



TrainerPlan: Tu plan de entrenamientos en tu móvil

Nombre Estudiante: Aitor López de Aberasturi Ortiz de Pinedo
Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Nombre Consultor/a: Eduard Martin Lineros

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

10 de Octubre de 2017



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>TrainerPlan: Tu plan de entrenamientos en tu móvil.</i>
Nombre del autor:	<i>Aitor López de Aberasturi Ortiz de Pinedo</i>
Nombre del consultor/a:	<i>Eduard Martin Lineros</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	01/2017
Titulación::	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>TrainerPlan Plan entrenamiento</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.</i></p>	
<p>La práctica deportiva se está convirtiendo en la principal actividad de ocio en nuestra sociedad. La planificación ayuda a los deportistas profesionales y amateurs a conseguir sus objetivos.</p> <p>Dentro del TFM vamos a dar una solución a la consecución de esos objetivos creando una aplicación. Para ello se ha realizado una planificación con una fase de diseño, fase de implementación y entrega final.</p> <p>El resultado de este TFM es TrainerPlan, una aplicación móvil desarrollada para terminales Android que va a permitir la generación y consulta de planes de entrenamiento que van a poder ser compartidos por diferentes usuarios.</p> <p>El patrón que se ha utilizado para el diseño de la arquitectura de nuestra aplicación ha sido el MVC (modelo vista controlador). Además nuestro sistema dispone de una Arquitectura SOA (Arquitectura orientada a servicios).</p> <p>La tecnología que se ha utilizado para el desarrollo de esta arquitectura ha sido un cliente desarrollado con Android Studio, que realiza peticiones sobre un servidor LAMP (Linux Apache Mysql PHP).</p> <p>El proyecto nos ha ayudado a realizar una aplicación real aplicando los contenidos aprendidos en otras asignaturas del Master. La planificación del proyecto nos ha ayudado a que los desvíos sobre los objetivos y diseño inicial</p>	

sean mínimos.

Abstract (in English, 250 words or less):

Sports practice is becoming the main leisure activity in our society. Planning helps professional and amateur athletes achieve their goals.

Within the TFM we will give a solution to the achievement of those objectives by creating an application. For this, a planning has been carried out with a design phase, implementation phase and final delivery.

The result of this TFM is TrainerPlan, a mobile application developed for Android mobiles that will allow the generation and consultation of training plans that can be shared for others users.

The pattern that has been used to design the architecture of our application has been the MVC (model view controller). In addition our system has an SOA (Service Oriented Architecture) architecture.

The technology that has been used for the development of this architecture has been a client developed with Android Studio, that makes requests on a LAMP server (Linux Apache Mysql PHP).

The project has helped us to make a real application applying the contents learned in other subjects of the Master. Project planning has helped us to make the deviations from the objectives and initial design minimal.

Índice de contenido

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	6
1.3 Enfoque y método seguido.....	8
1.4 Planificación del Trabajo.....	9
1.5 Breve resumen de productos obtenidos.....	11
1.6 Breve descripción de los otros capítulos de la memoria.....	12
2. Fase de Diseño.....	14
2.1. Análisis de los usuarios y contexto de uso.....	14
2.2. Diseño conceptual.....	22
2.3. Prototipado.....	28
2.4. Evaluación.....	32
2.5. Diseño de la base de datos.....	33
2.6. Diseño de la arquitectura.....	34
3. Implementación.....	38
3.1. Entorno de desarrollo.....	38
3.2. Desarrollo de la aplicación.....	41
3.3. Pruebas.....	50
3.4. Publicación.....	58
4. Valoración económica.....	60
4.1.- Costes de la aplicación.....	60
4.2.- Monetización de la aplicación.....	60
5. Conclusiones.....	61
6. Glosario.....	63
7. Bibliografía.....	64
8. Anexos.....	65

1. Introducción

1.1 Contexto y justificación del Trabajo

La práctica deportiva se está convirtiendo en la principal actividad de ocio en nuestra sociedad. La vida sedentaria o el salir de copas están siendo sustituidos por el **deporte** que además te ayuda a mantenerte física y mentalmente en forma. Deportes como el pádel o el running se han puesto de moda en nuestra sociedad.

El deporte ya no es cosa ni de edades ni de sexos: el 41% de los participantes de la maratón de Nueva York fueron mujeres y más del 80% mayores de 30 años. Basta con salir a los parques, pistas o montes para ver esta realidad, como ejemplo un entrenamiento del grupo de Running Fiz en el parque del Prado de Vitoria-Gasteiz:



En muchas ocasiones no sólo nos conformamos con practicar deporte, muchas veces nos animamos a **competir**, a participar en una carrera benéfica o en una popular como puede ser una San Silvestre, incluso los hay que deciden realizar una maratón, triatlones o una carrera de montaña. El número de participantes así como el de carreras en diferentes modalidades está incrementándose de forma exponencial.

La planificación a la hora de afrontar una competición para obtener unos objetivos es fundamental. Los **planes de entrenamiento** que siempre han utilizado los atletas profesionales han llegado a los deportistas amateur. Nos marcamos objetivos como terminar una maratón o bajar de un tiempo de 3 horas, y para ello lo ideal es planificar.

La **tecnología** está sufriendo grandes avances en la práctica deportiva. Los nuevos relojes-pulsómetros calculan nuestra velocidad mediante GPS; los SmartWatch están preparados para la práctica deportiva, calculan nuestra actividad y las pulsaciones sin necesidad de banda pectoral; y las pulseras de actividad y control del sueño, que nos proponen salir de nuestra vida sedentaria. Pero lo mejor está por llegar, el IoT (internet de las cosas o **Internet of Things**) será fundamental: Se crearán nuevos elementos como camisetas que nos controlarán nuestra frecuencia cardiaca, etc. Y por supuesto esa tecnología irá acompañada de **aplicaciones** que nos permitan monitorizar y gestionar esa información.

La idea de aplicación que planteo en el TFM es la de una aplicación que te permita consultar un plan de entrenamientos en el móvil para diferentes deportes (atletismo, triatlón, ciclismo, carreras de montaña...). El **tiempo de entrenamiento** es fundamental a la hora de realizar una planificación y conseguir un objetivo. Los deportistas a los que va dirigido no son profesionales, por lo que habrá que conocer el **tiempo libre** que dispone cada uno para poder practicarlo.

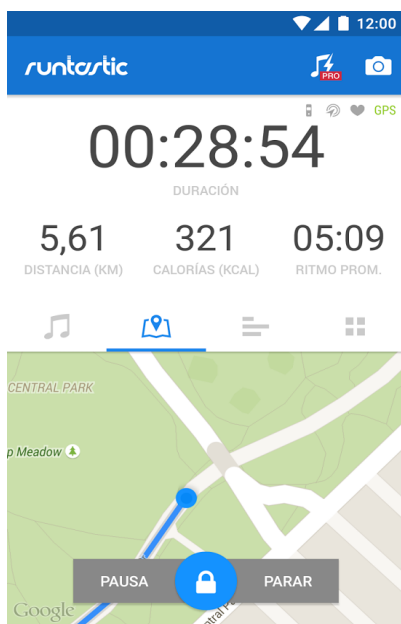
La idea de **TrainerPlan** surgió hace 4 años. Soy profesor de informática en ciclos formativos y anualmente suelo pensar en una idea para explicar a mis

La aplicación TrainerPlan la he planteado en el equipo **42195.es**, equipo al que pertenezco, y que en la actualidad cuenta con más de 2000 socios, muchos de ellos con licencias federativas en atletismo, triatlón y montaña.

[illegible]

- 3 -

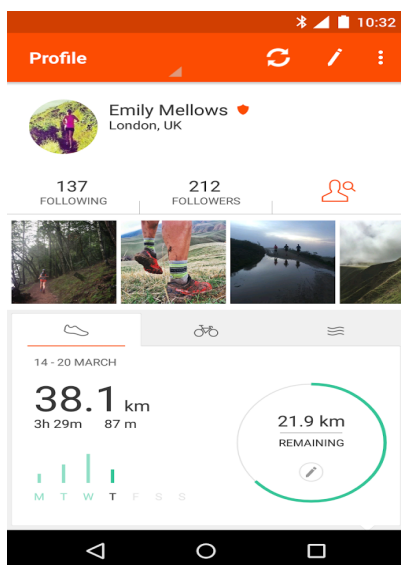
Existen muchas aplicaciones enfocadas a la práctica deportiva de diferentes deportes. Estos son algunos ejemplos:



- **Runastic:** mide la actividad deportiva, controla el tiempo real, el ritmo y te muestra el mapa por donde estás entrenando. Además te permite la carga de planes de entrenamiento standard.

Puntos fuertes: La gran cantidad de servicios que ofrece y la integración con otros dispositivos (Android Wear, Zapatillas).

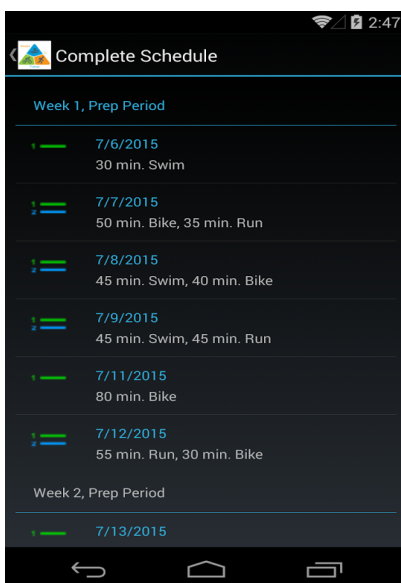
Puntos débiles: Está orientado a usuarios principiantes.



- **Strava:** Además de los servicios que puede ofrecer Runastic añade otros nuevos como crear rankings de distancias para ciclistas y corredores (ejemplo el ranking de la subida al Naranco).

Puntos fuertes: Está enfocado a todo tipo de deportistas.

Puntos débiles: No dispone de un servicio de planes de entrenamiento.



- **TriTrainer:** obtiene un entrenamiento para triatletas a partir de unos datos de entrada.

Puntos débiles: Se basa en planes de entrenamiento solo de triatlón. Es una aplicación de pago.

Muchas de estas aplicaciones hacen un seguimiento de los entrenamientos, calorías quemadas, recorrido realizado, etc. Además existen algunas aplicaciones que ofrecen planes de entrenamiento, en la mayoría de los casos planes de iniciación standard. La gente diseña sus planes de entrenamiento o adapta los planes de entrenamiento de Internet o de su entrenador. Los planes de entrenamiento no tienen en cuenta la disponibilidad de los deportistas. ¿Por qué no adaptar el plan de entrenamientos al deportista en lugar de que el deportista se adapte al plan de entrenamientos? Además esos planes se comparten con compañeros de entrenamiento por WhatsApp ¿por qué no hacerlo en una aplicación? Una aplicación que te permita crear tu plan de entrenamientos, compartirlo, una aplicación que se ajuste a tu horario, a tu tiempo libre.

Por lo tanto, el resultado del trabajo será una **aplicación móvil en Android** que te permitirá consultar los entrenamientos que haya que realizar diaria y semanalmente. Entrenamientos que puedan generarse automáticamente y que puedan ser adaptados a nuestras necesidades. Aunque inicialmente pretenda dar solución a una necesidad del equipo 42195.es podrá ser exportada a otros entornos.

1.2 Objetivos del Trabajo

El Objetivo principal del trabajo va a ser el desarrollo de la aplicación TrainerPlan para dispositivos móviles Android. La función contendrá unas **funciones principales** como van a ser la creación, modificación, eliminación y consulta de planes de entrenamiento, que las podríamos detallar en los siguientes:

- **Generador automático de plan de entrenamientos:** Genera un plan de entrenamientos a partir de variables básicas como tipo de deporte, fecha de competición y tiempo objetivo.
- **Modificar planes de entrenamiento:** Nos permite modificar y adaptar a nuestro horario un plan de entrenamientos.
- **Guardar plan de entrenamiento:** Nos permite guardar el plan de entrenamientos que hemos generado, creado o modificado.
- **Listar planes de entrenamiento:** Lista los planes de entrenamiento guardados.
- **Consulta semanal de plan de entrenamiento:** Obtiene un listado de los entrenamientos a realizar semanalmente y nos informa del tiempo total de dedicación.
- **Consulta diaria del plan de entrenamiento:** Obtiene un detalle del entrenamiento que tenemos que realizar un día concreto.
- **Eliminar planes de entrenamiento:** Nos permite eliminar los planes de entrenamiento guardados previamente.

Las funcionalidades que podría tener la aplicación podrían ser muchas más pero dada la limitación de recursos humanos y de tiempo que existe, he optado por garantizar el cumplimiento de los objetivos.

Además el proyecto tendrá los siguientes objetivos **no funcionales**, entre otros:

- Se aplicará el concepto de **usabilidad** a la aplicación: Con un tiempo de aprendizaje para el usuario sencillo, con un manual de usuario, con una ayuda en la propia aplicación y un diseño que se adapte a los diferentes

dispositivos Android del mercado: teléfonos móviles, tabletas y SmartTV con sistemas operativos Android.

- La aplicación será **segura**. La aplicación utilizará un servidor https donde se almacenarán y desde donde se servirán los planes de entrenamiento. Además se utilizarán patrones y recomendaciones de programación segura. Se realizarán copias de seguridad del servidor cada 24 horas.
- La aplicación cumplirá con la **legislación** actual: Ley de Protección de Datos, Ley de cookies...

Dentro del plan de Estudios del Máster la realización de esta aplicación me permitirá adquirir una experiencia de más de 600 horas en el desarrollo de aplicaciones Android, si añadimos al Trabajo de Fin de Máster las horas de los módulos de introducción a aplicaciones móviles en Android y desarrollo de aplicaciones avanzadas en Android.

1.3 Enfoque y método seguido

Actualmente partimos de la web Trainerplan.com que permite generar planes de entrenamiento para maratones a partir de una fecha de competición y un tiempo objetivo. Considero que la marca TrainerPlan es ideal para el proyecto que se quiere desarrollar: Una aplicación para móviles que te permita generar planes de entrenamiento para atletismo, triatlón y carreras de montaña.

La estrategia que voy a adoptar es el desarrollo de un **producto nuevo** ya que nuestro proyecto va a estar basado en **tecnología móvil** en lugar de tecnología web. Además, la aplicación actual sólo genera planes de entrenamiento para maratones. El nuevo producto no sólo generara planes de entrenamiento de atletismo sino que incorporará nuevos deportes y diferentes distancias.

La tecnología en la que voy a realizar la aplicación va a ser Android, aunque lo ideal hubiese sido realizar un proyecto que utilizase tecnología web y tecnología móvil para dispositivos iOS y Android. Las razones por las que me he decantado únicamente por esta tecnología son las siguientes:

- Actualmente más del 80% de los usuarios utiliza esta tecnología en sus móviles.
- Conozco mejor la tecnología, ya que he cursado el módulo de iniciación al desarrollo de aplicaciones Android y actualmente estoy cursando el de desarrollo avanzado.
- Para cumplir el objetivo, y teniendo el tiempo y los recursos limitados, hay que decantarse por una de las tecnologías.

1.4 Planificación del Trabajo

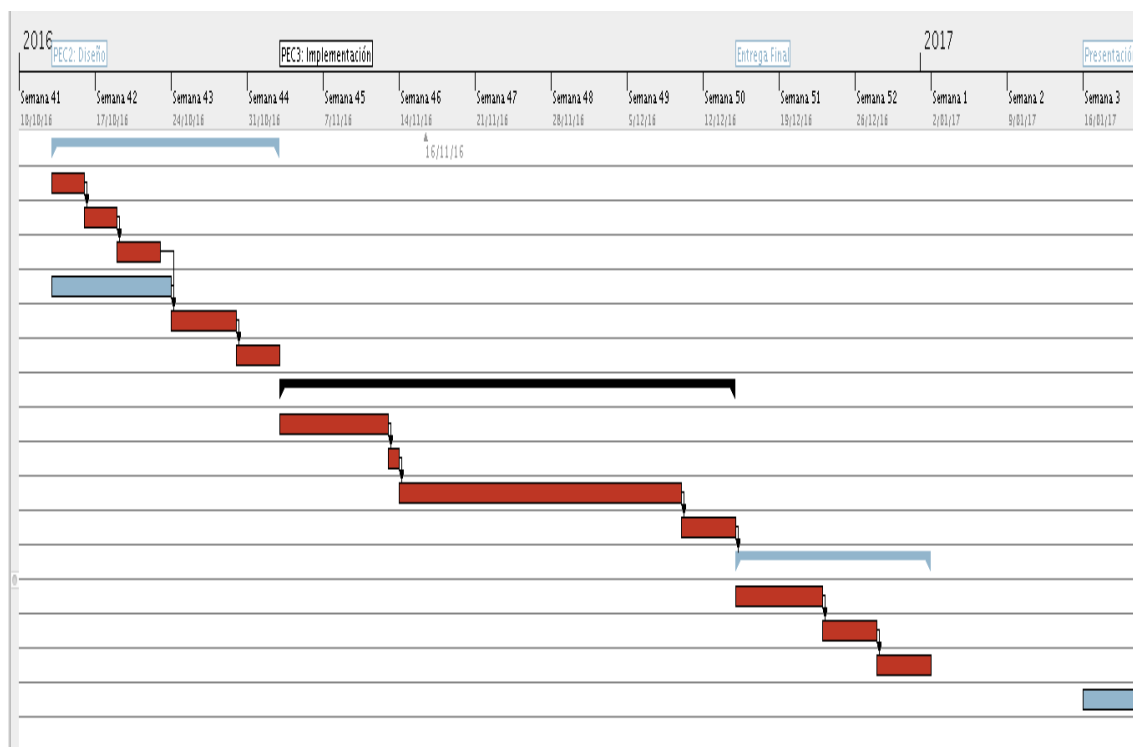
Para la realización de la elaboración de la planificación de trabajo he pensado en una dedicación de 2 horas diarias en los días laborables y 5 horas diarias en los días festivos. Para ello he tenido en cuenta mi horario laboral, incluyendo los puentes como días festivos.

Tabla de Hitos

Nombre	Horas (Lab. + Fest.)	Inicio	Final
PEC2: Diseño	60 (30 + 30)	13/10/2016	02/11/2016
Análisis: Usuario y contexto de uso	8 (4 + 4)	13/10/2016	15/10/2016
Diseño: Diseño conceptual	8 (4 + 4)	16/10/2016	18/10/2016
Diseño: Prototipado	8 (4 + 4)	20/10/2016	22/10/2016
Evaluación	6 (2 + 4)	13/10/2016	23/10/2016
Diseño de casos de uso	15 (10 + 5)	23/10/2016	29/10/2016
Diseño de la arquitectura	15 (2 + 13)	29/10/2016	02/11/2016
PEC 3: Implementación	100 (40 + 60)	03/11/2016	14/12/2016
Desarrollo entorno servidor	25 (14+11)	03/11/2016	12/11/2015
Pruebas servidor	5 (0+5)	12/11/2015	13/11/2015
Desarrollo de la aplicación cliente	60 (20+40)	13/11/2015	09/12/2016
Pruebas cliente	10 (6+4)	10/12/206	14/12/2016
Entrega Final	30 (2+28)	15/12/2016	01/01/2017
Video presentación	10 (2 + 8)	15/12/206	22/12/2016
Manual de usuario e instalación	10 (2 + 8)	22/12/206	27/12/2016
Completar la memoria del proyecto	10 (0 + 10)	27/12/206	01/01/2017
Presentación	10 (10+0)	16/01/2017	20/01/2017

La tabla de hitos está calculada a partir de las horas de dedicación en lugar de los días por lo que hay tareas que terminan el mismo día que comienzan otras.

Diagrama de Gannt:



Recursos Humanos

En el desarrollo del TFM Aitor Lopez de Aberasturi como autor del trabajo va a tener el soporte como consultor de Eduard Martin Lineros.

Recursos Materiales

Para el desarrollo del TFM vamos a disponer de los siguientes recursos materiales:

- Un ordenador MacBook Air
- Software: Android Studio, Balsamic, Gannt Proyect, Camtasia,...
- Servidor Web: Servidor LAMP.
- Cuenta Android Developer

1.5 Breve resumen de productos obtenidos

- **Aplicación TrainerPlan:** Será el producto final principal. Va a consistir en una aplicación realizada inicialmente para dispositivos Android que cumpla los objetivos del Trabajo Final de Master. Además de entregarse el apk correspondiente, la aplicación será subida a Google Play.
- **Código fuente de la aplicación:** Junto a la aplicación será entregado el código fuente de la aplicación comentado.
- **Manual de usuario:** Un manual que ayude a entender el uso de la app.
- **Manual de instalación:** Con los requisitos necesarios y guía de instalación.
- **Vídeo presentación del producto:** Realizado con la herramienta Camtasia. Será entregado a través de la herramienta @Presenta, explicando los objetivos del trabajo, los aspectos clave del desarrollo, las decisiones tomadas y las conclusiones y resultados obtenidos. Ésta será dirigida a un público generalista.
- **Memoria del proyecto:** Ésta misma.

1.6 Breve descripción de los otros capítulos de la memoria

En los próximos capítulos vamos a incorporar información relativa a los diferentes fases del proyecto: Diseño, implementación y entrega final.

Fase de Diseño

En proyectos basados en el desarrollo de una App constará de los documentos del Diseño Centrado en el Usuario (DCU) y del diseño técnico de la aplicación:

1. Análisis de los usuarios y contexto de uso: Definiremos las características de los usuarios, sus necesidades y objetivos, así como el contexto de uso.
2. Diseño conceptual: Se Elaborarán los escenarios de uso a partir de la información recopilada en la primera fase, para determinar necesidades de los usuarios y de diseño.
3. Prototipado: Se realizará un prototipo horizontal de la aplicación.
- 4,. Evaluación: Estas fases del DCU requerirán un seguimiento y evaluación.
5. Definición de los casos de uso: Estos casos de uso servirán para establecer las funcionalidades de la aplicación. Se definirán los actores y el flujo.
6. Diseño de la arquitectura: Se definirá la arquitectura del sistema, identificando las entidades que se representarán en la base de datos, las clases y objetos que se utilizarán para gestionar los diferentes procesos y la estructura de la API que servirá para realizar las peticiones enviados al servidor desde un cliente.

Fase de Implementación

- 1.- Pruebas servidor: En esta fase realizaremos el desarrollo de una API sobre un servidor LAMP y realizaremos un registro de pruebas sobre el servidor.
- 2.- Pruebas cliente: En esta fase realizaremos el desarrollo de un App que realizará peticiones sobre la API creada en la fase anterior, realizaremos un registro de pruebas realizadas sobre el cliente.

Entrega final

Además de preparar un manual de usuario e instalación que incorporaremos al anexo de esta memoria obtendremos las conclusiones del trabajo realizado: Lecciones aprendidas, cumplimiento de los objetivos , planificación correctas y líneas futuras de trabajo.

2. Fase de Diseño

En esta fase he realizado un análisis de los usuarios, para ello he realizado una encuesta y cuatro entrevistas. Con estos datos he obtenido el contexto de uso de la aplicación y los escenarios de uso, para determinar necesidades de los usuarios y de diseño. He identificado los actores, listado los casos de uso y he realizado un prototipo de la aplicación.

Finalmente he realizado un diseño de la base de datos y he definido el diseño de la arquitectura de nuestro sistema.

2.1. Análisis de los usuarios y contexto de uso

Para definir las características de los usuarios, sus necesidades y objetivos, así como el contexto de uso. Para ello he realizado una encuesta para tener una idea de las necesidades generales de los usuarios y además he realizado unas entrevistas a usuarios potenciales que me han ayudado a obtener los objetivos y contexto de uso de la aplicación.

2.1.1.- La encuesta

Como paso previo a la realización de una entrevista hemos realizado una encuesta para obtener más información de los usuarios potenciales de nuestra aplicación.

2.1.1.1.- La herramienta

Para realizar la encuesta hemos utilizado la herramienta que nos ofrece Google para hacer formularios. Este es el link Google Forms que he utilizado:

https://docs.google.com/a/kaixo.com/forms/d/1NVeKluFsSkgD9tvhtdUf544gefmiDak_T8EsK-WNhVw/viewform?c=0&w=1&usp=mail_form_link

2.1.1.2.- Las fuentes de información

Para dar a conocer la encuesta se han utilizado diferentes fuentes:

1.- **El foro del equipo 42195.es:** Es un foro que tiene bastante movimiento donde el perfil de usuario coincide al 100% con nuestro perfil de usuario potencial.

<http://foro.42195.es/viewtopic.php?f=19&t=1904>

2.- **Redes sociales 42195.es:** Se ha utilizado la red social del equipo en Facebook para obtener resultados.

3.- Mediante **WhatsApp** a mi grupo de entrenamiento y conocidos.

2.1.1.3.- Las preguntas

El objetivo de la entrevista ha sido conocer un poco más a los usuarios potenciales de nuestra aplicación. Por una parte, la tecnología que utilizan: móviles, tabletas y sus sistemas operativos. En segundo lugar, saber qué tipo de deportes practican. En tercer lugar si han utilizado algún plan de entrenamientos y si estarían dispuestos a utilizarlo en el móvil. He dejado las dos últimas preguntas abiertas para saber los objetivos y obtener información adicional, como el tiempo que dedican a entrenar o tiempo libre del que disponen.

Las preguntas que hemos realizado son las siguientes:

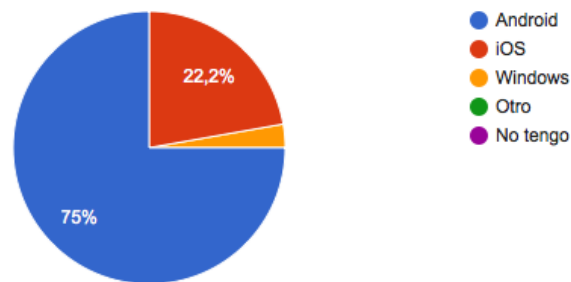
- ¿Qué sistema operativo tiene tu teléfono móvil?
- ¿Qué sistema operativo tiene tu tablet?
- ¿Qué deportes practicas?
- ¿Has utilizado alguna vez un plan de entrenamientos?
- ¿Te gustaría tener tu plan de entrenamientos en tu móvil?
- ¿En qué idioma?
- ¿En qué tipo de pruebas deportivas participas?
- ¿Cuál es tu próximo o próximos objetivos deportivos? (Ej: Medio Ironman Vitoria 5h.30min)
- Información adicional (Si quieres me puedes contar quién eres, tiempo que dispones para entrenar, cuándo y con quién entrenas,...)

2.1.1.4.- Los resultados

La encuesta la han respondido 72 personas. Comentamos algunas de las conclusiones que hemos obtenido:

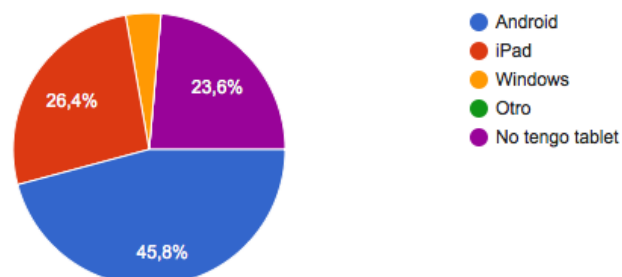
Todos los usuarios tienen smartphone, 75% tiene Android y un 22,2% iOS.

¿ Qué sistema operativo tiene tu teléfono móvil ? (72 erantzun)



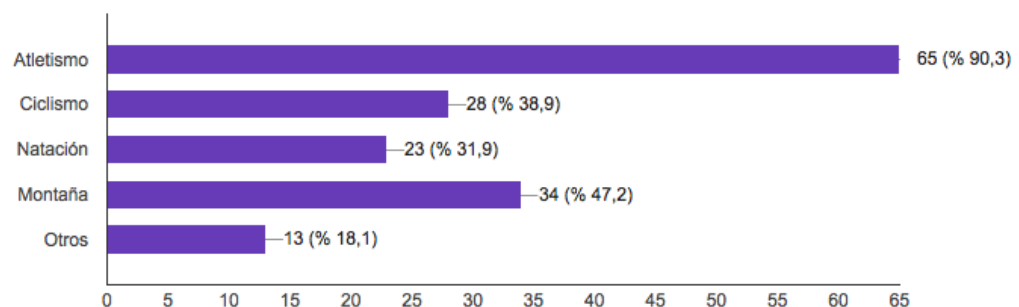
El 76% de los usuarios tiene tablet, el 45% con sistema Android y 26% iPad.

¿ Qué sistema operativo tiene tu tablet ? (72 erantzun)



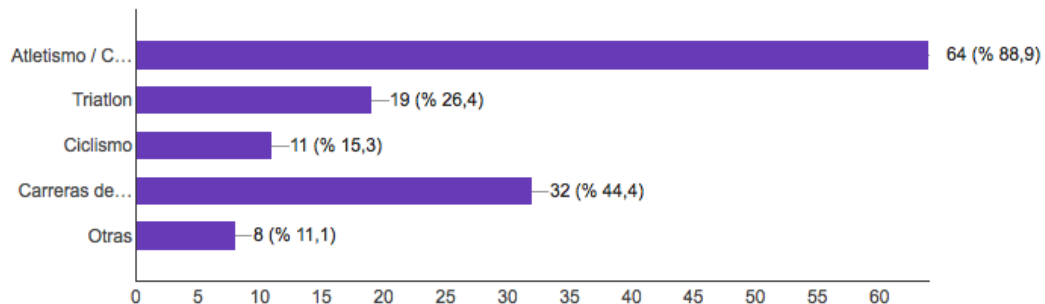
Destaca que el 90% de los usuarios practican el atletismo, frente a otros deportes, como natación o ciclismo (31,9% o 38,9% respectivamente).

¿ Qué deportes practicas ? (72 erantzun)



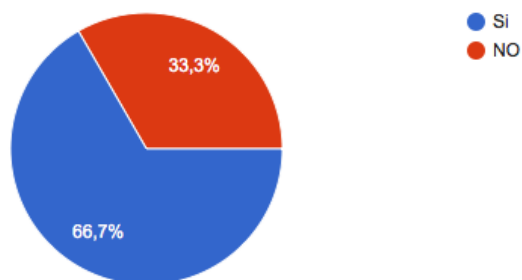
Prácticamente todos los que practican el atletismo participan en pruebas deportivas.

¿ En qué tipo de pruebas deportivas participas ? (72 erantzun)



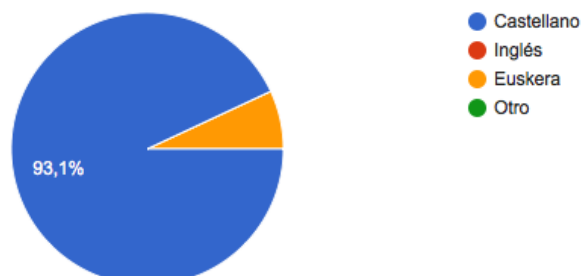
El 66% de los usuarios ha utilizado planes de entrenamientos.

¿ Has utilizado alguna vez un plan de entrenamientos ? (72 erantzun)



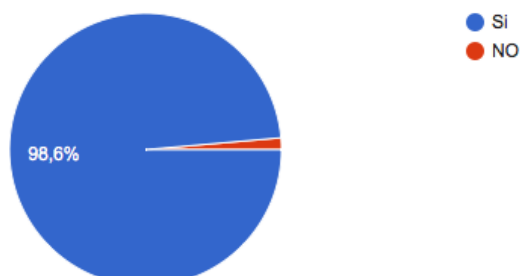
A la mayoría de los usuarios les gustaría que la aplicación estuviese en castellano.

¿ En qué idioma ? (72 erantzun)



A todos menos a un usuario le gustaría disponer un plan de entrenamientos en su móvil.

¿ Te gustaría tener tu plan de entrenamientos en tu móvil ? (72 erantzun)



2.1.2.- Entrevistas

Para obtener mayor información del uso se han realizado **entrevistas** a cuatro usuarios potenciales de la aplicación. Las preguntas se han centrado en obtener información de la forma en la que entrenan, con quién entrenan, cuándo entrenan, si utilizan planes de entrenamiento,... El resultado lo hemos plasmado en las fichas que detallamos a continuación, que nos van a ayudar para realizar un correcto contexto de uso.

Nombre	Victor Perdiz
Información Personal:	40 años, Director de Casa Rural
Información de Interés:	Víctor es el promotor del proyecto 42195.es que surgió como una idea personal con la finalidad de realizar la maratón más importante del mundo: la Maratón de Nueva York. Actualmente el equipo cuenta con más de 3000 socios y equipos federados de atletismo, triatlón y montaña. Fue uno de los 100 primeros atletas en realizar los 6 World Marathon Majors (Nueva York, Chicago, Boston, Berlín y Tokio). Su preferencia han sido siempre las carreras de medio fondo, aunque le gusta la aventura y últimamente apuesta por la libertad que dan las carreras de montaña.
Deportes:	Carreras de asfalto y de montaña.

Próximos objetivos:	Su próximo objetivo es participar en la Lenin Sky Race, la carrera más alta del mundo en el Elbrús, el monte más alto de Europa.
Plan de entrenamiento:	<p>Víctor intenta salir a correr casi todos los días. Normalmente cuando prepara alguna competición utiliza el plan de entrenamientos que le envía Víctor Clemente (su entrenador cuando era niño y de atletas ilustres como Martín Fiz y Maite Zúñiga) y lo publica en el foro del equipo. Normalmente suele salir a entrenar solo, cuando encuentra un hueco libre en su apretada agenda.</p> <p>Víctor carga semanalmente el plan de entrenamientos que le envía Víctor Clemente en el foro del equipo 42195.es.</p>

Nombre	Gorka Herce
Información Personal:	Profesor de Informática, 32 años, Oiartzun
Información de Interés:	Gorka, además de corredor y participante en carreras, colabora en la organización y colaboración de las mismas: Organiza la carrera de montaña Intxixu Trail en su pueblo. Además colabora con herrikrosa.eus , que se encarga del cronometraje y organización de carreras.
Deportes:	Carreras de asfalto y montaña
Próximos objetivos:	Maratón de Donostia en 2 horas y 45 minutos
Plan de entrenamiento:	Durante una temporada ha contratado los planes personalizados de entrenamiento a Juan María Jiménez (http://www.juanmariajimenez.com/), un contrastado entrenador y corredor. También utiliza los planes de entrenamiento que publica el equipo Green Power Sport Team (http://greenpowerst.com/foro/) en su web.

Nombre	Asier Fernández
Información Personal:	Empleado de banca, 39 años, Vitoria-Gasteiz

Información de Interés:	Asier es padre de familia numerosa. Su trabajo y atención a sus tres hijos hacen que sacar un minuto para practicar deporte sea prácticamente imposible. Ajusta el tiempo de sus entrenamientos a los fines de semana y a la hora en la que sus hijos están en el colegio.
Deportes:	Triatlón y carreras de asfalto
Próximos objetivos:	Triatlón de Vitoria en menos de 5 horas.
Plan de entrenamiento:	Sus días y horas de entrenamiento están perfectamente planificados: Martes noche: correr (entre 45 minutos y 1 hora). Miércoles: correr de 15:30h a 16:15h. Jueves: nadar de 15:30h-16:30h. Sábado: nadar de 8:15h-9:15h. Domingo: Vuelta en bicicleta. Los entrenamientos los realiza normalmente solo.

Nombre	Lourdes Carrasco
Información Personal:	Profesora, 51 años, Lasarte-Oria.
Información de Interés:	Lourdes es madre de familia numerosa y aprovecha tres días a la semana para hacer deporte con dos amigas.
Deportes:	Carreras populares
Próximos objetivos:	Media maratón Donibane Lohizune-Hondarribia.
Plan de entrenamiento:	Lourdes queda para entrenar los martes y jueves a las 6 de la tarde y los sábados a las 10 de la mañana. Normalmente suelen entrenar 3 horas semanales. Le gusta el ambiente de las carreras populares y ya ha participado varias veces en la Behobia-San Sebastián.

2.1.3.- Contexto de uso

He identificado dos contextos de uso diferenciados de la aplicación. Por una parte aquellos usuarios que no dispongan de plan de entrenamiento generarán

automáticamente el plan de entrenamientos y posteriormente realizarán el seguimiento del mismo. Por otra parte, aquellos usuarios que dispongan de plan de entrenamiento lo introducirán en la aplicación y posteriormente realizarán el seguimiento del mismo. Asier y Lourdes serán un claro ejemplo del primer caso y Víctor y Gorka del segundo.

Víctor recibirá semanalmente los planes de entrenamiento en una hoja excel de su entrenador Víctor Clemente y los introducirá en la aplicación. Diariamente realizará el seguimiento del mismo en su teléfono móvil.

Gorka introducirá los entrenamientos obtenidos la web Green Power Sport Team. Los domingos echará un vistazo al total de kilómetros y horas de dedicación semanal para preparar la maratón. Diariamente antes de entrenar consultará el tiempo y ritmo al que al que tendrá que ejercitarse. Finalmente, tras el entrenamiento marcará el grado de cumplimiento del mismo.

Asier obtendrá automáticamente un plan de entrenamientos de TrainerPlan y lo guardará en su móvil. Semanalmente ajustará los entrenamientos a su disponibilidad. Todos los días accederá a su plan de entrenamientos y marcará su grado de cumplimiento.

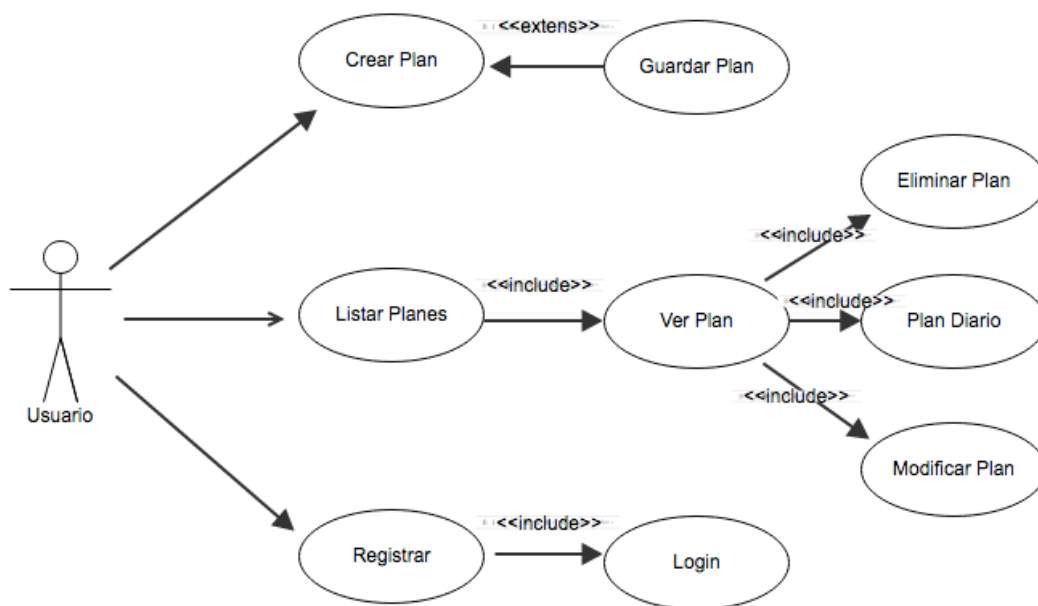
Para cumplir su objetivo, **Lourdes** generará automáticamente su plan de entrenamientos con TrainerPlan y lo adaptará a sus días de entrenamiento. Todos los días antes de entrenar consultará el entrenamiento que tiene que realizar.

2.2. Diseño conceptual

A partir de la información recopilada en la fase anterior he elaborado los escenarios de uso, para determinar necesidades de los usuarios y de diseño. Por una parte he identificado los actores y he listado los casos de uso donde se especifican los actores, precondiciones, flujo y postcondiciones.

2.2.1.- Actores y Flujo

El único actor que vamos a tener en nuestro sistema es el usuario deportista que utilizará la aplicación para la consulta de su plan de entrenamientos. En el siguiente diagrama UML nos permite identificar de forma gráfica el flujo:



2.2.1.- Casos de uso

A continuación se detallan las plantillas con los diferentes casos de uso:

Nombre: Crear Plan
Descripción: Permite al usuario crear un plan de entrenamientos.
Actores: Usuario.
Precondición: No aplicable.
Postcondición: El usuario ha creado un plan de entrenamientos.
Flujo principal: <ol style="list-style-type: none">1. El usuario solicita un nuevo plan de entrenamientos.2. El sistema muestra las variables para obtener el plan.3. El usuario selecciona las variables.4. El sistema valida los datos y obtiene el nuevo plan de entrenamientos.5. El caso finaliza.
Flujo alternativo: <ol style="list-style-type: none">1. En el punto 4 el sistema visualiza un error si los datos no son correctos.

Nombre: Ver Plan
Descripción: Permite al usuario ver un plan de entrenamientos.
Actores: Usuario.
Precondición: El usuario ha creado y guardado previamente un plan de entrenamientos.
Postcondición: El usuario ha consultado un plan de entrenamientos.
Flujo principal: <ol style="list-style-type: none">1. El sistema muestra una lista de los planes de entrenamiento.2. El usuario selecciona un plan.3. El sistema muestra los planes de entrenamiento a realizar semanalmente.4. El usuario selecciona la semana a consultar.5. El sistema le muestra un listado de los entrenamientos diarios.6. El usuario selecciona el día a consultar.7. El sistema muestra el plan de entrenamiento diario detallado.8. El caso finaliza.
Flujo alternativo: <ol style="list-style-type: none">1. En los puntos 4 y 6 el usuario puede finalizar el caso.

Nombre: Guardar Plan
Descripción: Permite al usuario guardar un plan de entrenamientos.

Actores: Usuario.
Precondición: El usuario ha creado previamente un plan de entrenamientos.
Postcondición: El plan de entrenamientos es guardado en la base de datos del sistema.
Flujo principal: <ol style="list-style-type: none"> 1. El usuario solicita guardar el plan de entrenamientos. 2. El sistema solicita la conformidad del usuario. 3. El plan de entrenamientos se guarda en el sistema. 4. El caso finaliza.
Flujo alternativo: <ol style="list-style-type: none"> 1. Si en el punto 2 el usuario rechaza guardar el plan, el caso finaliza.

Nombre: Modificar Plan
Descripción: Permite al usuario modificar un plan de entrenamientos.
Actores: Usuario.
Precondición: El usuario ha seleccionado previamente un plan de entrenamientos.
Postcondición: El plan de entrenamientos es modificado en la base de datos del sistema.
Flujo principal: <ol style="list-style-type: none"> 1. El usuario solicita modificar el plan de entrenamientos. 2. El sistema muestra las variables para que el usuario las modifique. 3. El usuario modifica las variables. 4. El sistema solicita la conformidad el usuario para modificar los datos. 5. El plan de entrenamientos modificado se guarda en el sistema. 6. El caso finaliza.
Flujo alternativo: <ol style="list-style-type: none"> 1. Si en el punto 2 el usuario cancela la modificación, el caso finaliza. 2. Si en el punto 4 el usuario rechaza guardar el plan, el caso finaliza.

Nombre: Eliminar plan
Descripción: Permite al usuario eliminar un plan de entrenamientos guardado en la base de datos.
Actores: Usuario.
Precondición: El usuario ha seleccionado previamente un plan de entrenamientos.
Postcondición: Se elimina el plan de entrenamientos seleccionado.

Flujo principal:

1. El usuario solicita eliminar el plan de entrenamientos.
2. El sistema solicita la conformidad del usuario para eliminar el plan.
3. El plan de entrenamientos se elimina del sistema.
4. El caso finaliza.

Flujo alternativo:

1. Si en el punto 2 el usuario rechaza eliminar el plan, el caso finaliza.

Nombre: Registro de usuario

Descripción: Permite al usuario registrarse en el sistema.

Actores: Usuario.

Precondición: Ninguna.

Postcondición: El usuario se registra en el sistema.

Flujo principal:

1. El usuario solicita el registro en el sistema.
2. El sistema muestra las variables para registrar al usuario.
3. El usuario introduce las variables.
4. El sistema valida los datos y registra al usuario en el sistema.
5. El caso finaliza.

Flujo alternativo:

2. En el punto 4 el sistema visualiza un error si los datos no son correctos.

Nombre: Login de usuario

Descripción: Permite al usuario ingresar en el sistema.

Actores: Usuario.

Precondición: El usuario se ha registrado previamente.

Postcondición: El usuario ingresa en el sistema.

Flujo principal:

1. El usuario solicita el acceso al sistema.
2. El sistema muestra las variables para acceder al sistema.
3. El usuario introduce las variables.
4. El sistema valida los datos y permite acceder al usuario en el sistema.
5. El caso finaliza.

Flujo alternativo:

1. En el punto 4 el sistema visualiza un error si los datos no son correctos.

Nombre: Compartir plan de entrenamiento
Descripción: Permite al usuario compartir un plan de entrenamiento.
Actores: Usuario.
Precondición: El usuario se ha registrado previamente.
Postcondición: El usuario comparte un plan de entrenamiento.
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1. El usuario solicita compartir el plan de entrenamientos. 2. El sistema solicita la conformidad del usuario. 3. El sistema comprueba que el usuario se ha registrado y comparte el plan de entrenamiento. 4. El caso finaliza.
<p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. En el punto 2 el usuario muestra su disconformidad el caso finaliza.

Nombre: Búsqueda de planes de entrenamiento
Descripción: Permite al usuario buscar un plan de entrenamiento
Actores: Usuario.
Precondición: El usuario se ha registrado previamente.
Postcondición: El usuario busca un plan de entrenamiento.
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1. El usuario solicita buscar un plan de entrenamientos. 2. El sistema muestra deferentes opciones de búsqueda. 3. El usuario selecciona los criterios de búsqueda. 4. El sistema visualiza los diferentes planes obtenidos con el criterio especificado por el usuario. 5. El caso finaliza.
Flujo alternativo:

Nombre: Importar planes de entrenamiento
Descripción: Permite al usuario importar un plan de entrenamiento
Actores: Usuario.
Precondición: El usuario se ha registrado previamente.
Postcondición: El usuario importa.
<p>Flujo principal:</p> <ol style="list-style-type: none"> 1. El usuario solicita importar los planes de entrenamiento. 2. El sistema solicita la conformidad del usuario.

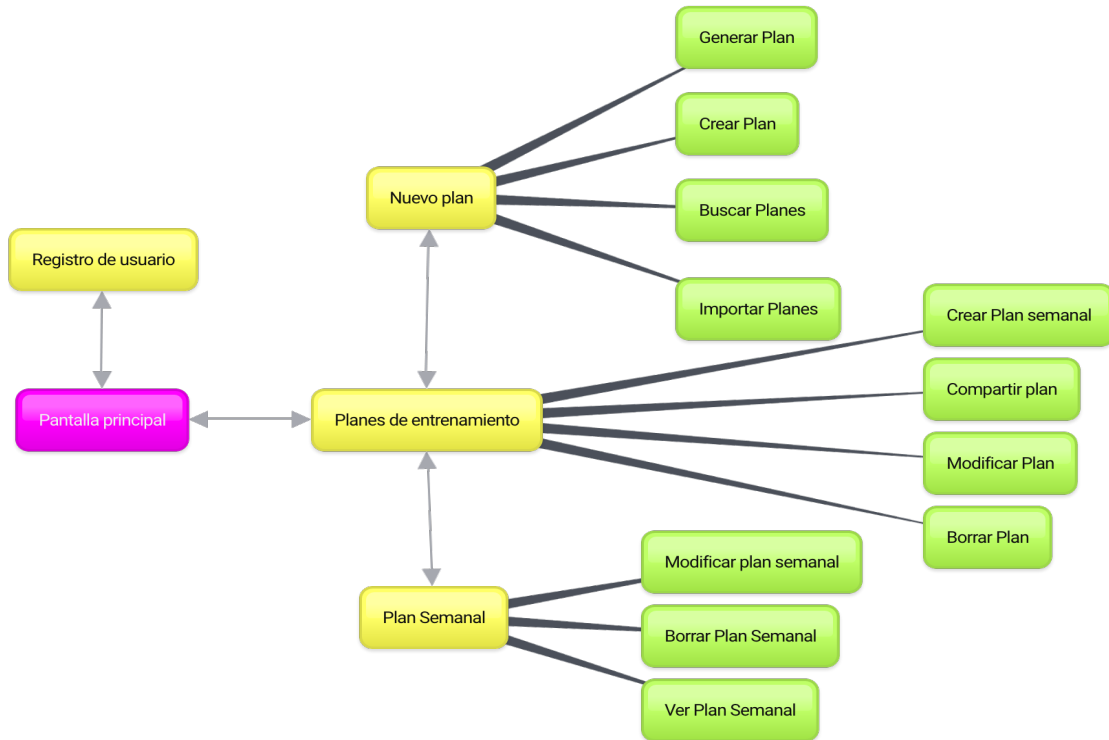
3. El acepta la importación y carga los planes de entrenamiento en el sistema.
4. El caso finaliza.

Flujo alternativo:

1. En el punto 2 el usuario muestra su disconformidad el caso finaliza.

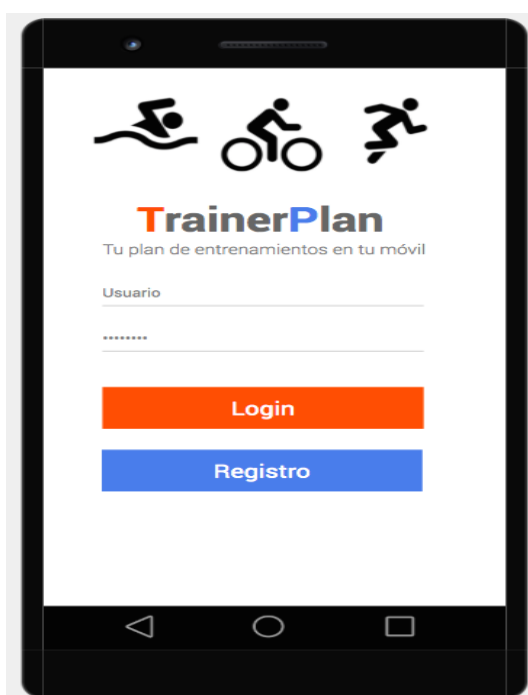
2.3. Prototipado

Antes de trabajar sobre el prototipado vamos a dibujar un mapa mental que nos va a ayudar a conocer la navegación principal de nuestro prototipo:

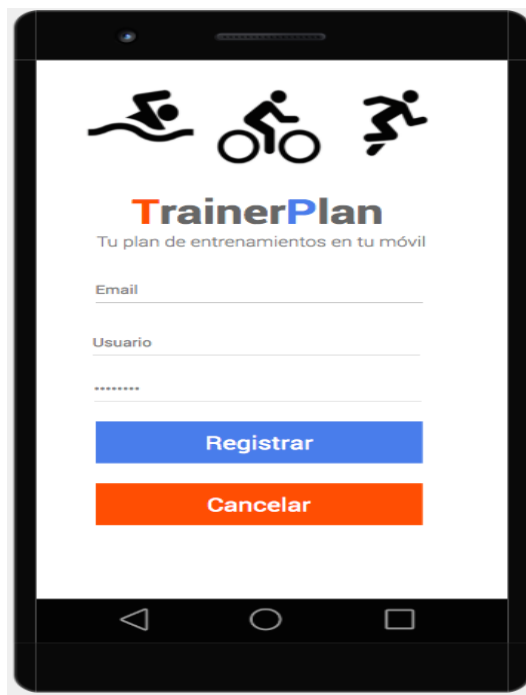


A continuación mostramos los prototipos de las diferentes pantallas que va a tener la aplicación:

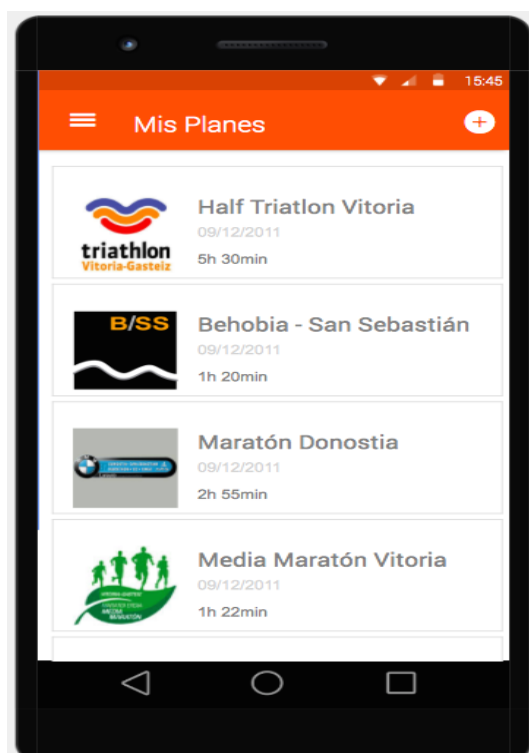
Pantalla de login:



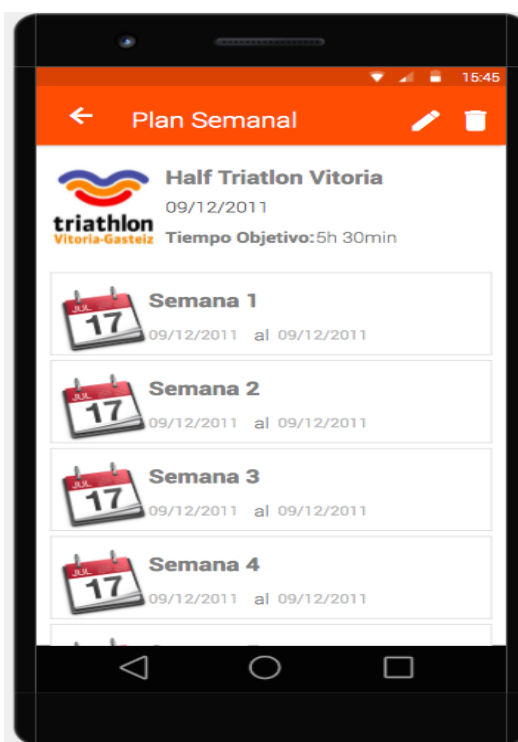
Pantalla de registro:



Planes de entrenamiento:



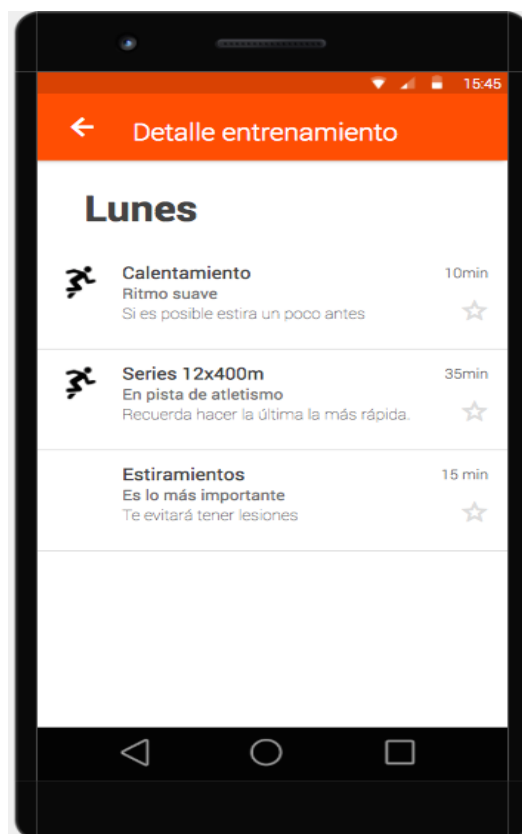
Plan semanal:



Plan diario:



Detalle plan diario:



Nuevo plan:

Nuevo Plan

Deporte
Atletismo

Disciplina
Maratón

Crear

Nuevo plan 2:

Nuevo Plan

Nombre: Maratón Barcelona

Logo: Browse...

Fecha: 03/05/2017

Tiempo Objetivo:

Crear

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Modificar plan:

Modificar Plan

Nombre: Half Triatlón Vitoria

Logo: ./images/378a0... Browse...

Fecha: 09/12/2011

Tiempo Objetivo: 5h 30min

Modificar Cancelar

Borrar plan:

Borrar Plan

¿ Desea reamente borrar el Plan ?

Borrar Cancelar

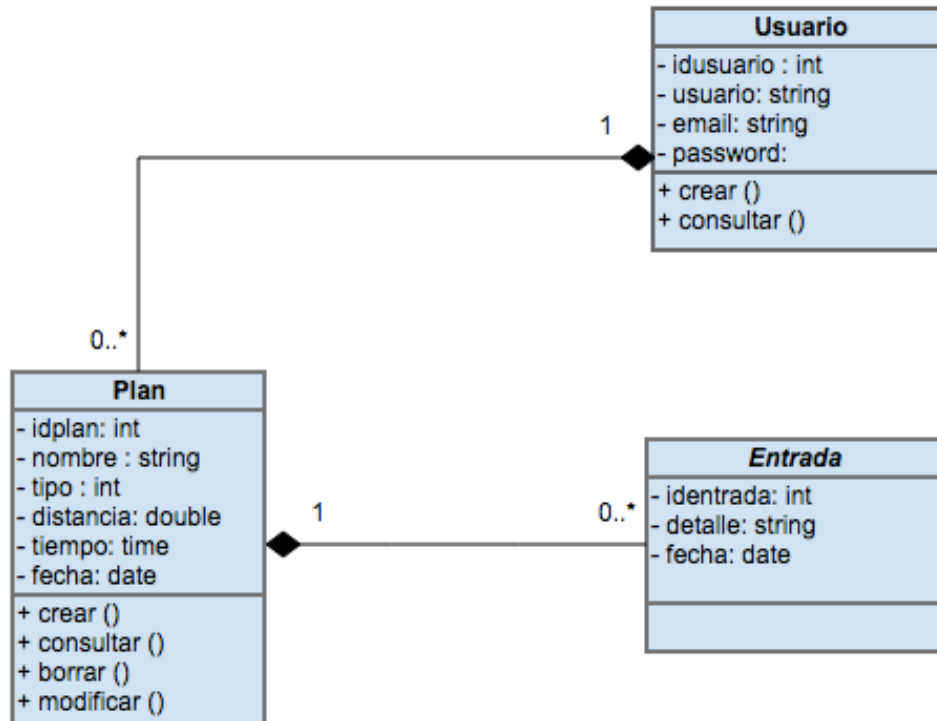
Los prototipos nos permiten hacer un acercamiento inicial al que va a ser el diseño de nuestra aplicación. Los anteriores prototipos nos dan una idea de cómo se van a implementar las diferentes funcionalidades de la aplicación (crear plan, visualizar planes, modificar, borrar, registro, identificación,...).

2.4. Evaluación

Durante el seguimiento de esta fase se han mencionado nuevas funcionalidades como pueden ser compartir un plan de entrenamientos o guardar el grado de cumplimiento de los mismos. Aunque pueden ser interesantes para los usuarios, he decidido no implementarlas inicialmente ya que pueden desviar la planificación inicial.

2.5. Diseño de la base de datos

El siguiente diagrama UML muestra el diseño de la base de datos:



Básicamente vamos a tener tres tablas:

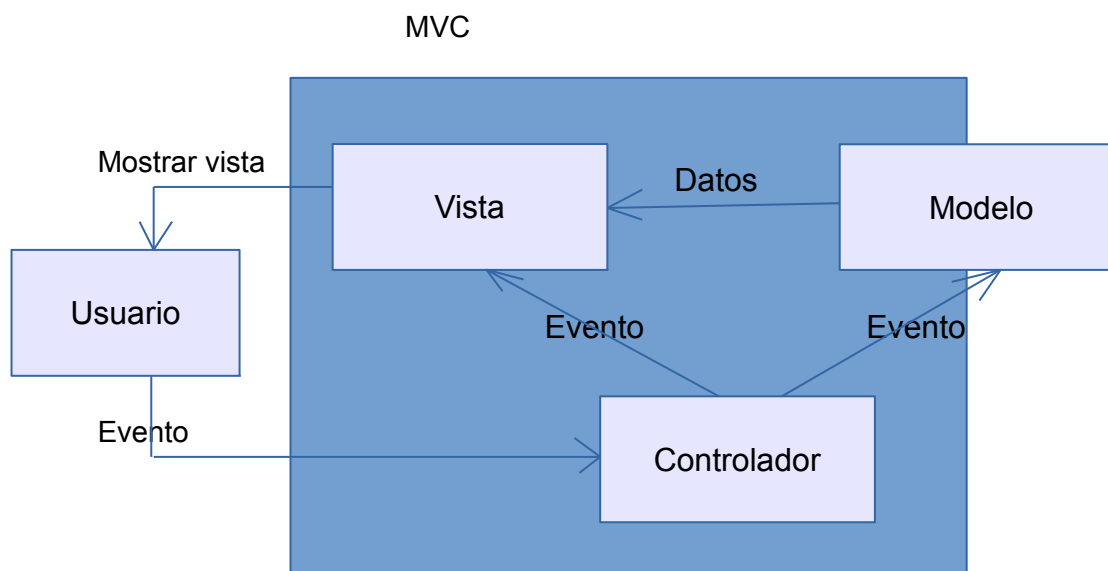
- **Usuario:** va a permitir guardar la información de los usuarios de la aplicación. Cada usuario puede guardar en el sistema varios planes de entrenamiento.
- **Plan de entrenamientos:** va a contener información del nombre del plan y sobre el objetivo (fecha, tiempo y distancia), además del tipo de plan del que se trate.
- Por cada plan vamos a tener varias **entradas** donde guardaremos información diaria de la actividad a realizar.

2.6. Diseño de la arquitectura

Para el diseño de la arquitectura de nuestro sistema vamos a utilizar los paradigmas MVC y SOA.

2.6.1.- Modelo Vista Controlador

El patrón que vamos a utilizar para el diseño de la arquitectura de nuestra aplicación va a ser **MVC** (modelo vista controlador)



Este patrón de arquitectura de software nos va a permitir separar los datos de la lógica de negocio de nuestra aplicación del interfaz de usuario.

Los componentes de un modelo MVC serían los siguientes:

- El **Modelo**: Representa la lógica de negocio. Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la Vista aquella parte de la información que en cada momento se le solicita para que sea

mostrada al usuario. Las peticiones de acceso o manipulación de información llegan al modelo a través del controlador.

- El **Controlador**: El Controlador hace de intermediario entre la Vista y el Modelo. Responde a **eventos** (normalmente acciones del usuario) e invoca peticiones al Modelo cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos). También puede enviar comandos a su Vista asociada si se solicita un cambio en la forma en que se presenta el Modelo (por ejemplo, desplazamiento o scroll por un documento o por los diferentes registros de una base de datos).
- La **Vista**: En nuestro caso van a ser los Layouts y diferentes elementos gráficos que va a visualizar el usuario. Da un formato correcto a la información proporcionada por el modelo para que pueda ser visualizada por el usuario.

2.6.2.- Arquitectura Orientada al Servicio

Además nuestro sistema va a disponer de una Arquitectura Orientada a Servicios (SOA: Service Oriented Architecture) que es un paradigma para diseñar y desarrollar sistemas distribuidos.

Esta arquitectura es necesaria utilizarla en nuestra aplicación para poder intercambiar información entre las aplicaciones. La información tiene que estar almacenada en un servidor.

Para el desarrollo de esta arquitectura vamos a necesitar unos servicios web. En este punto me he planteado la utilización de plataformas proveedoras de servicios como Backendless, Parse o Firebase vistas en diferentes asignaturas o realizar unos desarrollos propios.



La utilización de este tipo de herramientas de terceros te permite olvidarte del desarrollo de gran parte del código en la parte del servidor además de proporcionarte el espacio en la nube, sin embargo durante el master su uso no ha sido muy satisfactorio: Parse ya no admite nuevos usuarios y con Backendless tuvimos muchos problemas de conectividad. El utilizar herramientas de terceros te produce una dependencia hacia ellos y no te garantiza ni su uso

Por ello he decidido realizar mis propios servicios y utilizar un servidor en un hosting. Esto va a hacer que tenga que desarrollar toda la parte del servidor pero conseguiré independencia a la hora de tomar decisiones como cambio de servidor.

En nuestro caso vamos a desarrollar diferentes servicios en un servidor que nos permitan enviar y recibir información desde el cliente Android mediante mensajes de tipo JSON.

Se ha optado por una tecnología de servidor LAMP frente a otras como puede ser PHP o .NET por varios motivos: Conocimiento del entorno de desarrollo, disposición de servidores en la nube, etc.

2.6.3.- Aplicación Nativa, Web vs Híbrida

Otro de las decisiones que he tenido que tomar ha sido entre la utilización de una arquitectura web, nativa o una arquitectura híbrida.

En la siguiente tabla se puede observar las características a nivel general de cada una de ellas:

	WEB	HIBRIDA	NATIVA
Lenguaje	HTML, CSS, JavaScript	HTML, CSS, JavaScript	Java, c#, .NET
Coste desarrollo	BAJO	MEDIO	ALTO
Interfaz de Usuario	REGULAR	BUENA	BUENA
Rendimiento	MALO	REGULAR	BUENO
Multiplataforma	SI	SI	NO
Tiempo de desarrollo	BAJO	MEDIO	ALTO
AppStores	NO	SI	SI

Las arquitectura web y la arquitectura híbrida ya las conocía antes de realizar el master e incluso había realizado aplicaciones utilizando estas tecnología.

Probablemente para este tipo de aplicación lo lógico hubiera sido decantarme por una aplicación híbrida, ya que con menos esfuerzo podría tener la aplicación en las varias plataformas. Pero me he decantado por el desarrollo Nativo en Android por las siguientes motivos:

1. Realización de mi primer proyecto real utilizando tecnología Nativa.
2. He cursado las asignaturas Android básico y Android Avanzado y con el desarrollo de la aplicación en Trabajo de Fin de Master he adquirido una experiencia de más de 600 horas en el desarrollo de aplicaciones Android.
3. El rendimiento así como la interfaz de usuario es buena.
4. El 75 % de los usuarios que han participado en la encuesta tiene dispositivo Android.
5. Actualmente no dispongo de cuenta de desarrollador iOS, con lo que no se publicaría la aplicación en el AppStore.

3. Implementación

3.1. Entorno de desarrollo

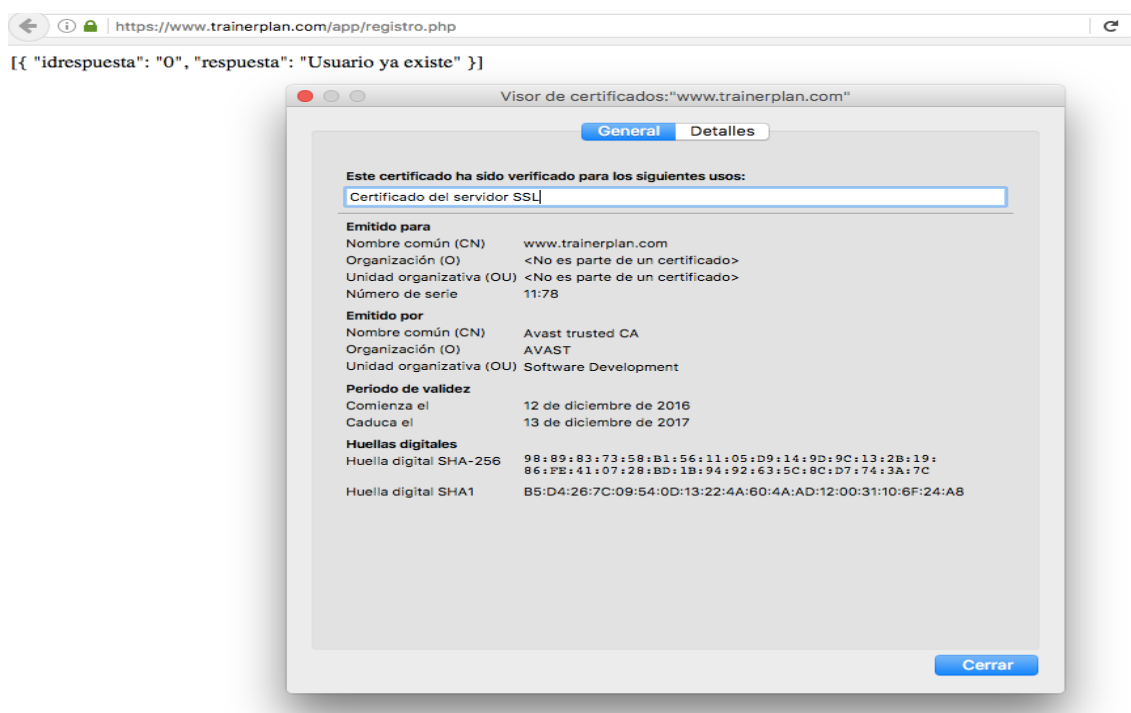
3.1.1.- Servidor

Para la programación del servidor he necesitado un entorno LAMP (Linux, Apache, Mysql y PHP). En la fase de desarrollo he utilizado la herramienta XAMP sobre un MacBook Air. En la fase de producción he utilizado un hosting de la empresa 1and1. Para la edición y programación de los ficheros se ha utilizado el editor vi.

He utilizado el servidor 1and1 Dual Limitado contratado en la empresa 1and1 en el que tengo la web antigua de TrainerPlan, con el dominio trainerplan.com. En ese servidor he creado la carpeta app donde he guardado las diferentes servicios que utiliza la aplicación. Además he contratado un servidor seguro para el dominio.

Servidor Seguro

He configurado un certificado de Seguridad SSL en la parte del servidor. Esto va a permitir que todas las comunicaciones entre el servidor y la aplicación y viceversa sean encriptadas:



Además a la hora de programar he utilizado otras técnicas, como el envío de la información mediante un POST en lugar de un GET, utilización de una variable denominada sesión para una doble validación o guardar el password encriptada con md5.

Base de datos

En el servidor he creado una base de datos **MySQL** donde se van a almacenar los usuarios, los planes compartidos por los usuarios y la tabla planes donde se guardan los datos para la generación de los planes de entrenamiento. A esta base de datos se puede acceder mediante phpmyadmin:

The screenshot shows the phpMyAdmin interface for a MySQL database named 'db447493637'. The table 'planescompartidosapp' is selected. The table structure is as follows:

Columna	Tipo	Función	Nulo	Valor
idplan	int(11)			17
idplanlocal	int(11)			43
nombre	varchar(255)			Media Maraton de Donosti
fecha	varchar(10)			2017/04/16
tiempo	varchar(5)			1:25
tipo	int(11)			1
lang	varchar(10)			en
idusuario	int(11)			8
json	longtext			{ "idplan": "1", "nombre": "Media Maraton de Donosti", "fecha": "2017/04/16", "tiempo": "1:25", "tipo": "1", "entrenamientos": { "semana": "0", "entrenamiento": "25 minutos 8x1.000 metros 15 minutos (recupera 1 minuto entre cada repetici\u00f3n. Ritmo de las repeticiones: entre 4:10 y 3:45). Descanso 60 minutos pesas 2 rectas. Descanso 20 minutos 2x4.000 metros 15 minutos (recupera 1 minuto entre cada repetici\u00f3n. Ritmo de las repeticiones: 16:40 y 16:00). Descanso 85 minutos, ejercicios de tobillo y 5 rectas." }, { "semana": "1", "entrenamiento": "25 minutos 2x7x400 metros 15 minutos (recupera de 45 segundos a 1 minuto entre cada repetici\u00f3n. Recupera 2 minutos entre cada tanda de 7 repeticiones. Ritmo de las repeticiones: entre 1:35 y 1:22). Descanso 60 minutos pesas 2 rectas. Descanso 20 minutos 7 kil\u00f3metros a ritmo controlado 10 minutos (Ritmo controlado: 29:00 en los 7 kil\u00f3metros). Descanso 85 minutos tobillos 5 rectas. " }, { "semana": "2", "entrenamiento": "25 minutos 5.000 metros 3.000 metros 15 minutos (recupera 2 minutos entre los dos tramos. Ritmo del 5.000 - 20:40. Ritmo del 3.000 - 12:00). Descanso 60 minutos pesas 2 rectas. Descanso 20
fechacreacion	datetime			2016-12-14 07:29:15

3.1.2.- Cliente

Para el desarrollo del cliente he utilizado la herramienta Android Studio versión 2.2.3. La aplicación se ha definido con un API level mínimo de 15 y con uno

máximo de 24. Con esta configuración se va a llegar al 97,4% de los dispositivos móviles con Android:

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION	Ice Cream Sandwich	
2.3 Gingerbread	10		Contacts Provider Social APIs User profile Invite intent Large photos Calendar Provider Calendar APIs Event intents Voicemail Provider Add voicemails to the device Multimedia Media effects for images and videos Remote control client Improved media player Camera Face detection Focus and metering areas Continuous auto focus Camera broadcast intents Connectivity Android Beam for NDEF push with NFC Wi-Fi P2P connections Bluetooth health profile Network usage and controls	
4.0 Ice Cream Sandwich	15	97,4%		
4.1 Jelly Bean	16	95,2%		
4.2 Jelly Bean	17	87,4%		
4.3 Jelly Bean	18	76,9%		
4.4 KitKat	19	73,9%		
5.0 Lollipop	21	40,5%		
5.1 Lollipop	22	24,1%		
6.0 Marshmallow	23	4,7%		
			Accessibility Explore-by-touch mode Accessibility for views Accessibility services Improved text-to-speech engine support User Interface Spell checker services Improved action bar Grid layout Texture view Switch widget Improved popup menus System themes Controls for system UI visibility Hover event support Hardware acceleration for all windows Enterprise VPN services Device policies Certificate management Device Sensors Improved sensors Temperature sensor Humidity sensor	

Base de datos

Dentro de la aplicación, los datos están almacenados en una base de datos **SQLite**. He utilizado la API o librería SugarORM para el mapeo de la base de datos utilizando el modelo de datos que se explicará más adelante. En ella he guardado los datos de los planes de entrenamientos generados por el usuario en la aplicación así como los entrenamientos de los que consta cada uno de los planes. Utilizo la librería Stetho para poder acceder mediante el navegador Chrome (*chrome://inspect/*) a su contenido:

Elements Network Sources Timeline Profiles Resources Audits Console						
Frames						
Web SQL						
TrainerPlan.db						
ENTRENAMIENTOS						
PLANES_ENTRENAMIENTO						
android_metadata						
sqlite_sequence						
IndexedDB						
Local Storage						
MisPreferencias						
Session Storage						
Cookies						
Application Cache						
ID	ID	FECHA	IDPLAN	NOMBREPLAN	TIEMPOOBJETIVO	
33	33	2017/07/09	1	Triatlon de Vitoria-Gasteiz distancia Half Ironman	5:15	
34	34	2017/07/30	1	Travesia a Nado Getaria-Zarauz	1:10	
35	35	2017/05/07	1	Maraton Martin Fiz	3:00	
37	37	2016/12/13	1	123	12:12	
43	43	2017/04/16	1	Media Maraton de Donosti	1:25	

3.2. Desarrollo de la aplicación

A continuación se va a explicar de forma general como se ha desarrollado la aplicación. Al tratarse de una arquitectura orientada al servidor y un MVC (modelo vista controlador), se explicará donde están definidas cada uno de los elementos que intervienen: Programas del servidor, modelo de datos, vistas, controladores, etc.

3.2.1.- Modelo de datos

El modelo de datos de la aplicación va a estar definido por la clase PlanesEntrenamientos y Entrenamientos. Donde se almacenará dentro de la aplicación Android los datos de los planes de entrenamiento y sus entrenamientos asociados:

```
public class PlanesEntrenamiento extends SugarRecord {  
    int idplan;  
    String nombreplan;  
    String fecha;  
    int tipo;  
    String tiempoobjetivo;  
    String idioma;  
}
```

```
public class Entrenamientos extends SugarRecord {  
    int semana;  
    String entrenamiento;  
    PlanesEntrenamiento plan;  
}
```

A continuación voy a explicar los principales controladores con sus vistas asociadas y la forma en la que se conectan con el servidor:

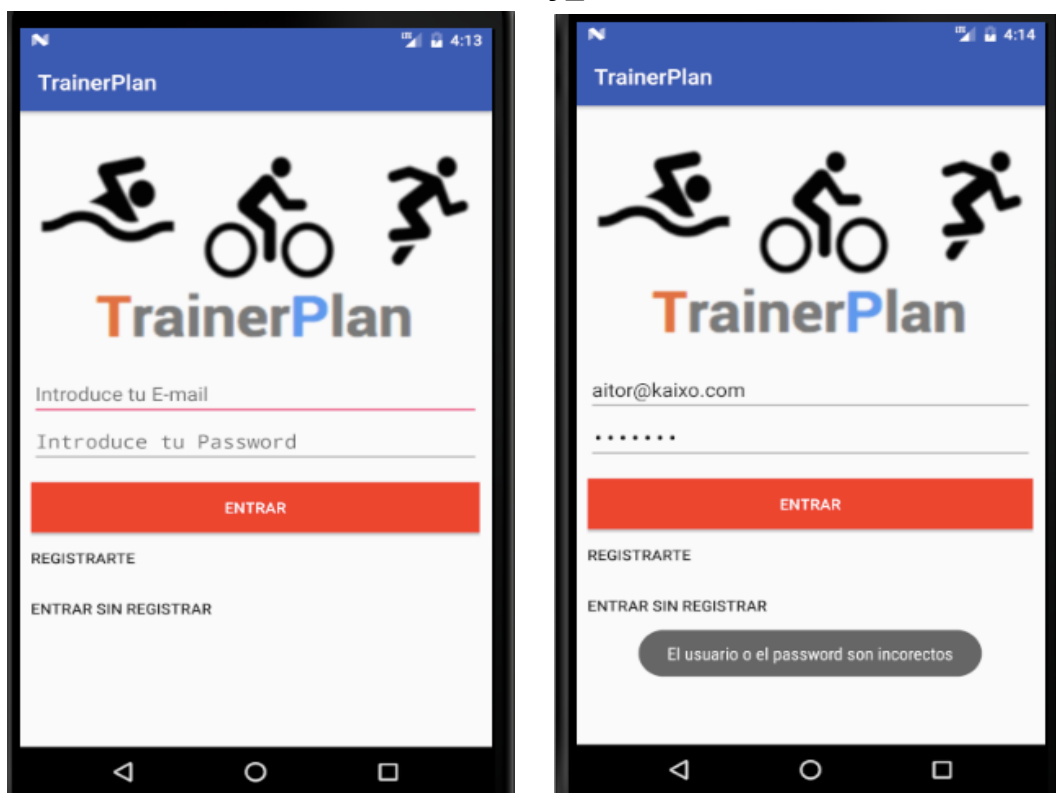
3.2.2.- Validación de usuario

La validación de usuario nos permite realizar actividades para los usuarios registrados como compartir nuestros planes de entrenamiento o importarlos desde el servidor, aunque podemos trabajar directamente sin registrarnos ni validarnos.

Para el desarrollo tenemos una parte del cliente que envía un mensaje **POST** al servidor con el email del usuario, password. Este mensaje se envía de forma asíncrona a la API **validacion.php** del que obtendremos una json con un valor idrespuesta. La respuesta puede ser 1 usuario correcto, 0 validación incorrecta o 2 que indica que el usuario se ha registrado pero no la ha activado. Si el usuario es correcto nos devuelve información adicional como el idusuario, email y sesión, que se almacenarán de forma persistente en la aplicación.

El controlador está desarrollado en la **clase Validar** entre otras cosas antes de validar el usuario realiza una validación en local para que los datos enviados al servidor estén completados y sean correctos.

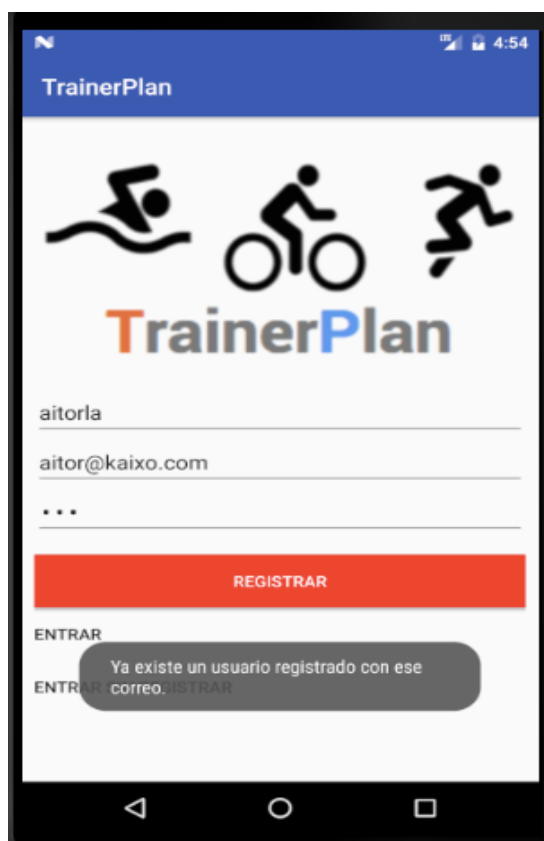
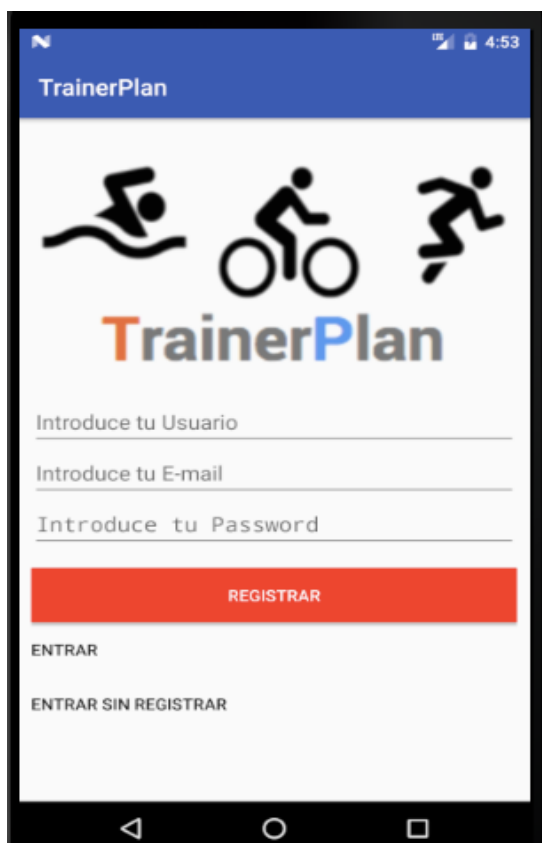
La vista está definida en el fichero **activity_validar.xml**.



Registro de usuario

El registro de usuario nos permite registrar el usuario en el servidor. Este registro va a permitir validar el usuario y poder compartir nuestros planes de entrenamiento o importarlos desde el servidor, no es necesario para el uso de la aplicación.

Para el desarrollo tenemos una parte del cliente que envía un mensaje **POST** al servidor con el nombre del usuario, email del usuario, password y lenguaje de configuración del terminal. Este mensaje se envía de forma asíncrona a la API **registro.php** del que obtendremos una json con un valor idrespuesta junto a un texto de la respuesta del servidor. La respuesta puede ser 1 alta correcta, 0 usuario ya existe o 2 que indica que no se ha podido dar de alta el nuevo usuario. En la base de datos guardaremos el password encriptada con md5, la sesión y la fecha de alta. Si el usuario se ha insertado correctamente, el programa enviará un correo al usuario para que active el usuario (utilizado para que los robots no creen usuarios). Esta validación se realizará con la API **activar.php**.



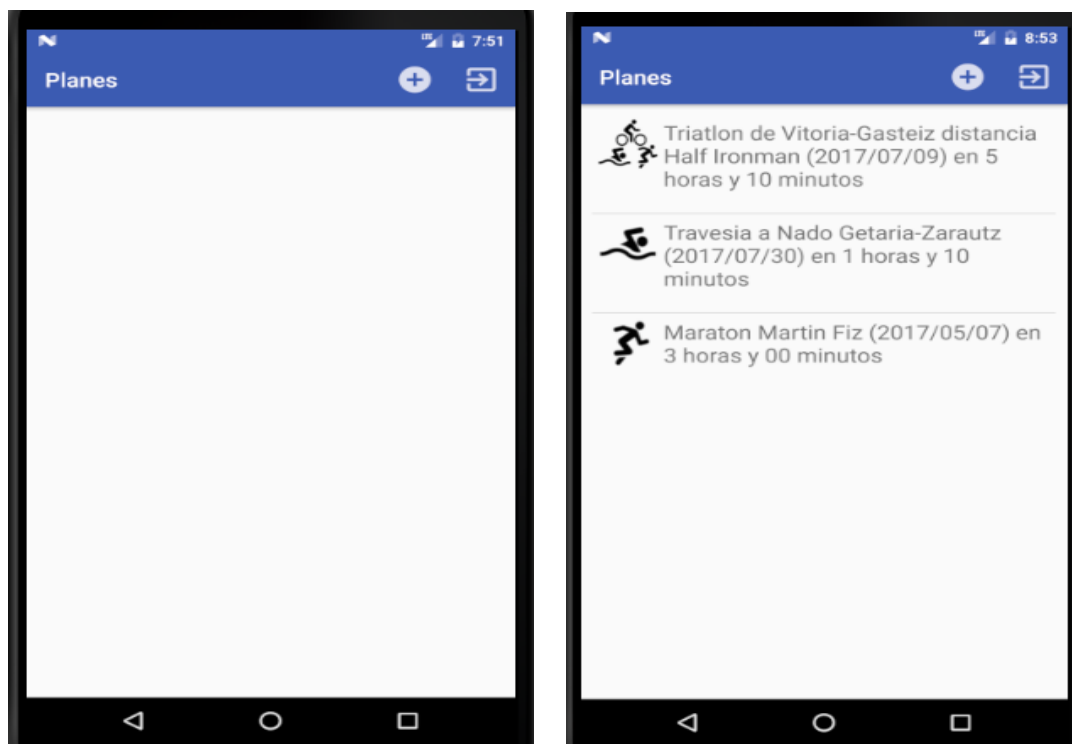
La parte del cliente está desarrollada en la **clase Registro** entre otras cosas antes de validar el usuario realiza una validación local para que los datos enviados al servidor estén completados y sean correctos. La vista está definida en el fichero **activity_registro.xml**.

Listado de Planes

Una vez que entremos en la aplicación obtendremos el listado de los planes que tenemos guardados en la base de datos local. Inicialmente este listado estará vacío. Podremos añadir nuevos planes manualmente, generados automáticamente o importados de un plan de entrenamientos compartido por otros usuarios. Para ello tendremos que pinchar sobre el icono “+” de la barra de herramientas.

Esta vista está programada con la **clase Planes**. Para obtener el listado he utilizado un adaptador **ListViewAdaterPlanes** en el que he cargado los planes de entrenamientos de la base de datos local y se visualizan en un ListView. Dependiendo del tipo de plan del que se trate (maratón, triatlón, natación,...) se mostrará una imagen u otra.

La vista está definida en el fichero **activity_planes.xml** y cada elemento de la lista en el fichero **list_row_plan.xml**.

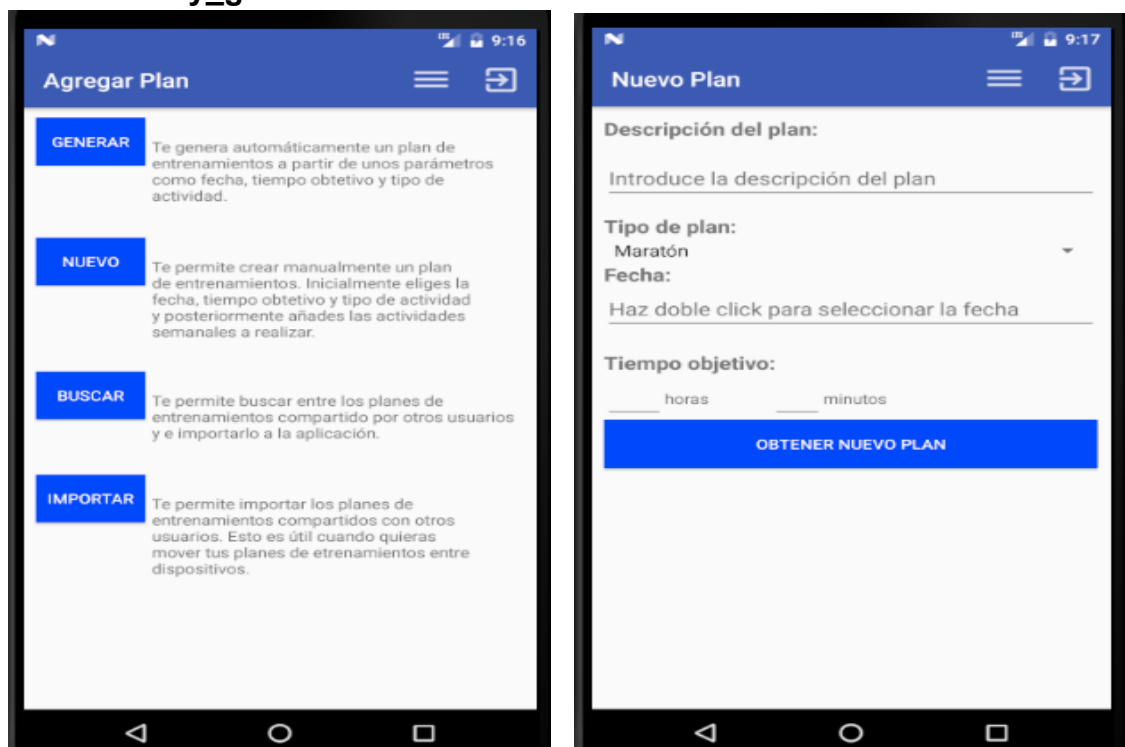


Agregar Nuevos Planes

Nos permite seleccionar las diferentes formas en las que podemos añadir un plan de entrenamiento:

- **Generar:** Te genera automáticamente un plan de entrenamiento a partir de unos parámetros como fecha, tiempo objetivo y tipo de actividad.
- **Nuevo:** Te permite crear manualmente un plan de entrenamiento. Inicialmente eliges la fecha, tiempo objetivo y tipo de actividad y posteriormente añades las actividades semanales a realizar.
- **Buscar:** Te permite buscar entre los planes de entrenamiento compartido por otros usuarios y e importarlo a la aplicación.
- **Importar:** Te permite importar los planes de entrenamientos compartidos con otros usuarios. Esto es útil cuando quieras mover tus planes de entrenamiento entre dispositivos.

El controlador está implementado en la clase **Generar.java** y la vista en el archivo **activity_generar.xml**.



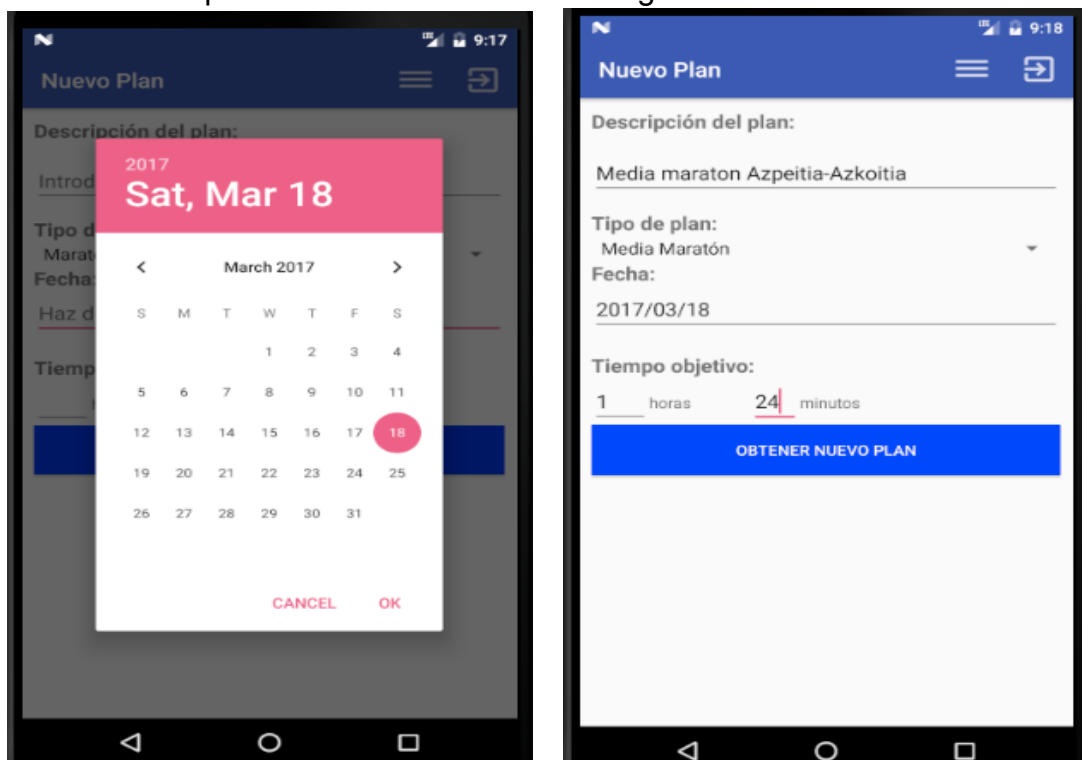
Los dos primeras opciones van a compartir un mismo controlador **Nuevo.java** y la vista **activity_nuevo.xml**

Nuevo Plan

Para generar o crear un nuevo plan rellenamos la vista **activity_nuevo.xml**. Se introduce el nombre del plan, el tipo de plan mediante un Spinner y para obtener la fecha utilizamos un Fragmento con un Picker.

El controlador comprueba si los datos se han rellenado y dependiendo del tipo de plan realizará las siguientes acciones:

- En caso de generar automáticamente el plan, los datos se envían mediante un POST al servidor los parámetros nombre, fecha, tipo y tiempo objetivo (hora y minutos) al programa **planes.php**. El servidor nos devuelve un plan de entrenamientos en un fichero JSON de forma asíncrona. El controlador cargará este plan y sus correspondientes entrenamientos en la base de datos local.
- En caso de un nuevo plan, los datos se guardan directamente en la base de datos local. Este plan inicialmente no tendrá ningún entrenamiento asociado.



Plan Semanal

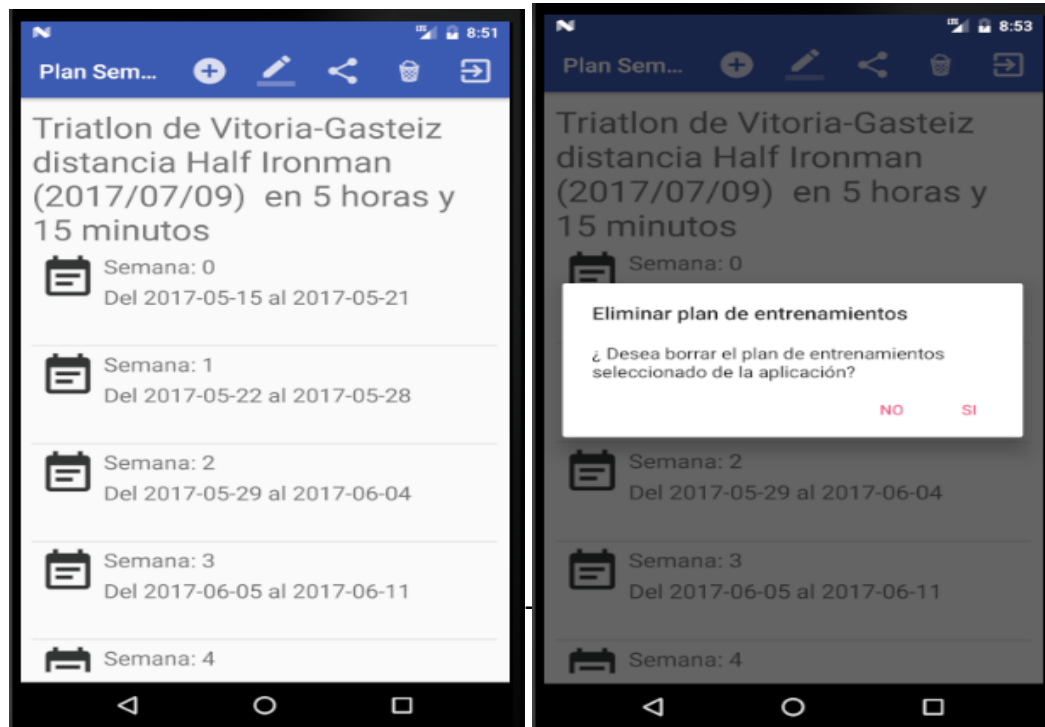
Una vez que entremos en el plan de entrenamientos obtendremos el listado del plan semanal que tenemos guardados en la base de datos local.

Esta vista está programada con la **clase PlanesSemana**. Para obtener el listado he utilizado un adaptador **ListViewAdaterPlanesSemanales** en el que he cargado los planes semanales de la base de datos local y se visualizan en un ListView.

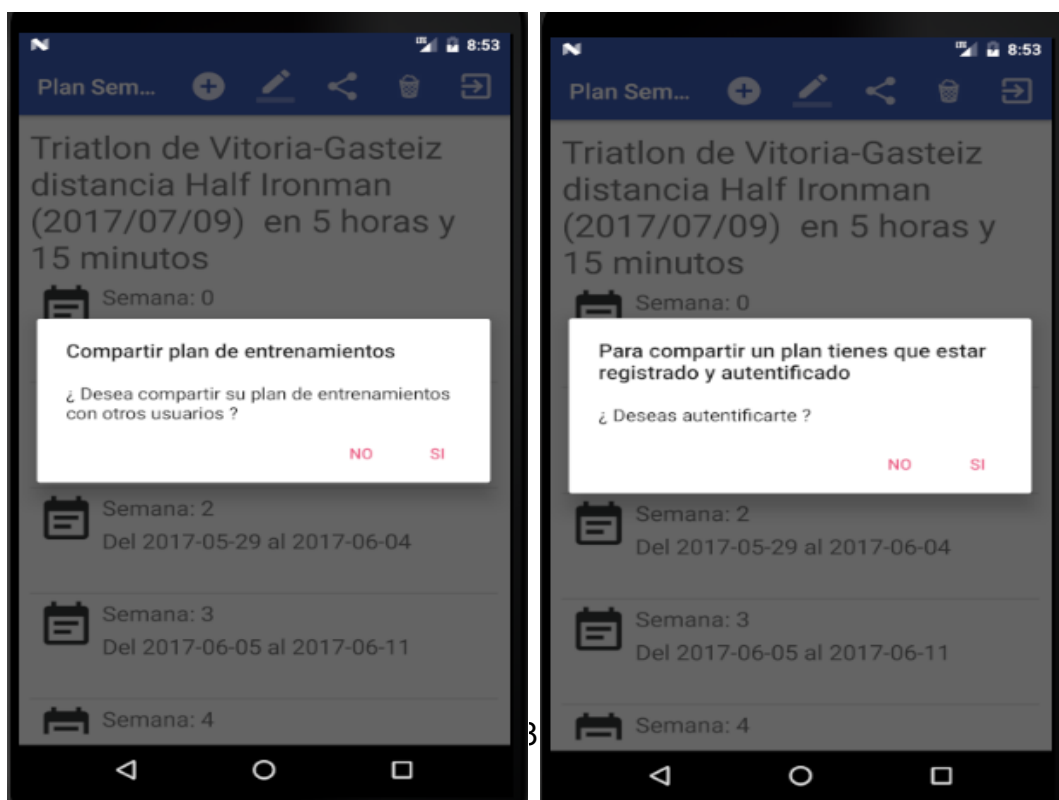
La vista está definida en el fichero **activity_planessemanales.xml** y cada elemento de la lista en el fichero **list_row_plansemanal.xml**.

El controlador implementa varias acciones que podemos hacer sobre el plan de entrenamientos:

- Podremos añadir nuevos entrenamientos para ello tendremos que pinchar sobre el icono “+” de la barra de herramientas.
- Podremos modificar el plan de entrenamiento. Utilizamos el controlador Nuevo ya explicado anteriormente para cargar los datos de la base de datos local y modificarlos.
- Pinchando sobre la papelera podremos borrar el plan de entrenamiento. Para ello nos solicitará una confirmación mediante un DialogBox. Si aceptamos el borrado el controlador eliminará el plan de la base de datos local.



Pinchando sobre el icono compartir podremos compartir el entrenamiento con otros usuarios. Para ello el controlador pide la confirmación y solicita al usuario que se registre. El controlador envía un fichero JSON mediante el método POST al programa **compartir.php** del servidor que lo almacena en la base de datos Mysql. Además el sistema envía otros datos como el email, idusuario y sesión para autentificar el usuario. Este programa devuelve un json con un un identificador de respuesta: 0 usuario incorrecto, 1 inserción correcta o 2 error en la inserción.



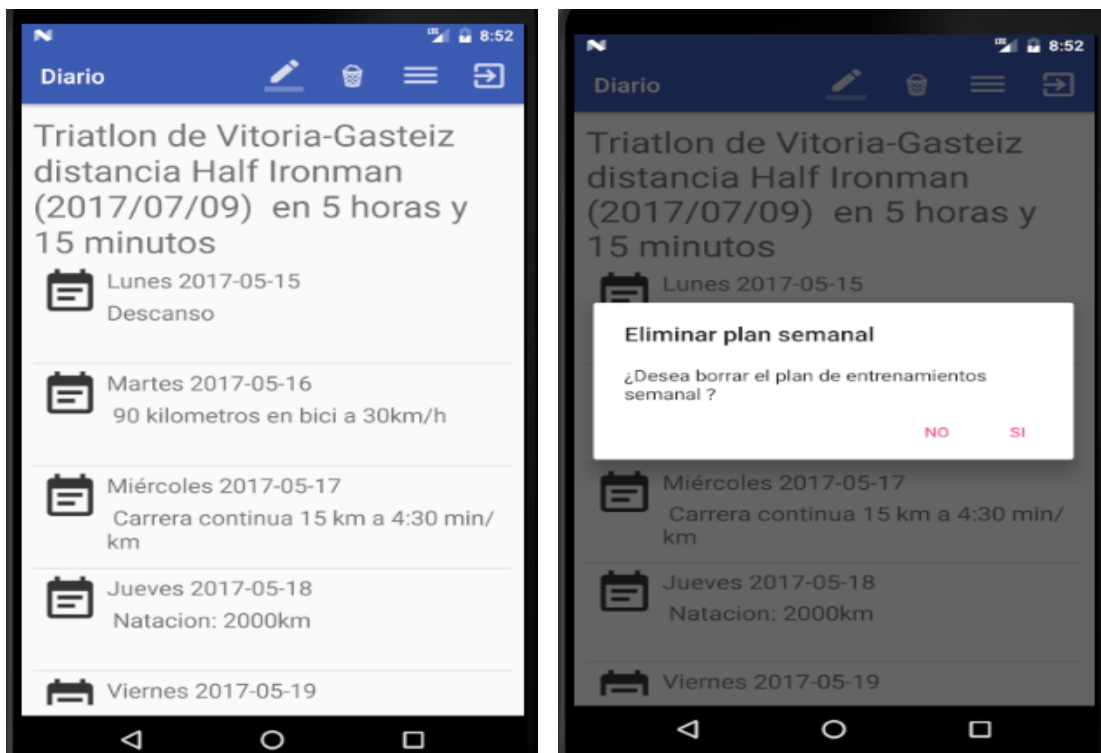
Plan Diario

Cuando pinchamos sobre elemento del listado del plan semanal obtendremos un plan diario a partir de los datos que tenemos guardados en la base de datos local.

Esta vista está programada con la **clase PlanesDiarios**. Para obtener el listado he utilizado un adaptador **ListViewAdaterPlanesDiarios** en el que he cargado los entrenamientos diarios a realizar a partir de la base de datos local y se visualizan en un ListView.

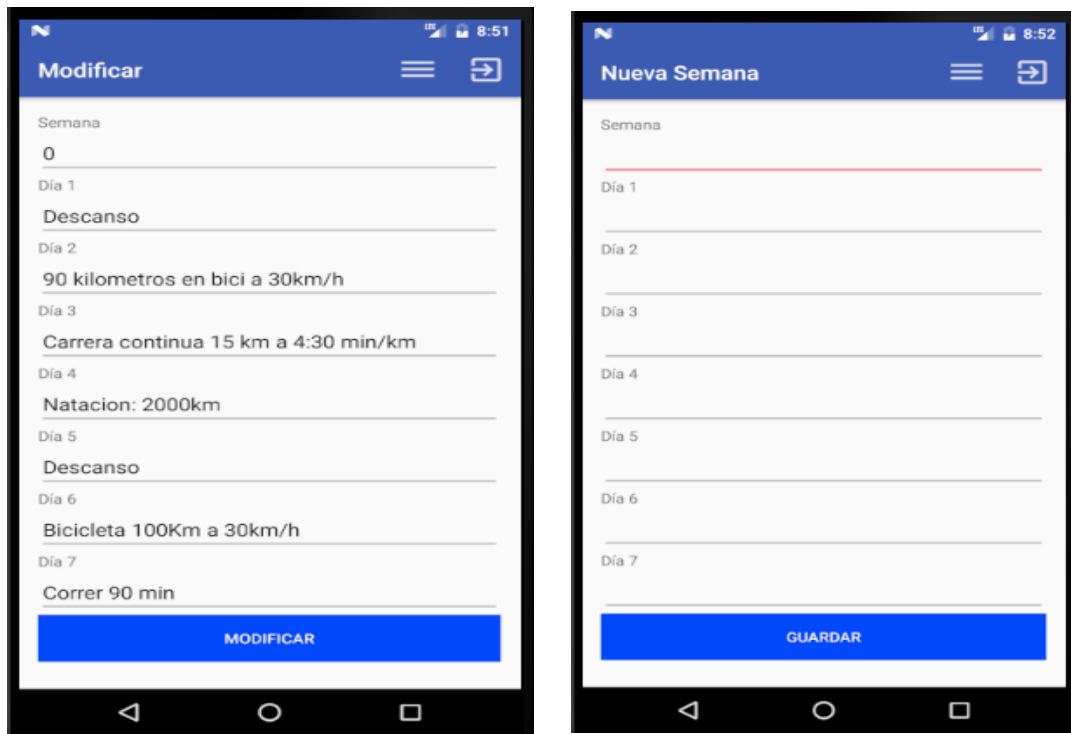
La vista está definida en el fichero **activity_planesdiarios.xml** y cada elemento de la lista en el fichero **list_row_planesdiarios.xml**.

Además, este controlador nos va a permitir eliminar un entrenamiento de la base de datos local pinchando sobre el icono papelera.



Nueva Semana

El controlador NuevaSemana nos va a permitir modificar o borrar un plan semanal de la base de datos local.



3.3. Pruebas

Se han realizado diferentes pruebas para comprobar el correcto funcionamiento de la aplicación. Por una parte se han realizado pruebas en diferentes terminales, pruebas en la funcionalidad de la aplicación, pruebas de conectividad y pruebas de usabilidad.

3.3.1.- Pruebas dispositivos

A la hora de desarrollo se han realizado pruebas sobre los siguientes emuladores:

- Emulador Nexus 5 con API 24
- Emulador Nexus 6 con API 24
- Emulador Nexus 10 con API 24

	Prueba 1
Tipo	Prueba de dispositivo
Descripción	Probar la aplicación sobre diferentes emuladores Nexus con diferentes pulgadas

Realización	Se ha ejecutado la aplicación y se ha instalado en el terminal y se ha instalado y ejecutado sin problemas.
Resultado obtenido	Correcto

Posteriormente se ha realizado pruebas sobre tres terminales reales:

- Teléfono móvil LG X5 con un Android 6.0
- Tablet Samsung Galaxy Tab 7 con Android 5.0.
- Teléfono móvil Orange Gova con un sistema Android 4.3.

	Prueba 2
Tipo	Prueba de dispositivo
Descripción	Probar la aplicación sobre un móvil LG X5 con un Android 6.0
Realización	Se ha descargado la aplicación en el terminal y se ha instalado sin problemas. El funcionamiento así como navegación es correcta.
Resultado obtenido	Correcto

	Prueba 3
Tipo	Prueba de dispositivo
Descripción	Probar la aplicación sobre una tablet Galaxy Tab 7 con Android 5.0.
Realización	Se ha descargado la aplicación en el terminal y se ha instalado sin problemas. El funcionamiento así como navegación es correcta.
Resultado obtenido	Correcto

	Prueba 4
Tipo	Prueba de dispositivo
Descripción	Probar la aplicación sobre un móvil Orange Gova Android 4.3
Realización	Se ha descargado la aplicación en el terminal y se ha instalado sin problemas. Se ha detectado la falta de iconos, en concreto los que tenían la opción inRoom.
Resultado obtenido	Incorrecto
Solución	Se ha optado por que todos los iconos sean visibles.

Al probar sobre el terminal 4.3 el resultado me he encontrado con un pequeño problema: los botones de la barra de herramientas con la opción ifRoom, en lugar de agruparse en un único icono que nos permitiese verse en forma de desplegable no aparecían directamente, por lo que he decidido que los botones aparezcan siempre (opción always). Esta solución probablemente se modifique en el futuro.

3.3.2.- Pruebas de orientación del dispositivo

Se han realizado diferentes pruebas de horientación del dispositivo:

- Pruebas dispositivo vertical
- Pruebas dispositivo horizontal

	Prueba 5
Tipo	Prueba orientación de dispositivo vertical
Descripción	Probar la aplicación sobre diferentes terminales en forma horizontal.
Realización	Se ha probado la aplicación en formato vertical, posicionando el dispositivo en formato horizontal para ver si funcionaba correctamente.
Resultado obtenido	Correcto

	Prueba 5
Tipo	Prueba orientación de dispositivo horizontal
Descripción	Probar la aplicación sobre diferentes terminales en forma horizontal.
Realización	Se ha probado la aplicación en formato horizontal, posicionando el dispositivo en formato horizontal para ver si funcionaba correctamente. En algunas pantallas no se ve el contenido total.
Resultado obtenido	Incorrecto
Solución	Se han añadido ScrollViews a pantallas como: Nuevo Plan, NuevoPlan Semanal,..etc

3.3.2.- Pruebas funcionales

Se han realizado diferentes pruebas funcionales de la aplicación:

	Prueba 6
Tipo	Prueba funcional para registro con datos vacíos o incorrectos
Descripción	Probar a realizar un registro con datos vacíos o incorrectos
Realización	Se ha probado a dejar los campos vacíos, poner un correo electrónico incorrecto y la aplicación nos ha detectado que los datos introducidos han sido los incorrectos.
Resultado obtenido	Correcto

	Prueba 7
Tipo	Prueba funcional para el registro de un usuario nuevo.
Descripción	Intentar registrarse en el sistema con un usuario nuevo.
Realización	Se ha intentado dar de alta el usuario aitor@kaixo.com con password "123456". La aplicación se ha conectado con el servidor y este le ha devuelto un fichero json indicado que el registro se ha realizado correctamente. Nos ha indicado que hay que validar el usuario.
Resultado obtenido	Correcto

	Prueba 8
Tipo	Prueba funcional para el registro con datos de usuario ya registrado.
Descripción	Intentar registrarse en el sistema con un usuario registrado anteriormente.
Realización	Se ha intentado dar de alta el usuario aitor@kaixo.com registrado anteriormente en el sistema. La aplicación se ha conectado con el servidor y este le ha devuelto un fichero json indicado que ese usuario ya existe en el sistema.
Resultado obtenido	Correcto

	Prueba 9
Tipo	Prueba funcional para el acceso con un usuario no validado.

Descripción	Intentar acceder al sistema con un usuario no validado.
Realización	Se ha intentado acceder con el usuario aitor@kaixo.com registrado anteriormente sin validar. La aplicación se ha conectado con el servidor y este le ha devuelto un fichero json indicado que ese usuario no está validado.
Resultado obtenido	Correcto

	Prueba 10
Tipo	Prueba funcional para el acceso con un usuario correcto y password incorrecto.
Descripción	Intentar acceder al sistema con un usuario con credenciales incorrectas.
Realización	Se ha intentado acceder con el usuario aitor@kaixo.com ya validado con un password incorrecto. La aplicación se ha conectado con el servidor y este le ha devuelto un fichero json indicado que el password es incorrecta.
Resultado obtenido	Correcto

	Prueba 11
Tipo	Prueba funcional para el acceso con credenciales correctas.
Descripción	Intentar acceder al sistema con un usuario con credenciales correctas.
Realización	Se ha intentado acceder con el usuario aitor@kaixo.com con password "123456". La aplicación se ha conectado con el servidor y este le ha devuelto un fichero json indicado que las credenciales son correctas y se han visualizado la pantalla de los planes de entrenamiento.
Resultado obtenido	Correcto

	Prueba 13
Tipo	Prueba funcional para generar un plan de forma automática
Descripción	Crear un plan de forma automática
Realización	Tras pinchar en el botón generar se han introducido los datos para obtener un plan de entrenamientos automático (Nombre: Maratón de Barcelona, Tipo: Maratón, fecha 12 de marzo y tiempo objetivo: 3 horas 10 minutos). El servidor nos ha devuelto un json que se ha cargado en la base de datos local de la aplicación.

Resultado obtenido	Correcto
--------------------	----------

	Prueba 14
Tipo	Prueba funcional para generar un plan nuevo de forma manual
Descripción	Crear un plan de forma manual
Realización	Tras pinchar en el botón nuevo se han introducido los datos para obtener un plan de entrenamientos automático (Nombre: Triatlón de Vitoria-Gasteiz, Tipo: Half Triatlón, fecha 9 de julio). Se ha cargado un plan vacío en la base de datos local.
Resultado obtenido	Correcto

	Prueba 15
Tipo	Prueba funcional buscar plan
Descripción	Buscar un plan compartido por otros usuarios y cargarlo en la aplicación.
Realización	Tras pinchar en el botón buscar la aplicación ha solicitado al servidor los planes compartidos y se han visualizado en la aplicación. Se han realizado diferentes filtros, por nombre, usuario e idioma. Finalmente se ha pinchado sobre un plan y se ha cargado correctamente en la aplicación.
Resultado obtenido	Correcto

	Prueba 16
Tipo	Prueba funcional para visualizar el plan diario.
Descripción	Visualiza el plan semanal y diario
Realización	Tras pinchar sobre el plan de entrenamientos, buscamos la semana actual marcada en la lista con un color mas oscuro, pinchamos sobre ella y visualizamos el listado de planes a realizar esta semana entre ellos el día de hoy aparece en oscuro.
Resultado obtenido	Correcto

	Prueba 17
Tipo	Prueba funcional baja de plan de entrenamientos

Descripción	Dar de baja un plan de entrenamientos
Realización	Una vez seleccionado el plan de entrenamientos pinchamos sobre el icono papelera. La aplicación mediante un mensaje nos indica si realmente queremos borrar el plan, una vez confirmado nos aparece el listado de planes sin el plan borrado.
Resultado obtenido	Correcto

	Prueba 18
Tipo	Prueba funcional compartir planes de entrenamiento
Descripción	Compartir plan de entrenamientos con otros usuarios
Realización	Una vez seleccionado el plan de entrenamientos pinchamos sobre el icono compartir. Si no estamos registrados el sistema nos lleva a la pantalla de registro, si lo estamos no indica si queremos compartir el plan con otros usuarios. Una vez confirmado la aplicación envía el plan de entrenamientos a compartir al servidor en un fichero json. Se ha comprobado que se ha compartido desde la opción de buscar planes compartidos.
Resultado obtenido	Correcto

	Prueba 19
Tipo	Prueba funcional importar planes
Descripción	Importar plan de entrenamientos
Realización	Tras acceder a nuevo plan pinchando sobre el icono importar la aplicación nos solicita la confirmación. Una vez aceptada la aplicación se conecta con el servidor obtiene los planes de entrenamiento compartidos por el usuario y los carga en la aplicación.
Resultado obtenido	Correcto

3.3.3.- Pruebas conectividad

Se han realizado diferentes pruebas con y sin conectividad.

	Prueba 20
Tipo	Prueba con conectividad a Internet
Descripción	Probar el funcionamiento de la aplicación con conexión a Internet.
Realización	Se ha probado la aplicación con conexiones Wifi y 2.5G 3G y 4G. La aplicación no consume muchos datos por lo

	que el resultado ha sido satisfactorio en todos los casos
Resultado obtenido	Correcto

	Prueba 21
Tipo	Prueba sin conectividad a Internet
Descripción	Probar el funcionamiento de la aplicación sin conexión a Internet.
Realización	Se ha probado la aplicación sin conexión a Internet y la aplicación daba error al intentarse conectar con el servidor.
Resultado obtenido	Incorrecto
Solución	Se ha modificado el código para comprobar la conexión antes de enviar o recibir información del servidor. Esta solución en los emuladores no funciona.

3.3.3.- Pruebas idioma

Se han realizado pruebas para la comprobación que la traducción de la aplicación al inglés y castellano ha sido correcta:

	Prueba 22
Tipo	Prueba idioma inglés
Descripción	Probar la traducción de la aplicación al inglés
Realización	Se ha probado la aplicación modificando los ajustes de la aplicación al idioma Inglés-UK. Se ha probado los diferentes mensajes y textos aparecen correctamente traducidos en la aplicación.
Resultado obtenido	Correcto

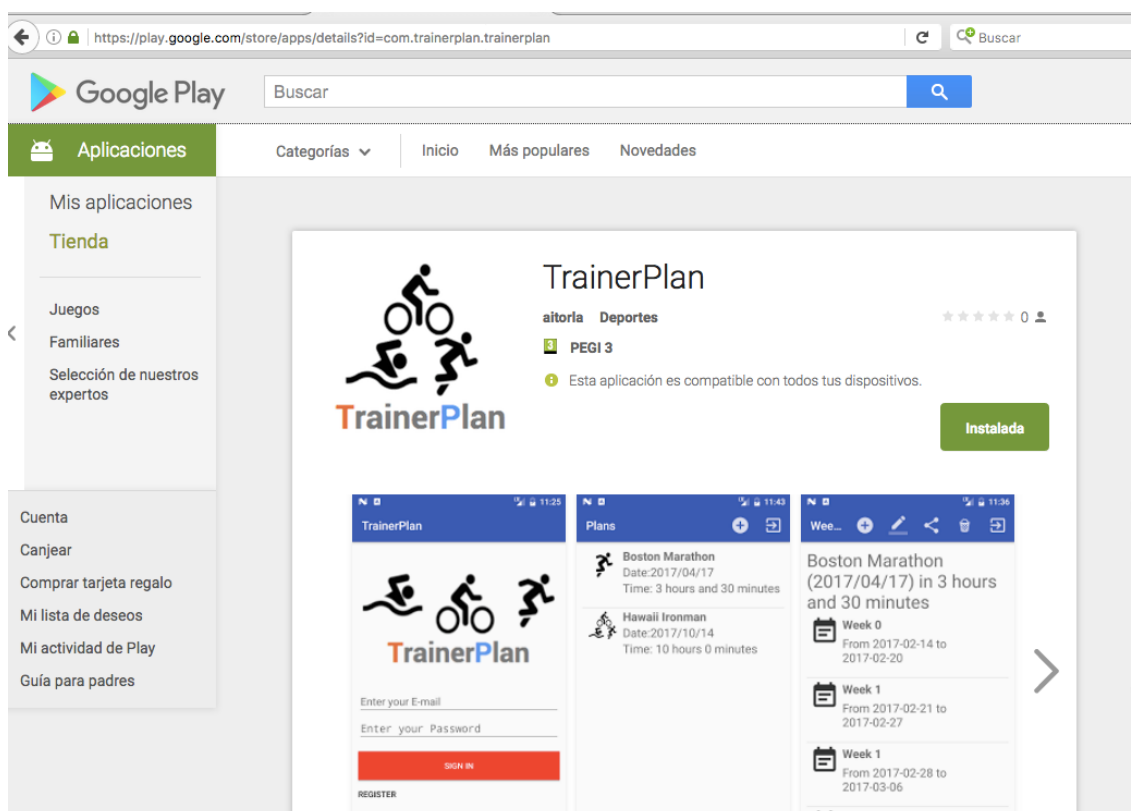
	Prueba 23
Tipo	Prueba idioma castellano
Descripción	Probar la traducción de la aplicación al castellano
Realización	Se ha probado la aplicación modificando los ajustes de la aplicación al idioma Español-España. Se ha probado los diferentes mensajes y textos aparecen correctamente traducidos en la aplicación.
Resultado obtenido	Correcto

3.3.5.- Pruebas usabilidad

Se ha probado la aplicación en usuarios reales para observar el modo en el que utilizan la aplicación y las dificultades que encontraban en el uso de la misma. La opción por la que se ha optado a la hora de visualizar los botones en la barra de herramienta tal vez pueda mejorarse.

3.4. Publicación

Una vez realizadas la pruebas y comprobar el correcto funcionamiento de la aplicación, se ha firmado y se ha subido a Google Play para que esté disponible para los usuarios.



Se puede realizar una búsqueda con la palabra TrainerPlan o directamente acceder a través de la URL:

<https://play.google.com/store/apps/details?id=com.trainerplan.trainerplan>

Inicialmente se ha configurado el inglés como idioma predefinido de la aplicación con la traducción al castellano.

La primera vez que se ha subido la aplicación me he llevado una desagradable sorpresa: la aplicación no era compatible con ningún dispositivo. Lo que realmente estaba ocurriendo es que utilizaba una librería **commons-io-2.4.jar** para realizar un control de los errores de entrada salida. Al final he quitado esa librería y he ajustado la aplicación. Así he conseguido que la aplicación sea compatible con 12722 dispositivos. En problema se aprecia en las siguientes dos pantallas:

INFORMACIÓN DEL APK		
com.trainerplan.trainerplan		
Código de versión 7	Nombre de la versión 1.0	Tamaño 3,48 MB
Dispositivos Android compatibles	12722 dispositivos	
Niveles de API	15+	
Diseños de pantalla	4 diseños de pantalla	
Ubicaciones	predeterminado + 82 idiomas	
Características	1 función	
Permisos necesarios	2 permisos	
Versiones de OpenGL ES	1.0+	
texturas OpenGL	todas las texturas	
Subido	3/1/2017 22:47:28 PST	
Compilación SHA1 del APK	565ea472ee383c0b775234a760475160671e0018	
Novedades de esta versión	Inglés (Reino Unido) – en-GB	
	Añadir texto	

INFORMACIÓN DEL APK		
com.trainerplan.trainerplan		
Código de versión 1	Nombre de la versión 1.0	Tamaño 3,71 MB
Niveles de API	15+	
Diseños de pantalla	4 diseños de pantalla	
Ubicaciones	predeterminado + 82 idiomas	
Características	1 función	
Permisos necesarios	2 permisos	
Versiones de OpenGL ES	1.0+	
texturas OpenGL	todas las texturas	
Plataformas nativas	commons-io-2.4.jar	
Subido	2/1/2017 2:09:04 PST	
Compilación SHA1 del APK	e0bebbe1870ca3d21bc99092fe9bf2e241033398	
Novedades de esta versión	Inglés (Reino Unido) – en-GB	
	Añadir texto	

4. Valoración económica

4.1.- Costes de la aplicación

Para la puesta en marcha de la aplicación he utilizado recursos que ya tenía contratados con anterioridad:

- Servidor Hosting 1and1 (80€/anuales)
- Dominio registrado trainerplan.com (incluido en el hosting)
- Alta como desarrollador en GooglePlay (20€ pago inicial)

Además el propio servidor de hosting ofrece un certificado de seguridad para un dominio que no lo estaba utilizando.

4.2.- Monetización de la aplicación

La idea inicial de la aplicación es poder ofrecer una herramienta al equipo deportivo al que pertenezco, por lo tanto no tengo pensado explotar económicamente la aplicación. Actualmente se ofrece de forma gratuita en Google Play.

Dependiendo del éxito que pueda tener la aplicación se podía pensar en facturaciones inn-app o incluso en una especie de “Market Place” de planes de entrenamiento donde los entrenadores profesionales puedan vender sus planes de entrenamiento a los deportistas y en el que TrainerPlan obtuviese un porcentaje de venta.

Otra de las opciones para monetizar la aplicación podría ser la inclusión de publicidad mediante Google AdMob.

5. Conclusiones

Con la realización del TFM he aplicado en un proyecto real los contenidos aprendidos en otras asignaturas del Master.

Otra de las cosas que he aprendido es que el prototipo puede ser un acercamiento a la idea que se tiene de la aplicación pero que al final hay que hacer un ajuste sobre el mismo. Además a medida que estás desarrollando el proyecto surgen nuevas ideas de funcionalidades y diseño.

La realización de la planificación inicial, y sobre todo el marcado de una fecha de entrega de los PECs, me ha ayudado con la realización y seguimiento del TFM. No se han realizado cambios significativos sobre la planificación inicial. En muchas ocasiones existen desvíos en los tiempos de ejecución, pero este no ha sido el caso.

He añadido nuevas funcionalidades que no estaban previstas antes de la entrega final: Traducción de la aplicación a varios idiomas, planificación en diferentes disciplinas y deportes, importación de planes de entrenamiento, búsqueda de planes compartidos por otros usuarios. Lo que ha hecho que la fase de implementación se haya alargado en el tiempo. Los usuarios que han probado la aplicación han visto su potencial y me han planteado nuevas funcionalidades que se pueden implementar en el futuro.

Además tengo pensado realizar en un futuro la nueva web de TrainPlan que disponga de los servicios que dispone la nueva aplicación desarrollada. Además de la aplicación híbrida o Nativa que permita a los usuarios de IOS utilizar la aplicación.

Finalmente, lo mejor de la aplicación es que sea usada. Yo personalmente ya me he marcado un objetivo deportivo y utilizaré la aplicación TrainerPlan para conseguirlo.

6. Glosario

- FERTRI: Federación Española de Triatlón.
- IoT: Internet of Things. Internet de las cosas.
- TFM: Trabajo Final de Máster.
- MVC: Modelo vista controlador.
- SOA: Service Oriented Architecture. Arquitectura Orientada a Servicios.
- LAMP: Linux Apache Mysql Php.

7. Bibliografía

Luke Humphrey with Keith and Kevin Hanson Velopress, segunda edición, (2016). Hansons Marathon Method: Run Your Fastest Marathon. Velopress.

Martín Fiz e Ignacio Romo (2003). Correr con Martín Fiz. Vitoria-Gasteiz: La esfera de los libros.

Dan Golding (2012). Triathlon For Beginners: Everything you need to know about training, nutrition, kit, motivation, racing, and much more.

Eduard Barceló (2015). Entrenamiento para ultra trail - Cómo sobrevivir a una carretera de larga distancia. Desnivel.

11 Mejores Apps para hacer deporte:

<http://computerhoy.com/listas/apps/11-mejores-apps-hacer-deporte-android-e-iphone-5784>

NY Marathon Finisher Demographics

<http://www.tcsnycmarathon.org/about-the-race/results/finisher-demographics>

Arquitectura Orientada a Servicios

https://es.wikipedia.org/wiki/Arquitectura_orientada_a_servicios

Modelo Vista Controlador

<https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

8. Anexos

Se adjuntan los siguientes anexos:

- **Manual de instalación:** Indica la forma de instalar la aplicación desde Google Play o mediante el apk que se adjunta en la entrega.
- **Manual de usuario:** Incluye el manual de usuario.
- **Manual de configuración:** Incluye las instrucciones de configuración del cliente y del servidor en un entorno de desarrollo y/o producción.

Además se añade al proyecto el código fuente y la aplicación:

- **Código Fuente:** La carpeta `CodigoFuente` Contiene las carpetas `ClienteTrainerplan` y `ServidorTrainerplan` con el código fuente del cliente y del servidor respectivamente. Además en la carpeta `ServidorTrainerplan` se ha incluido la base de datos del servidor en el fichero `trainerplan.sql`.
- **Aplicación:** Adjuntamos la aplicación firmada `trainerplan.apk`