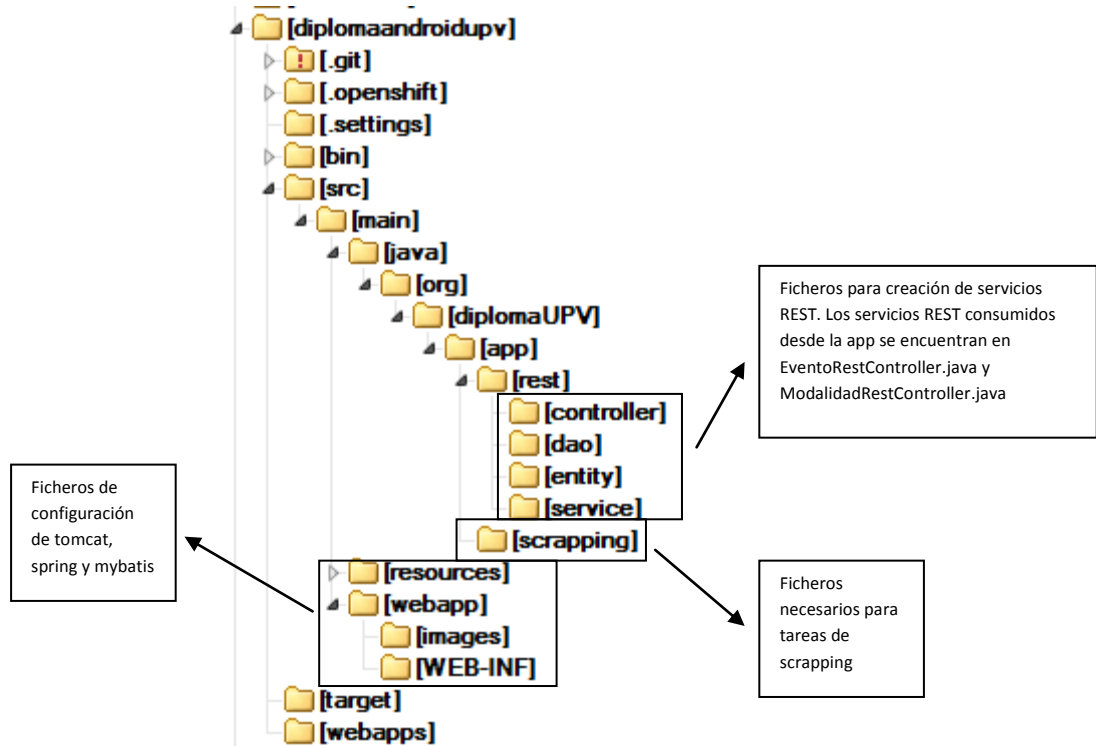
Arquitectura de la aplicación

Esquema del diseño

Para el diseño de la aplicación se ha hecho uso del patrón MVC (Modelo-Vista-Controlador). Propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario y se potencia la facilidad de mantenimiento, reutilización de código y la separación de conceptos. Por lo que respecta a nuestro proyecto, nuestras tres capas serán:

- Capa de datos: En la aplicación utilizaremos una base de datos gestionada por el servidor MySQL alojándose éste en una cuenta de Openshift (<https://openshift.redhat.com>).
- Capa de presentación: Se implementará esta capa por medio de una aplicación Android a través del entorno de desarrollo Android Studio. Las principales clases y layouts implicados son:
 - **MainActivity**: Activity encargada de mostrar y configurar la pantalla principal de la aplicación. Layout asociado `activity_main.xml`
 - **FilterOptionsActivity**: Activity encargada de mostrar la pantalla de filtrado avanzado. Layout asociado `activity_filter_competitions.xml`
 - **ListCompetitionsActivity**: Activity encargada de mostrar la lista de carreras de una modalidad determinada. Layouts asociados:
 - `activity_list_competitions.xml`
 - `activity_calendar_competitions.xml`
 - `activity_gmap_competitions.xml`
 - **SettingsActivity**: Activity encargada de mostrar la pantalla de configuración de la aplicación.

- Capa de negocio: La capa de presentación se ‘nutrirá’ de información gracias a la invocación de una serie de servicios web alojados también en la cuenta de Openshift. Esta se estructura de la siguiente manera:



Modelo de datos

La aplicación únicamente dispone de dos tablas * que son:

- Tabla con información relativa a Eventos o Competiciones (EVENTO)



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	1 ID	bigint(12)			No	Ninguna	AUTO_INCREMENT
<input type="checkbox"/>	2 TITULO	varchar(250)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	3 ID_DISCIPLINA	bigint(12)			Sí	NULL	
<input type="checkbox"/>	4 ID_MODALIDAD	bigint(12)			Sí	NULL	
<input type="checkbox"/>	5 ID_COMUNIDAD_AUT	bigint(12)			Sí	NULL	
<input type="checkbox"/>	6 LOCALIDAD	varchar(250)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	7 LOCALIDAD_GMAPS	varchar(250)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	8 URL_EVENTO	varchar(250)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	9 FECHA	date			Sí	NULL	
<input type="checkbox"/>	10 INFO_EXTRA	varchar(250)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	11 FECHA_CREACION	date			Sí	NULL	
<input type="checkbox"/>	12 URL_CARTEL	varchar(1024)	latin1_swedish_ci		Sí	NULL	
<input type="checkbox"/>	13 PUNTUACION_POS	bigint(12)			Sí	NULL	
<input type="checkbox"/>	14 PUNTUACION_NEG	bigint(12)			Sí	NULL	

- Tabla con información de las distintas modalidades de competición disponibles (MODALIDAD).

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	1 ID	bigint(12)			No	Ninguna	AUTO_INCREMENT
<input type="checkbox"/>	2 DESCRIPCION	varchar(250)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	3 DISTANCIA	varchar(250)	latin1_swedish_ci		No	Ninguna	
<input type="checkbox"/>	4 ID_DISCIPLINA	bigint(12)			Sí	NULL	

La relación entre ambas tablas se realiza por medio de la foreign key ID_MODALIDAD de la tabla EVENTO

* Existen otras tablas (tabla de comentarios) en el esquema que se han creado con vistas a posibles mejoras que se detallan en el apartado de conclusiones pero que no se han desarrollado todavía.

Como hemos remarcado en el punto anterior, para que la capa de presentación pueda mostrar toda aquella información almacenada en la capa de datos, será necesario que ésta haga uso de una serie de servicios web. Estos serán:

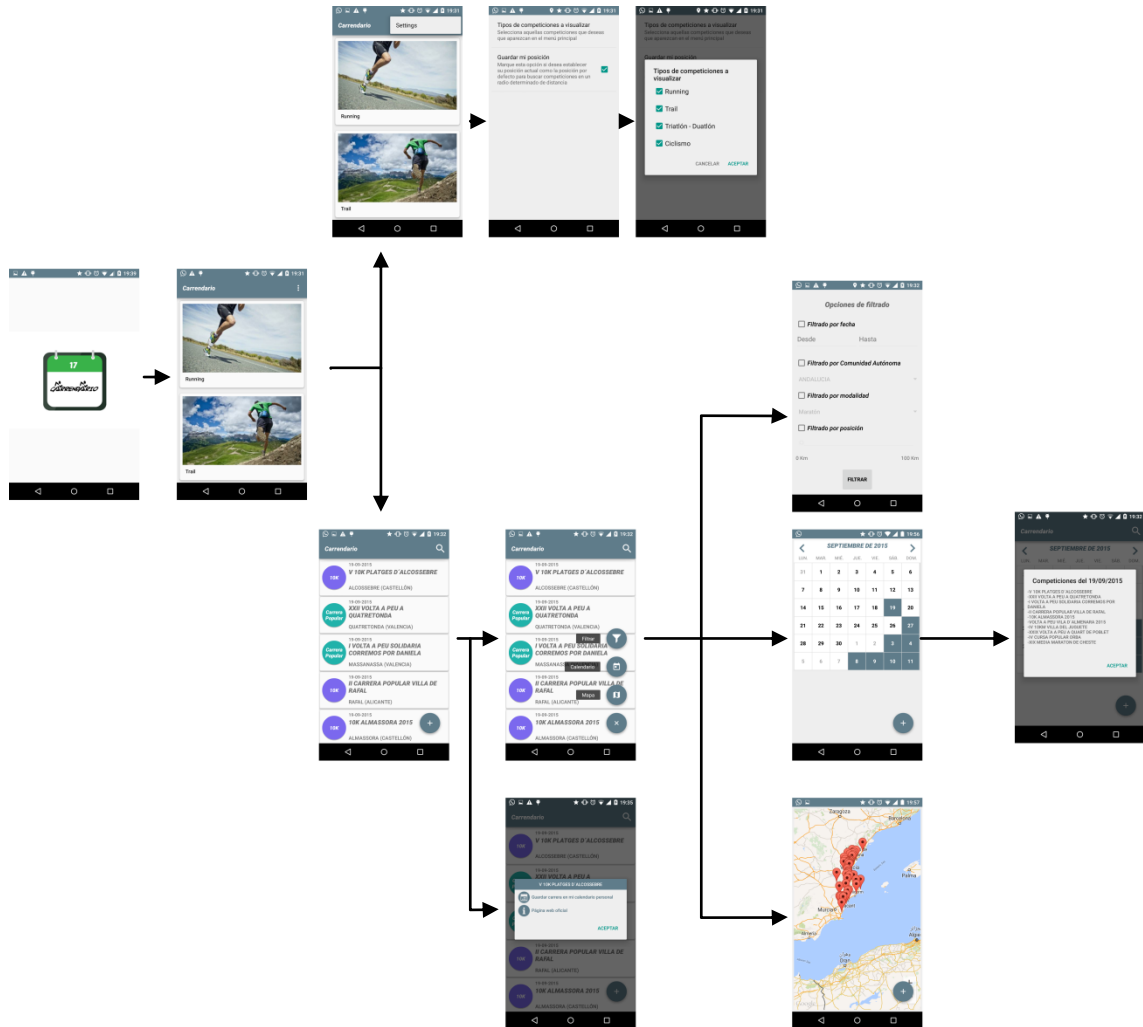
- SW para obtención de listado de eventos:
 - o <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/eventos/getAll>. Retorna un listado de todos los eventos disponibles en la base de datos en formato JSON
 - o <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/eventos/get>. Admite parámetros de entrada de manera que muestra aquellos eventos de la base de datos que cumplen con dichos parámetros de entrada especificados por medio de la url. Los parámetros disponibles son:
 - id



- titulo
- idDisciplina
- idModalidad
- localidad
- idComunidadAutonoma
- localidadGmaps
- urlEvento
- fecha
- infoExtra
- fechaCreacion
- urlCartel
- puntuacionPositiva
- puntuacionNegativa
- <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/eventos/insert> . Servicio REST para creación de eventos
- <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/eventos/update>. Servicio REST para modificación de eventos. Admite como parámetros el id del evento a modificar y el objeto Evento con los nuevos valores a establecer.
- SW para obtención del listado de modalidades
 - <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/modalidades/getAll>. Retorna el listado de todas las modalidades disponibles en la base de datos en formato JSON
 - <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/modalidades/get> . Admite parámetros de entrada de manera que muestra aquellas modalidades de la base de datos que cumplen con dichos parámetros de entrada especificados por medio de la url. Los parámetros disponibles son:
 - Id
 - descripción
 - distancia
 - idDisciplina
 - <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/modalidades/insert> . Servicio REST para creación de modalidades
 - <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/modalidades/update>. Servicio REST para modificación de modalidades. Admite como parámetros el id de la modalidad a modificar y el objeto Modalidad con los nuevos valores a establecer.
- SW de invocación a tareas de Scapping sobre páginas de competiciones deportivas.
 - <http://diplomaandroidupv-jovisal1.rhcloud.com/rest/scrappingEventos/CarrerasPopulares>. Invoca al proceso de obtención de la información de competiciones almacenada en la página web de carreraspopulares (<http://www.carreraspopulares.com/>)



Vistas



Conclusiones

El objetivo principal de la aplicación, que era la de mostrar de forma sencilla el listado de competiciones a poder realizar por usuarios deportistas, se ha cumplido. El usuario rápidamente puede disponer de un listado de carreras así como la posibilidad de 'agendar' éstas a su calendario personal. No obstante, quedan muchas líneas abiertas en el desarrollo de la aplicación. Motivo de ello es que ya existan tablas, entidades y otros desarrollos que, aunque ahora mismo no se utilicen, se pretende que en futuros desarrollos estén presentes. Las principales líneas a desarrollar serían:

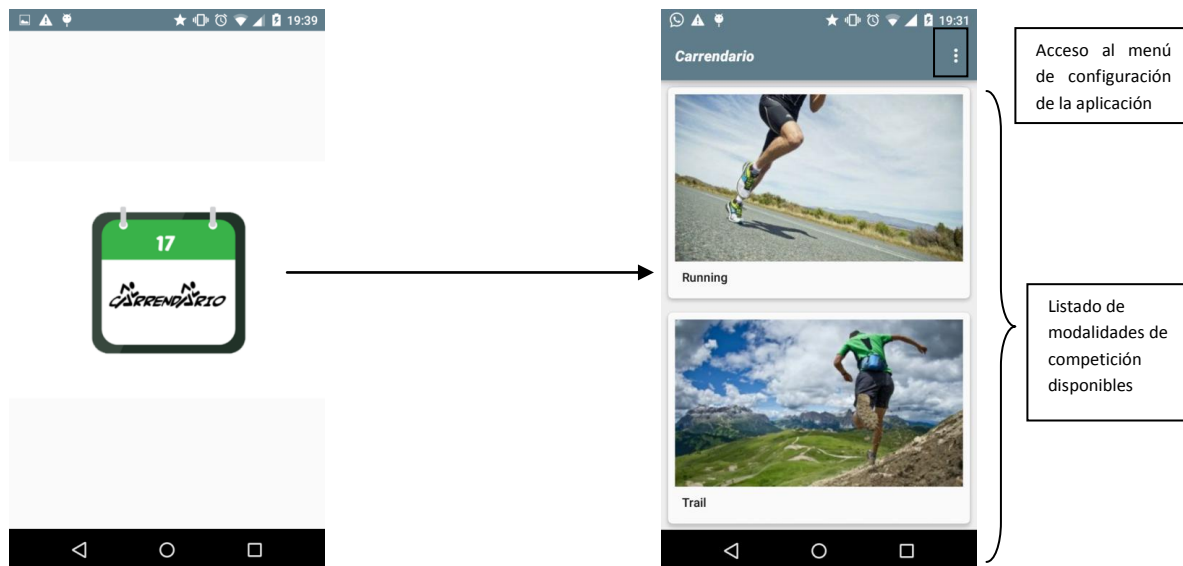
- Añadir la posibilidad de valorar positiva y negativamente una competición.
- Añadir la posibilidad de comentar una competición
- Posibilidad de compartir la información de una competición en redes sociales.
- Añadir un listado de clasificaciones asociadas a las competiciones.
- Añadir sistema de notificaciones en función de unos parámetros determinados (cercanía, tipo de competición, valoraciones...)
- Añadir la posibilidad de modificar el estilo de la aplicación.

Anexos

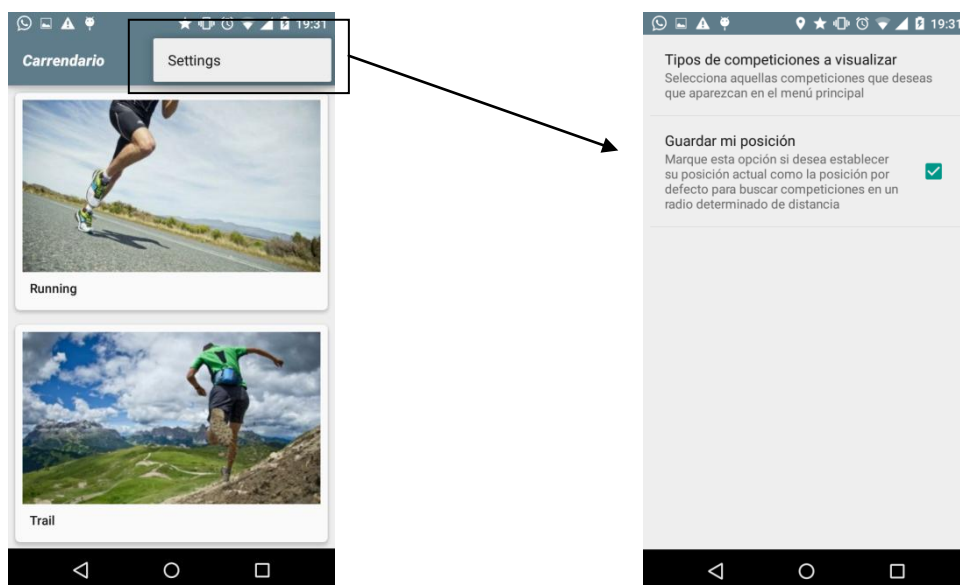
Listado de fuentes entregadas / Código fuente en GitHub

- `Codigo_fuente.zip`. Zip con que contiene:
 - `CompeticionesDeportivas_backend.zip`. Código del proyecto que implementa los servicios.
 - `CompeticionesDeportivas_frontend.zip`. Código del proyecto que implementa la capa de vista.
 - `Diplomaandroidupv.sql`. Script de base de datos.

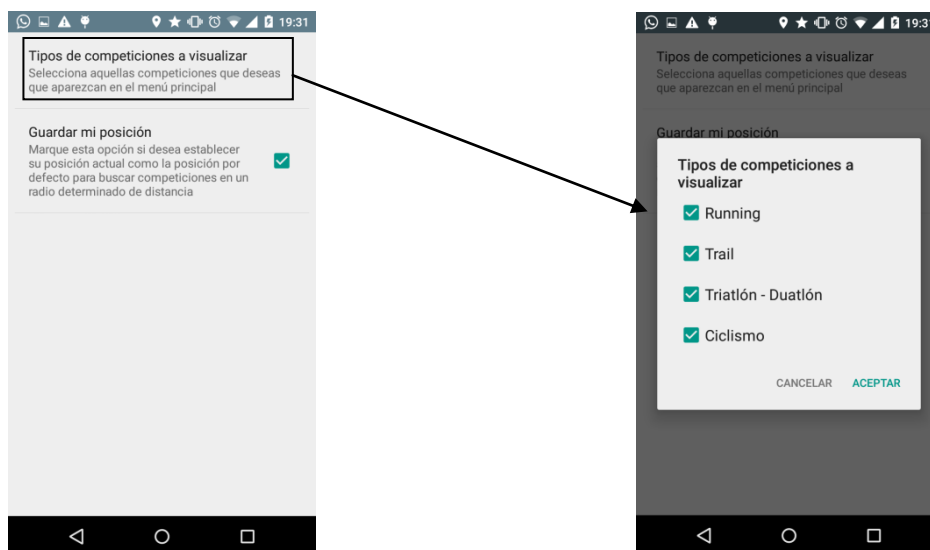
- Pantalla principal. Tras una primera pantalla de splash con el logo de la aplicación, se le muestra al usuario la pantalla principal de la aplicación



- Si el usuario selecciona la opción de menú settings, accede a la pantalla de configuración de la aplicación



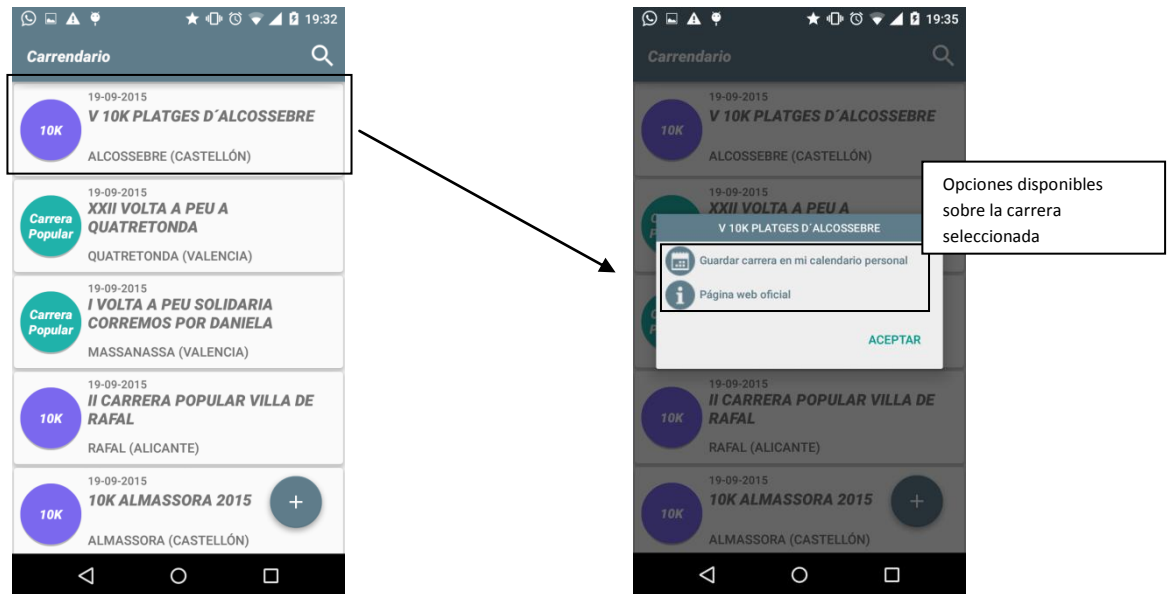
- Dentro de la ventana de configuración, el usuario dispone de dos opciones. La primera para establecer aquellas modalidades de competición a visualizar en el menú principal, y en segundo lugar, establecer su posición geográfica a utilizar en posibles tareas de filtrado posteriores.



- Listado de competiciones: Una vez el usuario selecciona la modalidad de competición en la pantalla principal, accede a un segundo listado en dispone de todas las carreras o eventos de ese tipo a los que puede asistir.



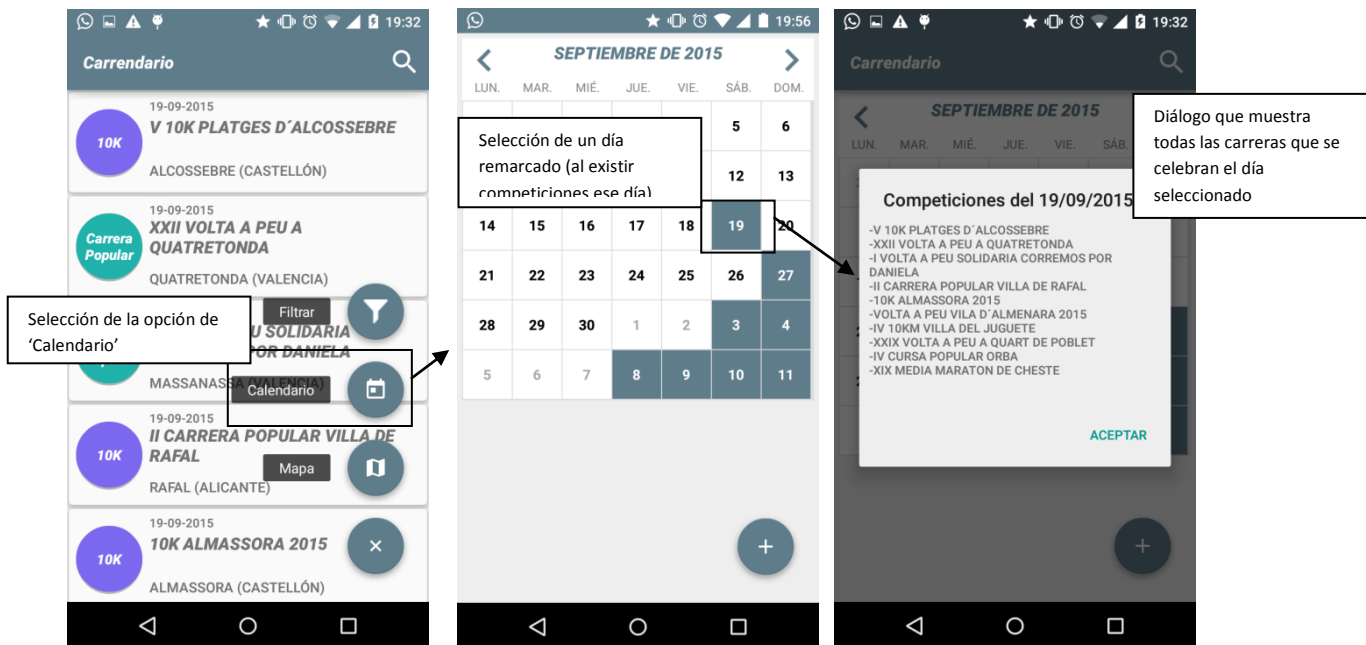
- Al presionar de forma prolongada sobre una carrera, se le muestra al usuario un diálogo desde el que puede, ‘agendar’ la carrera en su calendario personal, o bien acceder a la página web oficial de la carrera (en caso de que ésta disponga de una).



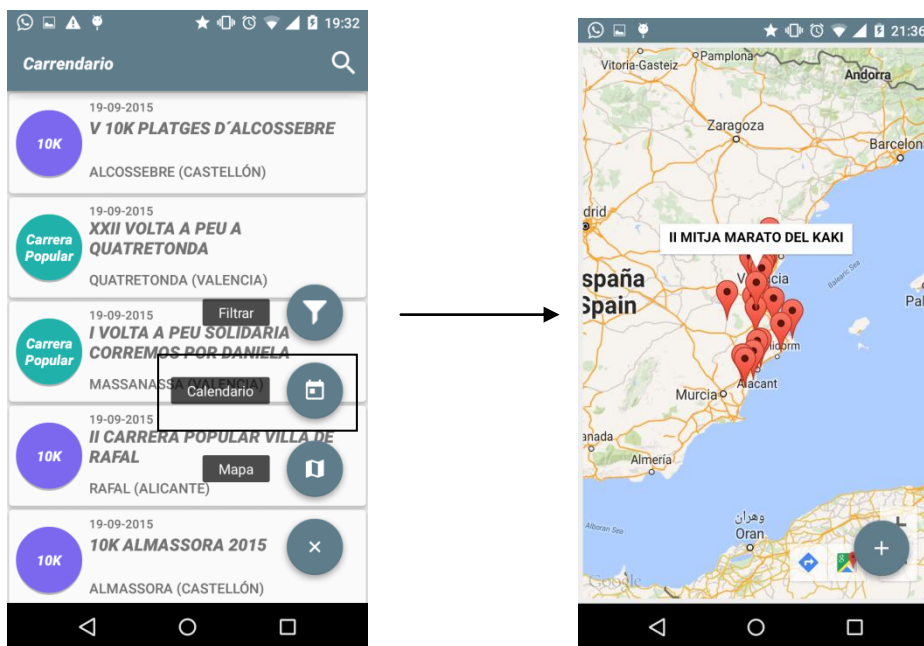
- Menú principal del listado de carreras. Una vez el usuario se encuentra en la pantalla que lista las competiciones o carreras disponibles, dispone de un menú desplegable que le permitirá:
 - Cambiar la visualización a forma de calendario.
 - Cambiar la visualización a forma de mapa.
 - Volver a visualizar el listado.
 - Acceder a la pantalla de opciones de filtrado avanzado.



- Acceso al listado de carreras en la vista de calendario. Una vez el usuario se encuentra en la pantalla que lista las competiciones o carreras disponibles, seleccionando la opción de calendario podrá visualizar las competiciones marcadas sobre un calendario.



- Acceso al listado de carreras en la vista de mapa. Una vez el usuario se encuentra en la pantalla que lista las competiciones o carreras disponibles, seleccionando la opción de mapa podrá visualizar las competiciones marcadas sobre un mapa de google maps.



- Acceso a la pantalla de opciones de filtrado avanzado. Una vez el usuario se encuentra en la pantalla que lista las competiciones o carreras disponibles, seleccionando la opción de filtro podrá acceder a una pantalla de filtrado en la que seleccionar una serie de criterios de búsqueda de las competiciones.

