

Anteproyecto Trabajo Fin de Grado
Aplicacion para moviles inteligentes para el deporte
Application for smartphone for sport

Director
David Santo Orcero - correo:dsanto@lcc.uma.es

Alumno
Rubén Gutiérrez Cazorla - correo:gutyruben@gmail.com

Departamento
Lenguajes y Ciencias de la Computación

2 de enero de 2018

1. Introducción

En los últimos años, los móviles han sufrido un rápido avance que aún no hemos asimilado. La llegada de internet a estos realizó un gran salto y posteriormente el salto a los smartphone. Con ellos hubo un cambio de paradigmas: en la forma de comunicarnos, de comprar, alimentarnos y como no: también la forma de hacer deporte.

Con la llegada del GPS a nuestros móviles nos han aportado una gran herramienta para ser utilizada para mapas y poder llegar a sitios de una manera sencilla. Por otro lado se puede utilizar desde un enfoque más deportivo y usarse para una medición precisa de nuestros entrenamientos.

Realizar una aplicación para dispositivos móviles que permite registrar todas tus actividades deportivas: vía GPS, con ella podemos tener un historial y seguimiento de todas nuestras actividades, como: correr, caminar, ciclismo y mucho más.

Para llevar el control de nuestras actividades realizamos un seguimiento de ciertos parámetros como: la distancia, tiempo, calorías quemadas, velocidad, altitud y mucho más.

Con el registro de podremos llevar un control sencillo de las actividades y progresar con interesantes datos y estadísticas.

Para los que quieren ponerse en forma de manera sencilla, hacer ejercicio, adelgazar corriendo, empezar a correr con un asistente personal, terminar esa maratón o hacer carreras populares.

La principal funcionalidad de la aplicación es la medición del ejercicio, deportivo realizado. Donde el usuario antes de iniciar la actividad, necesitará iniciar la aplicación y en la primera vista, pulsar un botón, el cual activará la funcionalidad GPS y comenzará a realizar un seguimiento. El conjunto de las ubicaciones GPS y el tiempo en el que realiza la actividad, nos dará como resultado la velocidad. También se podrá obtener otras datos como las calorías (aproximadas) quemadas en

dicho entrenamiento. Esto quedará almacenado en la memoria del teléfono para poder usar los datos.

Dicha base de dato ha de ser lo más liviana posible, puesto que no todos los terminales disponen de una gran capacidad y con la arquitectura que más se adecue en relación: espacio/velocidad*.

*Dicha relación se refiere al espacio que ocupa la base de dato y la velocidad que tenga de transmisión de la información.

En otra vista tendremos una lista de ejercicios deportivos realizados por el usuario, donde podrá visualizar los entrenamientos realizados, ordenados por fecha, siendo la última realizada la primera en aparecer. También contendrá datos relevantes del entrenamiento como: distancia recorrida, tiempo y calorías quemadas, entre otros. Para ello nos apoyaremos en las bases de datos.

Enfocado para el usuario que quiera afinar más en el campo de la quema de grasas, tendremos una vista que permita introducir: su peso, sexo, altura y edad. Con estos tres parámetros y gracias a la fórmula de Harris-Benedict, es la que vamos a utilizar para calcular la cantidad de calorías quemadas, y se calcula de la siguiente manera:

- Mujeres: $[655 + (9.6 \times \text{Peso kg})] + [(1.8 \times \text{Altura cm}) - (4.7 \times \text{Edad})] \times \text{Factor actividad}$
- Hombres: $[66 + (13.7 \times \text{Peso kg})] + [(5 \times \text{Altura cm}) - (6.5 \times \text{Edad})] \times \text{Factor actividad}$

Cuando se nombra factor de actividad se refiere a lo siguiente:

- Personas sedentarias (hace poca actividad física): 1.2
- Actividad ligera (hace actividad física 1 a 3 veces por semana): 1.375
- Actividad moderada (hace actividad física 3 a 5 veces por semana): 1.55
- Actividad intensa (hace actividad física 6 a 7 veces por semana): 1.725
- Actividad extremadamente alta (atletas profesionales mucha actividad física): 1.9

Esta actividad irá de la mano del historial que tengas en la vista antes mencionada. Y no se incluirá el factor actividad en la vista en la que tenemos la listas de los ejercicios realizados si no que se usará para la vista que mencionamos a continuación.

Por último y no menos importante una vista en la que nos proporciona el resultado de todo el trabajo realizado, y el cálculo total de calorías quemadas, usando el factor actividad, puesto que ya sabremos cuantas veces a la semana se está realizando la actividad.

2. Estado del arte

2.1. Introducción

Para poder comenzar a hablar de nuestra aplicación y sus beneficios tenemos que entender que es la actividad física. Pues según la OMS “la actividad física cualquier movimiento corporal producido por los músculos esqueléticos que exija gasto de energía.” También nos indica los factores que provocan la mortalidad mundial y de cómo una actividad regular tiene efectos beneficiosos en la salud y previene numerosas enfermedades.

En la actualidad vivimos en una sociedad que ha ido añadiendo poco a poco una actividad deportiva en sus vidas, además de una alimentación sana. Sin importar lo que los mueve ya sea sobre peso, una búsqueda más profesional del ejercicio físico o para encontrarse bien, encuentran cómoda una aplicación que supervise su progreso.

Como observamos, en el libro Anuario de Estadísticas deportivas 2016, perteneciente al Ministerio de educación, Cultura y Deporte y editada por la Secretaría General de Documentación y Publicaciones, observamos distintos indicadores de aumento en la realización de actividades deportivas, si lo hacemos desde un punto de vista empresarial observamos en la siguiente tabla como hay un aumento de actividades deportivas en empresas vinculadas al deporte por actividad económica:

Como vemos en la figura (1) existe un incremento desde el año 2011 al 2015 de actividades deportivas. Ergo si las empresas ven un incremento en sus actividades deportivas esto significa que hay una demanda de las misma siendo un indicador de un aumento de usuarios que realizan actividades deportivas.

Otro gran indicativo, más directo que el de las empresas vinculadas al deporte por actividad económica, es la información relativa al gasto de consumo de los hogares.

En esta figura (2) nos muestra una disminución del gasto vinculado al deporte, pero teniendo en cuenta que el estudio se realiza en un periodo que abarca desde el 2008 al 2014 y teniendo en cuenta “En diciembre de 2008 la economía entró en recesión y consiguió salir de ella en mayo de 2010, aunque por poco tiempo ya que el PIB siguió cayendo en tasa interanual hasta que en abril de 2012 encadenó dos trimestres consecutivos de descensos y España volvió a entrar en recesión.” (Periódico La Razón: <http://www.larazon.es/economia/fechas-clave-de-la-crisis-economica-en-espana-CX4077260>). Pero este gasto no tiene porque influir en la actividad deportiva sin costo como puede ser correr o el ciclismo y teniendo en cuenta que la cantidad no disminuye considerablemente se deduce que aún en tiempos de desdicha económica se mantiene esa cultura del deporte.

Centrandonos en la población española la información relativa a los hábitos y prácticas deportivas desde una edad de 15 años en adelante.

En la figura (3) se observa que en cuanto al deporte practicado en 2015 es superior al 2010 con una diferencia de 9,2 % de la población. Siendo las Mujeres las que más se han lanzado a la realización de deporte. En cuanto a la edad comprendida entre 15 y 24 años este aumento a supuesto 18,2 %. Siendo unos indicadores más que importantes para lanzar una aplicación para dispositivos móviles, pues este sector es el que más interiorizado tiene estas tecnologías.

Paremos un momento para analizar el estado actual de la sociedad española en el campo de la tecnología. En cuanto a la cantidad de la población adulta que tiene algún tipo de móvil es del 96 %, haciéndonos ver que hay una normalidad en nuestra sociedad al uso de teléfonos pero lo mas llamativo es que el estudio nos muestra un 80 % de la población con teléfonos inteligentes. Entre las acciones que realizamos los españoles nos encontramos en primer lugar tenemos con un 51 % aplicación de mensajería, un 38 % para reproducir vídeos, en cuanto a la cifra mas baja tenemos con un 25 % el uso para jugar, por otro lado tenemos a las finanzas con un 33 % y por último tenemos con un 39 % de la población que usa servicios de mapas en sus dispositivos móviles. Como nos explica XatakaMovil (En: <https://www.xatakamovil.com/movil-y-sociedad/espana-territorio-smartphone>).

Como resumen tenemos que en “2015 el 53,5 % de la población de 15 años en adelante practicó deporte en el último año. La mayor parte de ellos, el 86,3 %, con gran intensidad, al menos una vez

	VALORES ABSOLUTOS					DISTRIBUCIÓN PORCENTUAL				
	2011	2012	2013	2014	2015	2011	2012	2013	2014	2015
TOTAL	24.655	26.011	26.697	28.735	31.139	100	100	100	100	100
Actividades deportivas	16.827	18.167	19.270	21.506	24.003	68,2	69,8	72,2	74,8	77,1
Fabricación de artículos de deporte	175	175	174	177	204	0,7	0,7	0,7	0,6	0,7
Comercio al por menor de art. deportivos	7.653	7.669	7.253	7.052	6.932	31,0	29,5	27,2	24,5	22,3
<i>En porcentaje del total de empresas</i>										
TOTAL						0,8	0,8	0,8	0,9	1,0
Actividades deportivas						0,52	0,57	0,61	0,69	0,75
Fabricación de artículos de deporte						0,01	0,01	0,01	0,01	0,01
Comercio al por menor de art. deportivos						0,24	0,24	0,23	0,23	0,22

Fuente: INE. Directorio Central de Empresas

Figura 1: Empresas vinculadas al deporte por actividad económica

	2008	2009	2010	2011	2012	2013	2014
Total gasto vinculado al deporte (Millones de euros)	4.865,9	4.555,1	4.693,2	4.778,8	4.468,8	4.570,4	4.211,4
Gasto medio por persona (Euros)	106,7	99,1	101,7	103,2	96,5	99,0	91,6
<i>Gasto vinculado al deporte en porcentaje del total de gasto</i>	0,9	0,9	0,9	0,9	0,9	0,9	0,9

Fuente: INE. Encuesta de Presupuestos Familiares. Base 2006

Figura 2: Gasto en bienes y servicios vinculados al deporte

(En porcentaje de la población total investigada de cada colectivo)		
PRÁCTICA DEPORTIVA SEMANAL		
	2010	2015
TOTAL	37,0	46,2
SEXO		
Hombres	45,4	50,4
Mujeres	28,8	42,1
EDAD		
De 15 a 24 años	57,9	76,1
De 25 a 54 años	40,2	53,2
De 55 y más años	22,2	26,0
NIVEL DE ESTUDIOS		
Educación primaria, secundaria	33,2	39,4
Educación superior o equivalente	54,9	64,1

Fuente: MECD. Encuesta de Hábitos Deportivos en España

Figura 3: Personas que practican deporte semanalmente según características personales

a la semana. Los resultados muestran que un 70,6 % de la población suele realizar esta actividad y el 68,2 % al menos una vez por semana. Respecto al hábito de andar y su relación con la práctica deportiva, un 81,1 % de la población manifiesta que suele andar o practicar deporte semanalmente. La encuesta investiga asimismo tanto la asistencia.“ (dicho en en el libro Anuario de Estadísticas deportivas 2016).

También existen datos que nos indica que hay una parte de ella la cual es obesa, como estima el boletín de la OMS en que existe un tanto por ciento de esta población practica de porte un tanto por ciento a la semana por lo tanto la relevancia de la aplicación es mas que significativa por una ayuda del control de los datos biológicos del sujeto en la practica deportiva

Según “Las recomendaciones mundiales sobre la actividad física para la salud“ escritas por la OMS y nos muestra tres grupos acotados en función de la edad, desde niños a ancianos y a cada grupo se le introduce un mínimo de tiempo, tipo de actividad e intensidad de la misma. Con el fin de evitar lesiones cardíacas, mejora de la musculatura.

Hay que tener en cuenta que no podemos darle la misma actividad física a todos los usuarios y eso es fundamental. Lo que podemos sacar de este estudio es que todos necesitamos una actividad física adecuada a nuestro momento

Como inicio a las aplicaciones y cerrando la introducción vamos a visualizar las aplicaciones y la cantidad de usuarios y descargas que hay:

- Runtastic:
 - “Runtastic hace público su número de usuarios: El éxito de la compañía de salud y forma física móvil continúa y celebran 30 millones de descargas de sus apps, 25 millones de usuarios móviles y 10 millones de usuarios registrados en el sitio Runtastic fitness. Además lanza su línea de hardware en EEUU e inicia una cooperación con Power Music.“ (<https://www.socialetic.com/runtastic-25-millones-de-usuarios-y-una-descarga-de-su-app-cada-segundo.html>)
- FITAPP GPS - Correr y Caminar:
 - Tiene más de un millón de descargas (Dato aportado por Play Store) y la cantidad de usuarios es desconocida.
- Strava GPS Correr Ciclismo:
 - Tiene más de diez millón de descargas (Dato aportado por Play Store) y la cantidad de usuarios es desconocida.
- RunKeeper - GPS Correr Caminar:
 - Tiene más de diez millón de descargas (Dato aportado por Play Store) y la cantidad de usuarios es desconocida.

2.2. Aplicaciones analizadas

2.2.1. Runtastic

Aqui va la imagen: Imagenes runtastic

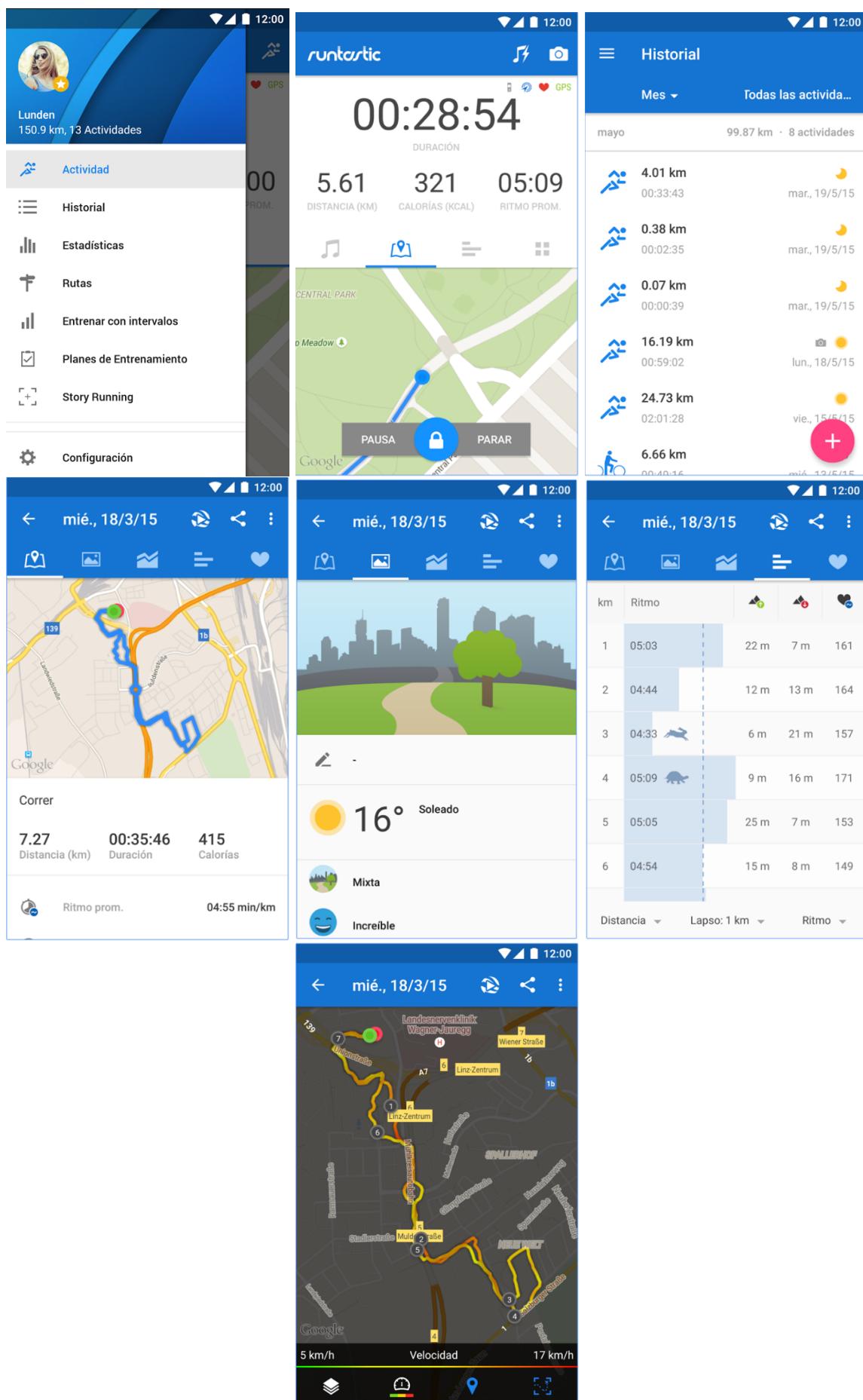


Figura 4: Múltiples imágenes de runtastic

- FUNCIONES DE LA APP

- Tu entrenador personal de running y fitness: correr, caminar, ir en bici, etc.)
- Entrenador por voz: tu entrenador personal te dice cómo lo vas haciendo según tus preferencias
- Android Wear 2.0: puedes dejar el móvil en casa y ver tus estadísticas en tu smartwatch
- Grupos: crea un grupo con tus amigos para motivarlos
- Objetivo de running anual: ¡Ponte un objetivo! Si quieres empezar a correr o habituarte a hacer ejercicio, ¡aquí encontrarás tu motivación!
- Control de zapatillas: registra el kilometraje de tus zapatillas de running para que sepas cuándo debes cambiarlas por unas nuevas
- Clasificación: ¿te has propuesto adelgazar corriendo con tus amigos? ¡A ver quién puede correr más!
- Seguimiento y motivaciones en vivo: recibe mensajes de apoyo de tus amigos en vivo y en directo
- Powersong: reproductor de música integrado y powersong. ¡Así correr o caminar para ponerse en forma cuesta mucho menos!
- Runtastic Wearable Connect: las estadísticas de tus actividades, ahora en la pantalla de tu Runtastic Orbit o Runtastic Moment
- Integración con Google Fit y MyFitnessPal

2.2.2. FITAPP GPS - Correr y Caminar

Aquí va la imagen: imágenes de fitfapp

- FUNCIONES DE LA APP

- Registra la duración, la distancia y el ritmo a través de GPS
- Seguimiento de peso y cuenta calorías según tu especificaciones (ayuda a mantener la pérdida de peso)
- Comentarios de voz (duración total, calorías, distancia, velocidad actual, ritmo medio)
- Función SNAP (toma SNAPS geniales de tus proezas deportivas y comparte con tus amigos)

2.2.3. Strava GPS Correr Ciclismo

Aqui va la imagen: imagenes strava

- FUNCIONES DE LA APP

- Monitoriza gratis las carreras, los recorridos y otras actividades
 - Seguimiento de actividades: Durante la actividad y después de esta, consulta todas las estadísticas importantes como la distancia, el ritmo, la velocidad, el desnivel positivo y las calorías quemadas. También dispondrás de un mapa interactivo de tu actividad.

- Reto personal: Participa en los retos personales creados para que superes tus propios límites
- Segmentos: ¿Cuál es tu subida de montaña favorita? ¿Y tu tramo de carretera preferido? Mira a ver cómo te ha ido en ciertos momentos de tu actividad con los segmentos de Strava
- Conecta con tus amigos y compañeros
 - Entrenamientos siempre acompañados: Sigue a tus amigos, compañeros de entrenamiento y deportistas profesionales para ver sus actividades y animarlos con los kudos y los comentarios
 - Clubes: Sea cual sea tu deporte favorito, seguro que existe un club; desde las marcas deportivas más importantes del mundo, hasta tus colegas del barrio con los que sales a correr. Únete a los que más te gusten para no perderte ningún evento, participar en los foros o simplemente para descubrir las últimas novedades
 - Fotos de las actividades: Presume de los mejores momentos de tu carrera o recorrido en bici
 - Competición por diversión: Dalo todo y consigue la mejor marca de tiempo en las tablas de resultados de segmentos
 - Strava y las redes sociales: Comparte todos los detalles de las actividades en Facebook, Instagram y Twitter
- Utiliza tu tecnología favorita
 - Compatible con la mayoría de dispositivos: Strava funciona con la mayoría de dispositivos con GPS incorporado, entre los que se incluyen los relojes para corredores, los ordenadores para ciclistas y los monitores de actividades.
 - Ritmo cardíaco: Entrena con un monitor de ritmo cardíaco para obtener más datos sobre tu rendimiento.
 - Android Wear: Gracias a los relojes Android Wear 2.0, ya no necesitarás tu teléfono móvil para registrar y subir una actividad de Strava.

2.2.4. RunKeeper - GPS Correr Caminar

Aqui va la imagen: imágenes runkeeper

■ FUNCIONES DE LA APP

- Monitoriza tus actividades deportivas y diviértete haciéndolo
 - Verás estadísticas detalladas de tu ritmo, distancia y tiempo
 - Obtén estadísticas, progreso y entrenamiento por voz a través de tus auriculares
 - Toma fotos durante el entrenamiento y compártelas mientras tanto.
- Mide la evolución de tu rendimiento
 - Historial detallado de tus actividades para ver como lo estás haciendo.
 - Notificaciones cuando consigues nuevas marcas personales e hitos.
 - Mide tus progresos para conseguir tus objetivos y metas.
 - Sigue detallados planes de entrenamiento para conseguir objetivos deportivos específicos.
 - Convierte cualquier actividad en una ruta para poder realizarla más adelante.

- Comparte con amigos
 - Publica tus actividades , logros y planes en para que los vean tus amigos de Facebook, Twitter
 - Permite que tus fans vean en directo sobre mapas tus entrenamientos y carreras mientras los llevas a cabo (se necesita suscripción a RunKeeper Elite)
- Obtén una visión más amplia de tu salud en RunKeeper.com
 - Integra los datos de tu actividad con más de 70 aplicaciones y servicios de terceros incluyendo , Withings, Zeo, Garmin y muchos más, para conseguir una conocimiento más profundo de tu salud general

2.2.5. Corre con Map My Run

Aquí va la imagen: Multiples imágenes de Mapmy

- FUNCIONES DE LA APP
 - Sigue y sitúa en el mapa tus entrenamientos
 - ◊ Registra además más de 600 actividades deportivas como, por ejemplo, montar en bicicleta, entrenamientos en gimnasio, cross training o yoga.
 - ◊ Obtén información audio en cada entrenamiento que sigas mediante GPS con comentarios de voz personalizables (ritmo, distancia, calorías o desnivel).
 - ◊ Conecta más de 400 dispositivos, importa y analiza información, y almacénala en un solo lugar.
 - ◊ Conecta tus zapatillas y sigue tus kilómetros automáticamente. Recibirás una notificación cuando sea el momento de cambiarlas para evitar lesiones.
 - ◊ Utiliza la función Rutas para encontrar nuevos lugares por los que correr.
 - Conecta otras apps y dispositivos
 - ◊ Sigue tu actividad con la app Android Wear y consulta tu progreso a simple vista.
 - ◊ Deja que las zapatillas UA Record Equipped sigan de manera automática tu actividad y sincronicen la información con la app.
 - ◊ Sincroniza tu información con las mejores apps y dispositivos como Google Fit, Android Wear, Garmin, Fitbit o Jawbone, entre otros.
 - ◊ Controla tu consumo calórico con MyFitnessPal.
 - Únete a la comunidad
 - ◊ Consulta la actividad de tus amigos en el Feed y mantén la motivación.
 - ◊ Comparte tus entrenamientos en Facebook y Twitter.
 - ◊ Únete a desafíos para competir con otros usuarios, subir puestos en la clasificación y ganar magníficos premios.

3. Herramientas utilizadas

3.1. Introducción

En nuestros días la mejora en los navegadores internos de los dispositivos está haciendo posible el auge de aplicaciones híbridas multiplataforma, acercándolas cada día mas a la experiencia

y rendimiento de las aplicaciones nativas pero con la ventaja de ser implementadas una única vez.

Este tipo de tecnologías nos permiten crear aplicaciones móviles con tecnología web (CSS, HTML y JS) que pueden ser ejecutadas y distribuidas en el market de cada plataforma.

Lo que aporta Ionic es un SDK que facilita la construcción de pantallas (botones, listas, ...) respetando la guía de estilo de cada plataforma, de forma transparente al desarrollador, es decir, que inicialmente no tenemos que añadir una sola línea de código para conseguirlo.

Esta tecnología no es nueva y siempre ha estado ligada con el framework AngularJS y Apache Cordova, aquí tenéis un tutorial en el que ya hablábamos de estas tecnologías.

Así que este trinomio se ha seguido manteniendo con la versión 2 de Angular, que como ya vimos en este otro tutorial mejora significativamente la productividad de los equipos a la hora de desarrollar cualquier tipo de aplicación.

La versión 1 de Ionic ofrecía su SDK como un conjunto de directivas de AngularJS, así que esta versión 2 hace lo mismo pero con componentes, lo que mejora significativamente el rendimiento de su antecesor.

Un punto crítico a la hora de adoptar este tipo de tecnologías es la compatibilidad con los distintos dispositivos en sus distintas versiones y plataformas; sobre todo si tiene o no compatibilidad con dispositivos antiguos, por aquello de no dejar “colgados” a un número significante de usuarios.

En este punto os puedo decir, después de muchas pruebas empíricas, que este tipo de tecnologías funcionan perfectamente (sin hacer ninguna configuración adicional) en versiones iguales o superiores a la 4.4 de Android, a la 8.0 de IOS y en Windows 10.

En caso de necesitar compatibilidad con versiones 4.2 y 4.3 de Android, se puede añadir el módulo de Crosswalk a través de Cordova, que añade un navegador “moderno” a nuestra aplicación para que pueda ejecutarse en estas versiones antiguas. La única pega es que el tamaño de la aplicación aumenta considerablemente, hemos probado con una aplicación de 4 Mb, y el APK ha pasado a 25 Mb y una vez instalada ocupa 35 Mb; así que puede ser una buena solución si el espacio no es un problema y avisamos fehacientemente al usuario que descargue la aplicación a través de una red Wifi. Pero ya os digo esto solo afectaría a dispositivos 4.2 y 4.3 de Android. Aquí tenéis más información sobre el proyecto crosswalk.

Para Windows el equipo de Ionic considera que no merece la pena mantener compatibilidad con versiones anteriores a Windows 10. Esto viene motivado por la poca cuota de mercado y a la mejora drástica, en rendimiento y compatibilidad, del navegador en su versión 10.

En conclusión esta tecnología es muy valida siempre que se impongan al cliente ciertas restricciones en las versiones de los dispositivos soportados. Android 4.4+, IOS 8+ y Windows 10+ (Phone y Desktop).

3.2. HTML5

Usado para estructurar y presentar el contenido para la web. Es la quinta revisión del estándar que fue creado en 1990. La W3C la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos. También entra en desuso el formato XHTML, dado que ya no sería necesaria su implementación.

HTML5 es un sistema para formatear el layout de nuestras páginas, así como hacer algunos ajustes a su aspecto. Con HTML5, los navegadores pueden saber cómo mostrar una determinada página web, saber dónde están los elementos, dónde poner las imágenes, dónde ubicar el texto. En este sentido, el HTML5 no se diferencia demasiado de su predecesor, un lenguaje del cual hablamos hace algunos meses en nuestra guía básica de HTML. La diferencia principal, sin embargo, es el nivel de sofisticación del código que podremos construir usando HTML5.

3.3. CSS

Hojas de estilo en cascada o CSS, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.

Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web, y GUIs para muchas aplicaciones móviles.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

3.4. TypeScript

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases. Typescript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (NodeJS).

TypeScript extiende la sintaxis de JavaScript, por tanto cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original.

TypeScript soporta ficheros de definición que contengan información sobre los tipos de librerías JavaScript existentes, similares a los ficheros de cabeceras de C/C++ que describen

la estructura de ficheros de objetos existentes. Esto permite a otros programas usar los valores definidos en los ficheros como si fueran entidades TypeScript de tipado estático. Existen cabeceras para librerías populares como jQuery, MongoDB y D3.js, y los módulos básicos de NodeJS.

3.5. Node.js

NodeJS es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. NodeJS usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de NodeJS, npm, es el ecosistema mas grande de librerías de código abierto en el mundo.

La gran fortaleza de NodeJS es su entorno multiplataforma de código abierto para el desarrollo de aplicaciones de servidor y redes. Las siguientes características son las mas importantes a la hora de elegir para los arquitectos de software a NodeJS:

- Asíncrono y controlado por eventos: todas las API de su biblioteca son asíncronas sin bloqueo. Básicamente, significa que un servidor basado en NodeJS nunca espera que una API devuelva datos. El servidor pasa a la siguiente API después de llamarlo y un mecanismo de notificación de eventos de NodeJS ayuda al servidor a obtener una respuesta de la llamada API anterior.
- Altamente escalable: el mecanismo de eventos ayuda al servidor responder de forma no bloqueante y hace que el servidor sea altamente escalable en comparación con los servidores tradicionales que crean subprocesos limitados para manejar las solicitudes. NodeJS utiliza un único programa de subprocesos y el mismo programa puede proporcionar servicio a un número mucho mayor de solicitudes que los servidores tradicionales como Apache HTTP Server.
- Sin búfer: las aplicaciones nunca almacenan en búfer ningún dato. Estas aplicaciones simplemente generan los datos en fragmentos.
- Muy rápido: al estar creado en el motor de JavaScript V8 de Google Chrome.

¿Cómo sabemos lo bueno que es Node? Pues mirando quienes la usa: Uber, NETFLIX, Medium, Wikipedia, Product Hunt, Flipboard, Trello ... Pues esto es un indicador pues ellos que tienen de los mejores ingenieros han decidido usar esta tecnología y nos da un indirectamente que la tecnología funciona en ambiente a gran escala, porque al usar una tecnología nueva y esta no es escalable pues frustrará todo tu trabajo.

3.6. SQL-lite

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

3.7. Gráfica

De manera general, cuando tenemos una gran cantidad de datos ya sea como texto o como tablas, se prefiere tener toda la información lo más esquematizada posible y esto se consigue de distintas formas, un ejemplo de ello sería un cuadro de mandos, “Los cuadros de mando han de presentar sólo aquella información que sean imprescindible, de una forma sencilla y por supuesto, sinóptica y resumida.“ (Según la wikipedia: https://es.wikipedia.org/wiki/Cuadro_de_mando). Por ende una buena forma de presentar los datos es mediante gráficas, que dependiendo de cual puede ser muy intuitiva de cara al usuario.

Entre todas las librerías analizadas nos declinamos en usar Chart.js pues esta es una poderosa biblioteca de visualización de datos, pero sé que puede ser difícil comenzar y obtener un gráfico para mostrar. Pero, ¿Qué es Chart.js? es una biblioteca que visualiza datos mediante JavaScript, la cual es mantenida por la comunidad y esta disponible en GitHub. Es similar a otras bibliotecas como las de Google Charts. Podemos ver los tipos de gráficas que tienen en su web con la figura (9 y 10).

Para poder dibujar un gráfico con Chart.js debemos:

1. Descargar el paquete desde Github.
2. Definir donde ponerlo en la página web.
3. Elegir que tipo de gráfico.
4. Suministrar datos, etiquetas y más.

Y ya lo tendremos en nuestro proyecto. Como dato: estos gráficos son muy personalizables, como:

- Variable data:
 - labels: son las etiquetas que aparecen en la gráfica, un ejemplo sería para una gráfica de deporte: “Lunes“, “Martes“, “Miércoles“ y así para el resto de días de la semana.
 - datasets: son grupos de información que se mostraran en la gráfica.
 - ◊ label: es la etiqueta de dataset, es el significado de los datos de la gráfica. Continuando el ejemplo anterior: “Días de entrenamiento“.
 - ◊ data: es una lista que le dan valor a la gráfica y su longitud ha de ser igual al numero de labels.
 - ◊ backgroundColor: propiedad que sirve para elegir el color de fondo de la gráfica. Hay que darle color a cada labels de la gráfica.
 - ◊ borderColor: es similar al anterior pero con la diferencia de que es el color del borde de la gráfica.
 - ◊ borderEidth:

Para terminar con chartJS mostramos una imagen con el ejemplo de los definiciones antes mencionado, este lo podemos ver en la figura (11).

3.8. Google Maps

Google Maps es un servicio de mapas desarrollado por Google. Ofreciendo imágenes de satélite, mapas de calle, vistas panorámicas de calles de 360, condiciones de tráfico en tiempo real y

planificación de rutas para viajar ya sea a pie, en automóvil, en bicicleta o transporte público.

Google Maps se lanzó en febrero de 2005. La interfaz del servicio utiliza JavaScript, XML y Ajax. Google Maps ofrece una API que permite incrustar mapas en sitios web de terceros, y ofrece un localizador para empresas urbanas y otras organizaciones en numerosos países del mundo. Google Map Maker permitió a los usuarios expandir y actualizar colaborativamente el mapeo del servicio en todo el mundo, pero se suspendió a partir de marzo de 2017.

La API de Google Maps fue lanzada en junio de 2005, para permitir que los desarrolladores integren Google Maps en sus web. Siendo este gratis, y actualmente no contiene anuncios.

Google Maps API está disponible para muchas plataformas como: Android, iOS, navegadores web y a través de servicios web HTTP. En su página: <https://developers.google.com/maps/> nos encontramos con las distintas opciones según el entorno de trabajo en el que vamos a desarrollar:

- Android
 - Google Maps Android API
 - ◊ Agrega mapas a tu app de Android. Integra mapas básicos, edificios en 3D, planos interiores, imágenes de Street View y satelitales, marcadores personalizados y mucho más.
 - Google Places API for Android
 - ◊ Implementa la detección de sitios en el dispositivo y acciones de autocompletar, y agrega información sobre millones de ubicaciones a tu aplicación.
- iOS
 - Google Maps SDK for iOS
 - ◊ Agrega Google Maps a tu aplicación de iOS.
 - Google Places API for iOS
 - ◊ Implementa la detección de sitios en el dispositivo y acciones de autocompletar, y agrega información sobre millones de ubicaciones a tu aplicación.
- Web
 - Google Maps JavaScript API
 - ◊ Personaliza mapas con tus imágenes y tu contenido propios. Soporte de funciones sólido.
 - Google Static Maps API
 - ◊ Agrega un mapa de Google a tu sitio sin límites de escritura de código ni de cuota.
 - Google Street View Image API
 - ◊ Panoramas e imágenes reales.
 - Google Static Maps API
 - ◊ Incorpora una imagen de mapa sin carga de página dinámica ni JavaScript.
 - Biblioteca JavaScript de Google Places API
 - ◊ Información actualizada sobre millones de lugares.
- HTTP
 - Google Maps Android API

- ◊ Realiza conversiones de direcciones a coordenadas geográficas.
- Google Places API for Android
 - ◊ Implementa el autocompletado y agrega información actualizada sobre millones de ubicaciones a tu sitio o aplicación.
- Google Maps Elevation API
 - ◊ Datos de elevación para cualquier punto en el mundo.
- Google Maps Directions API
 - ◊ Calcula indicaciones entre varias ubicaciones.
- Google Maps Roads API
 - ◊ Habilita la funcionalidad snap-to-road para rastrear de forma precisa rutas de navegación de GPS.
- Google Maps Time Zone API
 - ◊ Proporciona datos de zonas horarias para cualquier lugar del mundo.
- Google Maps Geolocation API
 - ◊ Encuentra una ubicación según información de torres celulares y nodos WiFi.
- Google Maps Distance Matrix API
 - ◊ Calcula el tiempo de viaje y la distancia para varios destinos.

3.9. GPS

El Sistema de Posicionamiento es un conjunto de satélites que se usa para el conocimiento de la posición de un receptor de dichos satélites. Este sistema es de procedencia americano, y permite la localización en cualquier condición climatológica. Iniciado en los 70. Esta formado por un total de 32 satélites, de los cuales 24 están activos. En cuanto al número de usuarios a los que este sistema le da servicio es ilimitado debido a que es un sistema de recepción pasivo.

El conjunto de satélites están posicionados de tal manera que desde cualquier punto de la Tierra puedan ser visible de entre cuatro a ocho satélites y un ángulo mínimo sobre el horizonte de quince grados.

El segmento espacial, como hemos mencionado antes, está formado por una constelación de veinticuatro satélites a una altitud de veinte mil doscientos kilómetros y forman seis órbitas diferentes con cuatro satélites en cada una de ellas. Su periodo de rotación es de doce horas.

En cuanto al uso: cada satélite transmite, permanentemente, un mensaje de navegación indicando su posición orbital y la hora exacta de emisión del mensaje. También transmite información que permite a los receptores GPS poder localizar a los demás satélites. Dichos satélites van equipados con relojes atómicos, cuatro para ser exactos. La transmisión de cada uno de los satélite está sincronizado con el resto a través de los antes mencionados relojes. El receptor GPS necesita como mínimo cuatro satélites para calcular la ubicación 3D con precisión.

3.10. Angular JS

La consolidación del framework más demandado de JavaScript desarrollado por el gigante Google, Angular 2, ha sido clave en la creación de Ionic. Desde hace un tiempo venimos escuchando hablar mucho de aplicaciones web escritas con AngularJS. Muchos o la mayoría

ya habréis tenido contacto con este Framework o simplemente le habéis echado un ojo sin meteros en desarrollo con él. Los que lo estáis usando no podréis negar la velocidad de desarrollo que se adquiere en nuestros proyectos y la rápida curva de aprendizaje que tiene. No obstante, nunca está de más hablar de este Framework MVC para los que aun no lo conocen y como nos puede beneficiar a la hora de programar una aplicación con Phonegap.

AngularJS es un framework JavaScript creado y mantenido por Google que es utilizado para la creación de aplicaciones web siguiendo el patrón Modelo Vista Controlador, capacitado para extender nuestros documentos HTML añadiendo nuevas etiquetas y atributos que cumplirán con funciones específicas ya definidas o programadas por el desarrollador.

Ventajas de usar AngularJS:

- Con AngularJS tendrás tu código JavaScript organizado.
- Desarrollo más rápido.
- Es compatible con JQuery y otras librerías (con el uso veremos que con AngularJS es suficiente para manipular el DOM).
- Facilidad para hacer tests.

Desarrollando con AngularJS Según empezamos con AngularJS, aprenderemos diferentes conceptos como son módulos, controladores, directivas, expresiones, filtros, etc... que nos permitirán realizar muchas funcionalidades con un menor esfuerzo, así como organizar nuestro código y poder mantenerlo limpio y entendible. Veremos con qué facilidad se referencian los datos que tenemos en los modelos a las vistas, así como la organización del funcionamiento de cada vista con su propio controlador. Podremos separar nuestro código en módulos para facilitar su organización e incluso poder exportarlo a otros proyectos.

3.11. Conectividad internet

3.11.1. 1G

Allá por los setenta se dió origen a un nuevo estándar de conectividad que cambiaria al mundo. Esto permitió realizar llamadas con telefonos sin cable. Esta tecnología se basa en dividir el área de cobertura en celdas, estas celdas tienen una distribución hexagonal debido a que es la forma más optima. Se reutilizan las frecuencias, para optimizar recursos, de manera que la celdas contiguas a una misma no tenga la misma frecuencia, como en el juego sudoku. Gracias a esto al teléfono sin cable se le comenzó a denominar “celular”.

3.11.2. 2G

Con el crecimiento de esta tecnología, a principio de los noventa, se dio el siguiente salto de generación, que gracias al estandar GSM permitio ademas de realizar llamadas de voz, tener mayor velocidad de transferencias de datos y la posibilidad de enviar SMS. También los móviles más modernos tenían la posibilidad de acceder algunas páginas web adaptadas a estos dispositivos.

3.11.3. 3G

Antes de la llegada de esta generación, era impensable ver videos o sostener una videocofrenencia a través de un teléfono móvil. En esta tercera ola, nacen los primeros móviles inteligentes. Gracias al estándar UMTS se consigue hasta los 2 Mbps, también tiene soporte simultáneo de voz y datos. Hubo grandes cambios pues la llegada de los teléfonos inteligentes junto a la nueva tecnología 3G nos dio lugar a un cambio en la sociedad ya no sólo realizamos llamadas, también navegamos por internet, respondemos a nuestros correos, almacenamos nuestras fotos, etc.

3.11.4. 4G

La llegada de la cuarta generación fue una evolución natural de las tecnologías anteriores, que nos catapultó en cuanto a velocidad de transferencia de datos se refiere, llegando a los 3 Gbps en descarga y 1.5 en subida. En un uso normal se le da importancia a servicios en la nube.

3.12. Ionic

Ionic es un framework para el desarrollo de aplicaciones híbridas, su directriz inicial era pensado en los smartphones (formato móviles y tablets), pero también es posible su uso para implementar aplicaciones web. Su característica fundamental es que usa por debajo cordova y Angular 2, además de otras componentes, que facilitando así el desarrollo.

Pensada para obtener resultados de una manera rápida y con una menor inversión económica, ya que permite crear aplicaciones para distintas plataformas móviles con una misma base de código. Estas son:

Aquí va la imagen: plataformas de ionic

Hemos mencionado que Ionic desarrolla aplicaciones híbridas, pero ¿Qué es una aplicación híbrida?

Es la que nos permite desarrollar mediante las tecnologías web: HTML + CSS + Javascript. A la hora de instalarla el dispositivo no ve diferencias con una realizada a través de Android Studio o Xcode, que puedes descargar a través de las tiendas de cada sistema, por lo que en principio los usuarios finales no hallarán diferencia con respecto al resto de aplicaciones nativas.

Otra de sus grandes ventajas es que al ejecutarse con tecnologías web, los desarrolladores expertos en este medio pueden aprovechar dichos conocimientos teniendo una adaptación rápida en el desarrollo de apps para móviles.

Ejecutan en lo que se denomina un “web view”, que no es más que una especie de navegador integrado en el móvil y en el que solamente se ejecuta la app híbrida.

Las aplicaciones híbridas son interesantes por diversos motivos:

- Con una misma base de código serán capaces de compilar apps para funcionar correctamente en una gran cantidad de sistemas operativos de móviles o tablets. Generalmente nos será suficiente que nuestra app funcione en iOS y Android, pero Ionic es capaz de compilar a otros sistemas como Windows Phone.
- El coste del desarrollo es sensiblemente menor, ya que no es necesario contar con varios equipos de desarrollo para cada lenguaje concreto de cada plataforma.
- El tiempo de desarrollo también es menor, ya que solo es necesario construir la aplicación una vez e inmediatamente la tendremos en todas las plataformas a las que nos dirigimos.
- Es de más fácil adaptación para los desarrolladores que vienen de la web.

Para el desarrollo de una aplicación varía en función del entorno de desarrollo. Si queremos desarrollar en windows queda descartada la posibilidad de compilación de la app iOS. Pero podemos desarrollar en windows, para android y para windows phone, y posteriormente exportarlo a un macOS y desde este compilarlo en Xcode y obtener las tres plataformas más importantes.

Los requisitos para desarrollar en los distintos SO son:

- NodeJS
- Paquetes Cordova e Ionic
- SDK de Android / Xcode
- Java JDK (Android)
- Y sólo para macOS: apache-ant

Con ello ya estaremos listos para desarrollar nuestra aplicación. Para empezar a desarrollar es tan simple como ejecutar en el directorio que queremos el proyecto: ionic start nombreApp tipoPlantilla -v3

Como vemos en el comando lo primero que tenemos es start para indicarle a ionic que queremos iniciar un proyecto, el segundo parámetro es el nombre de la aplicación, tercero es el tipo de plantilla que veremos los distintos tipos que nos ofrece ionic y por último es la versión que queremos que se instale, las versiones van desde la uno, pasando por la dos y finalizando en la tres.

Esto permite iniciar de manera rápida un proyecto y con un diseño en la interfaz de nuestra app. Dicha interfaz se observa que guarda una disposición ordenada y limpia. La manera de usar los elementos como los iconos, la letra o la disposición de los mismos, varía dependiendo de la plataforma a utilizar, como hemos visto en las imágenes anteriores.

Retomando el tema de las versiones vemos diferencias en su estructura, de un simple vistazo vemos:

[AQUÍ VA LA IMAGEN: ESTRUCTURA EN PROYECTO IONIC](#)

Ahora vamos a describir los distintos

- hooks: Son scripts que se disparan durante el proceso de construcción. Generalmente se usan por comandos de Cordova para customización y la construcción de procesos automáticos.
- platforms: Aquí es donde se crean los proyectos de Android y de IOS. Podemos encontrarnos algunos problemas específicos de la plataforma durante el desarrollo que requieren estos archivos. Pero las dejaremos intactas durante la mayor parte del tiempo.

- plugins: Aquí se encuentran los módulos de Cordova. Cuando se crea una aplicación de Ionic, ya hay algunos de estos módulos instalados.
- resources: Esta carpeta se utiliza para añadir recursos como el icono y la pantalla de bienvenida.
- scss: Desde que el núcleo de Ionic está construido con Sass, ésta es la carpeta donde se encuentra el archivo.
- www: Es la carpeta donde trabajaremos de forma habitual y mantiene una estructura de carpetas en su interior por defecto, pero se pueden modificar según las necesidades del proyecto.

Aquí va la imagen: Lugar de trabajo

Aquí ya vemos el “lugar de trabajo” que empezaremos a describir los distintos directorios, de www, en la versión uno:

- css: aquí estarán nuestros estilos CSS.
- img: para las imágenes.
- js: contiene el fichero principal de configuración app.js, los componentes de AngularJS (controladores, servicios, directivas) y todos los archivos js.
- libs: aquí pondremos las librerías.
- templates: para los archivos html.
- index.html: el punto de inicio de la aplicación.

Por otro lado tenemos los directorios de src para la versión dos y tres:

- app: contiene el fichero principal de configuración app.js, los componentes de AngularJS (controladores, servicios, directivas) y todos los archivos js.
- assets: archivos de imágenes y otros de materiales externos que vas a usar en las páginas.
- pages: se alojarán las páginas que contenga nuestra aplicación.
- theme: directorio en el cual se puede hacer cambios en las variables generales de la interfaz de nuestra app.

Lo más notorio en estas dos estructuras es el directorio pages que nos ofrece las últimas versiones. Gracias a este cambio es mucho más intuitivo, en este directorio nos encontramos tres ficheros por cada vista que tenga la app, y son:

- .html: es la necesaria para poder realizar la vista.
- .css: que nos permite introducir código CSS que será usado por la extensión .html.
- .ts: que se trata del controlador.

En la versión uno tenías un sólo fichero TypeScript para todas las vistas, uno solo para todo el CSS y otro. Esto hace que la curva de aprendizaje sea positivamente acelerada. Y le digo de buena tinta, ya que he pasado por todas las versiones, que este pequeño cambio marca un antes y un después.

También tenemos dos comandos que son de utilidad para realizar la vista de nuestro proyecto y son:

- ionic serve
- ionic serve -lab

Esto nos permite visualizar en nuestro navegador web, como podemos ver en las figuras (13, 14 y 15), los cambios en la interfaz principalmente pero también el controlador. Gracias a esto cada vez que hagamos un cambio con nuestro editor de texto y lo guardemos automáticamente aparecerá el cambio en el navegador web. También por parte del explorador de Google, podemos ver dos elementos principales uno “element” y el otro “console”.

Gracias a “element” podemos ver el contenido html que contiene la vista que tenemos ante nosotros como en la figura (18).

En cuanto a la consola es una gran herramienta para los desarrolladores. Aquí encontrará errores y mensajes que puede poner allí para descubrir qué está ocurriendo en la aplicación. El comando de consola, de Ionic, más básico es “console.log('texto');”. En el entorno que vemos en la figura (19).

Como observamos en la figura (20) es el mejor ejemplo de como usar el comando “console.log” y una demostración de la potencia que tiene NodeJS. En este ejemplo se observa una respuesta en una comunicación mediante el protocolo HTTP.

En este ejemplo vemos que podemos usarlo para un testeo, a la hora de crear la aplicación y un servidor, pues se puede crear un servidor con NodeJS lo cual facilitaría muchas cosas.

Pero Ionic no acaba aquí con otra herramienta de testeo es el mismo la misma terminal, pues ella nos mostrará que paquetes o líneas de código son incorrectas, esto es ocurre cuando se introducen los comandos de Ionic.

Por otro lado Ionic ofrece una plataforma para realizar las aplicaciones, llamado Ionic Creator, de manera rápida y sencilla sin tener que escribir una línea de código. Las herramientas que nos aporta son las más destacadas de esta plataforma. Para añadir elementos a nuestras vistas tendremos que usar el llamado arrastrar y soltar, de manera intuitiva podremos ir construyendo nuestra aplicación. Dichas modificaciones se irán mostrando en la misma plataforma mediante un simulador de móvil.

Para empezar nos dirigiremos al portal (<https://creator.ionic.io/app/login?didUserLogOut=true>) e ingresaremos con nuestra cuenta de Ionic, en caso de no tener sólo tendremos que registrarnos, creamos un proyecto y nos dan a elegir entre los tres tipos de plantilla que tiene Ionic (en este ejemplo seleccionamos Tabs). Podemos echarle un vistazo a la figura (21).

Tras tener la plantilla elegida, Ionic permite añadirle algún estilo a la aplicación como: los colores de las cabeceras, menús, fuentes por defecto, entre otros. Pero si queremos podemos saltar este paso y seguir con la aplicación (figura 22).

Ahora observamos que en la parte izquierda observamos que se divide en dos categorías: páginas que nos permite añadir vistas a la aplicación y cambiar otros aspectos y los componentes que muestra las herramientas que podemos añadir (algunas de estas herramientas nos tenemos que hacer una cuenta “PRO” para usarlas). Una vez que arrastremos y soltemos

un elementos podemos editar sus características como el tamaño simplemente agrandando en su interfaz, situada en la columna central. Por último cuando seleccionamos un elemento podemos cambiar las características principales en la columna de la derecha de la plataforma. (figura 23).

Si queremos continuar con el proyecto y realizar algunos cambios de manera más precisa o simplemente queremos tener el proyecto en nuestro escritorio y de esa manera obtener de el APK para poder subirlo a la “Play Store“ sólo tenemos que exportarlo a través del apartado “Export“, como observamos en la figura (24).

Como hemos visto, Ionic cuenta con una gran cantidad de servicios que facilitan a la hora de realizar proyectos, pero esto no acaba aquí. Ionic pone a nuestra disposición una tienda (<https://market.ionicframework.com>) la cual se divide en tres grandes categorías:

- En primer lugar encontramos lo que Ionic denomina como “Starters“ que son aplicaciones creadas con ciertos componentes, diseños y módulos que permiten al usuario empezar con una base en su proyecto.
- También permite descargar temas para nuestro proyecto. Aquí nos encontramos con todo tipo de temas e incluso con módulos ya instalados y así hacerlo todo más rápido y sencillo.
- Por último tenemos a los módulos, estos nos permite solucionar problemas o realizar ciertas partes de nuestro proyecto de manera más rápida, por ejemplo algo como un calendario, que desde cero nos llevaría semanas gracias a esto tardaríamos minutos.

Estas categorías se alimentan por otros desarrolladores y usuarios de Ionic. Los productos de las distintas categorías nos los encontramos con un precio, pero también los hay gratuitos.

3.12.1. Api Google Maps

En Ionic tenemos la versión 2 del módulo de google maps. Para poder integrarla primero necesitamos tener una cuenta de Google para desarrolladores y generar API KEY de la aplicación. Esto es obligatorio para poder utilizar la API, como más adelante veremos. Al obtener y configurar la API KEY para el sistema operativo al que queremos utilizar Google Maps.

Para poder instalar este módulo introduciremos estos dos comandos:

- \$ ionic cordova plugin add cordova-plugin-googlemaps
 - -variable API_KEY_FOR_ANDROID = “ANDROID_API_KEY_AQUI“
 - -variable API_KEY_FOR_IOS = “IOS_API_KEY_AQUI“
- \$ npm install –save @ionic-native/google-maps

Y donde aparece las API KEY les pondremos las correspondientes a su sistema operativo. Con estas dos instrucciones ya tenemos la configuración necesaria para que el proyecto tenga Google Maps. Ahora que ya lo tenemos en nuestras librerías tenemos que declararlo en el fichero “app.module.ts“, como podemos ver en la figura (25), esto es necesario para que sea cargado el módulo en la aplicación.

El siguiente paso es introducirlo en la vista que queremos usarlo para ello tendremos primero que añadirlo en el fichero typeScript: “entrenamiento.ts“, de la manera que se muestra en la figura (26), con lo que nos permitirá usar el método: “Spherical.computeDistanceBetween(latLng-A,latLng-B)“ con lo que devolverá la distancia entre los puntos A y B en metros. Con la ayuda del método “LatLng(latitud,longitud)“, donde las latitudes y longitudes las obtendremos del GPS del dispositivo.

3.13. Conectividad internet en Ionic

En cuanto a la conectividad a internet, su uso es muy sencillo pues con un simple comando como el siguiente podemos realizar una petición HTTP. También se puede realizar desde el fichero de TypeScript de Ionic, desde aquí habrá que descargar un módulo de cordova para poder establecer una comunicación.

Por último para cerrar todo con lo referente a Google Maps que complementa a todo lo anterior, es el uso de servicios web, más concretamente es un servicio HTTP. Como vemos en esta línea de código, de HTML, que mostramos a continuación:

```
1 
```

Como podemos observar es una etiqueta HTML, de imagen, esta realiza una petición al servidor de Google que devolverá una imagen de Google Maps, con un tamaño definido, la cual se le trazaría una línea que indicaría el camino recorrido, dado la situación GPS. Ahora expandiremos la parte más destacable:

- La primera línea es una línea que comunica con el servidor de google.
- La segunda línea permite elegir el zoom que se quiere hacer sobre el mapa.
- En la tercera es una línea que se dibuja en el mapa y que elegiremos su color en función de las posiciones GPS.
- La cuarta línea es para el tamaño/grosor de la trazo que se va a dibujar.
- Por último tenemos “GPS“ que es una variable la cual obtiene de la base de datos todas las posiciones.

3.14. Api GPS

Para completar el módulo Google maps para Ionic, hemos de implementar el módulo GPS. Este complemento proporciona información sobre la ubicación del dispositivo, como la latitud y la longitud. Las fuentes comunes de información de ubicación incluyen el Sistema de Posicionamiento Global (GPS) y la ubicación deducida de las señales de red como la dirección IP, RFID, WiFi y las direcciones MAC Bluetooth, y las ID de las células GSM/CDMA.

Para poder instalar este módulo introduciremos estos dos comandos:

- \$ ionic cordova plugin add cordova-plugin-geolocation
- -variable GEOLOCATION *SAGE DESCRIPTION* “Localizame”\$npm install --save@ionic-native/geolocation

Cómo ocurría en el módulo de Google maps, con ese comando ya lo tenemos en nuestras librerías y ahora tenemos que declararlo en el fichero “app.module.ts”, como podemos ver en la figura (25), esto es necesario para que sea cargado el módulo en la aplicación.

El siguiente paso es exactamente igual al anterior módulo mencionado. En la vista que queremos usarlo tendremos que declararlo, para ello tendremos primero que añadirlo en el fichero typeScript: “entrenamiento.ts”, de la manera que se muestra en la figura (26), con lo que nos permitirá usar el método: “Spherical.computeDistanceBetween(latLng-A,latLng-B)” con lo que devolverá la distancia entre los puntos A y B en metros. Con la ayuda del método “LatLng(latitud,longitud)”, donde las latitudes y longitudes las obtendremos del GPS del dispositivo.

```
1 this.geo.getCurrentPosition().then((res) => {  
2   this.listPosicion= this.listPosicion+res.coords.latitude+', '+res.coords.longitude  
3 } )
```

4. Análisis y diseños

4.1. Casos de usos

Aquí va la imagen: Caso de uso

4.2. Escenarios

4.3. Diseño de objetos

4.4. Programar

5. Medios Materiales

- Herramientas Hardware:
 - PCs impreso
 - Impresora
- Herramientas Software:
 - Sistema operativo ESPECIFICA
 - Herramientas de edición y de desarrollo de programas ESPECIFICA

6. Bibliografía

- ARTICULO1

- ARTICULO2
- ARTICULO3

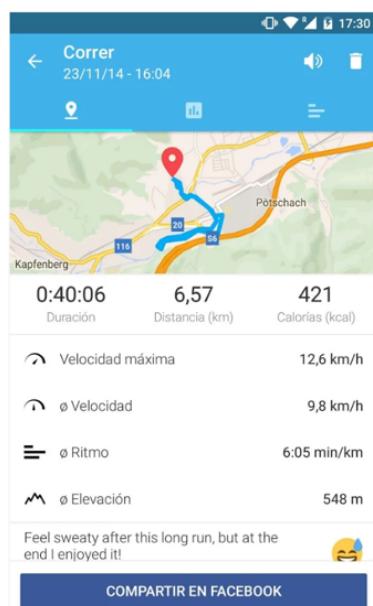


Figura 5: Múltiples imágenes de fitfapp

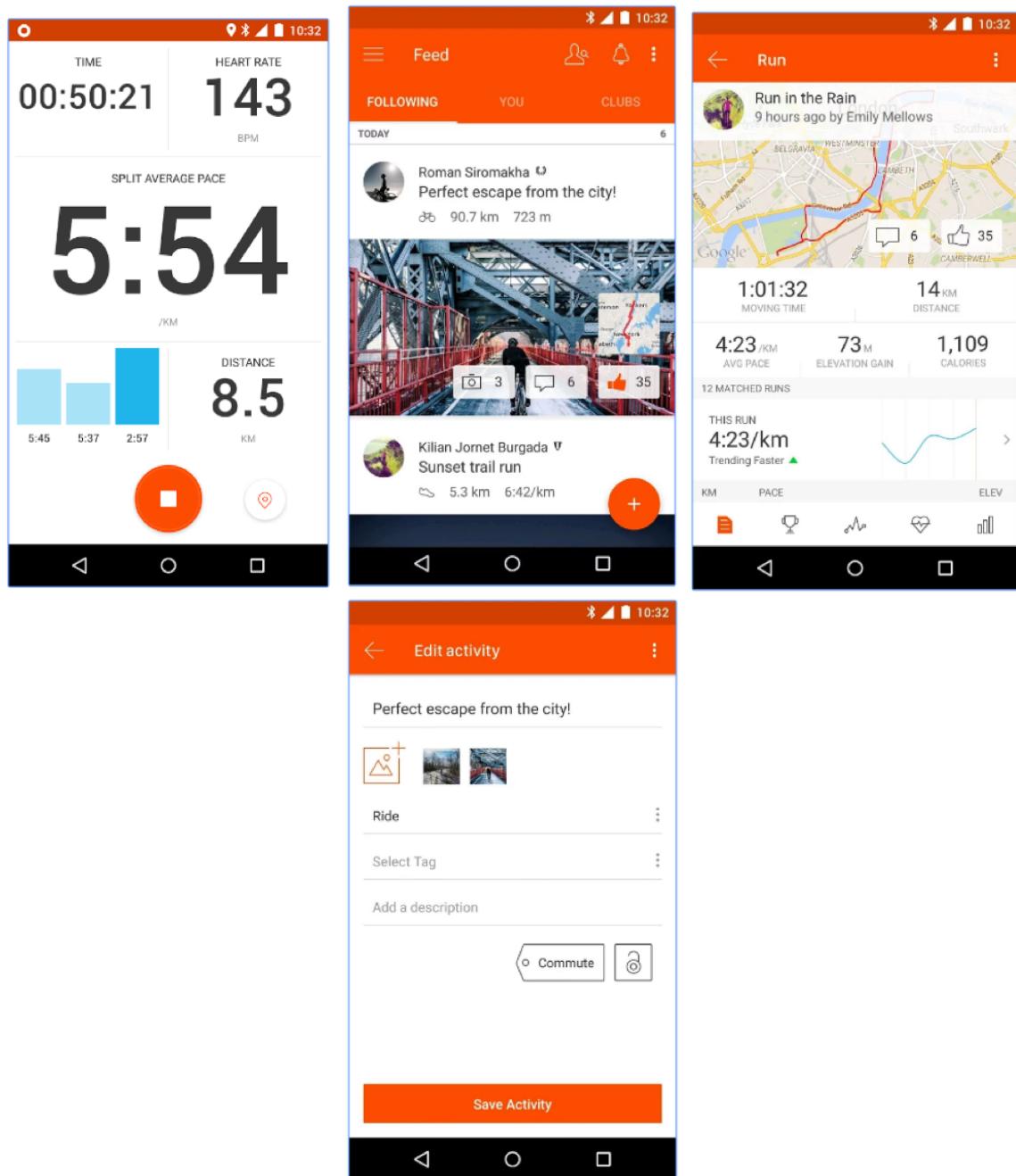


Figura 6: Múltiples imágenes de Strava

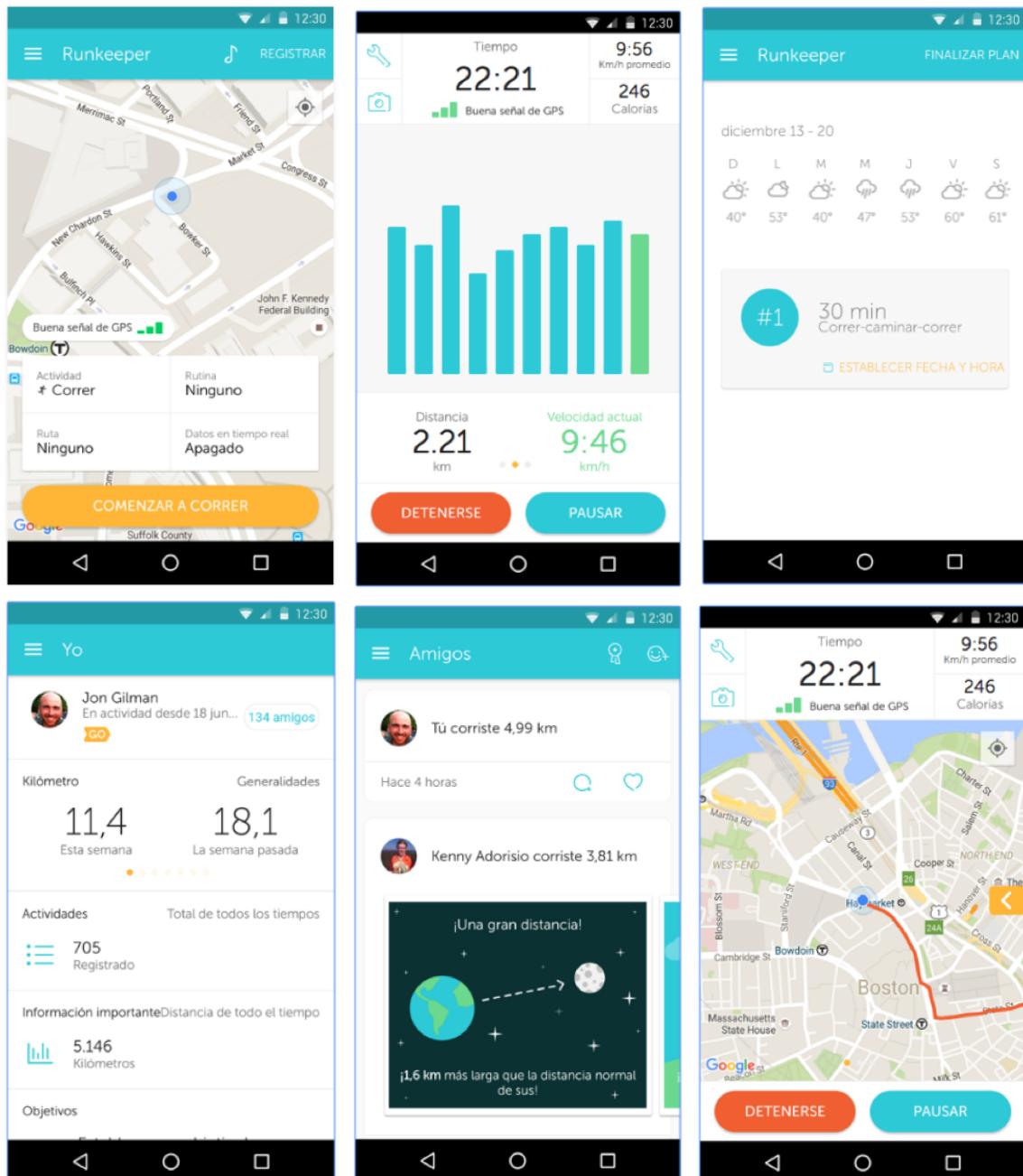


Figura 7: Múltiples imágenes de Runkeeper

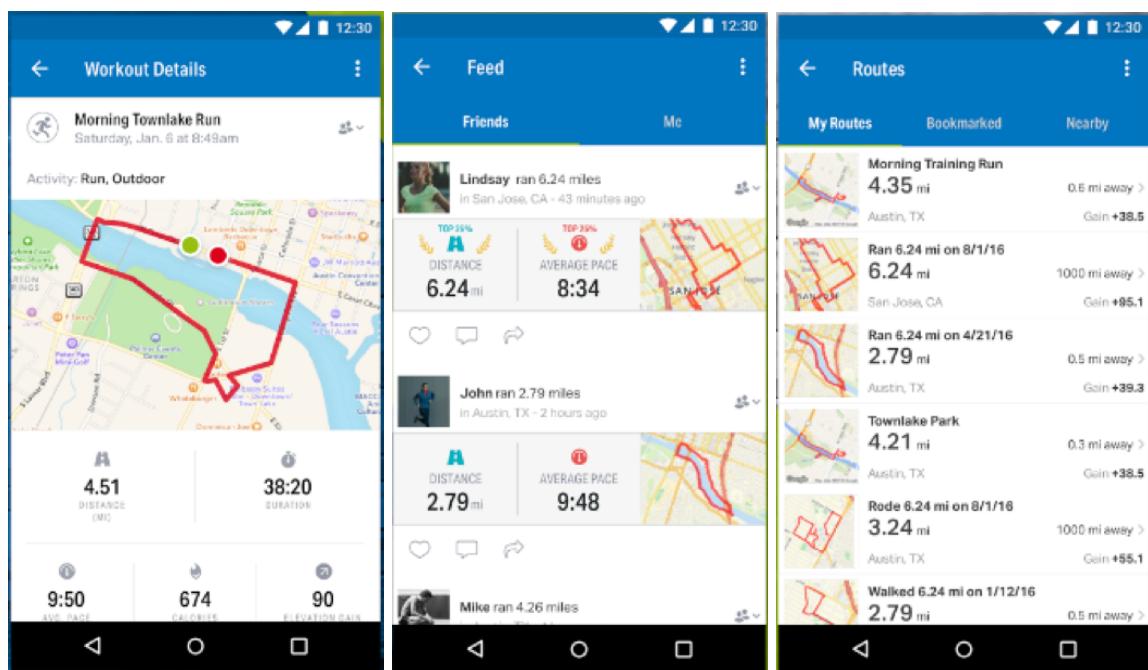


Figura 8: Múltiples imágenes de MapMy

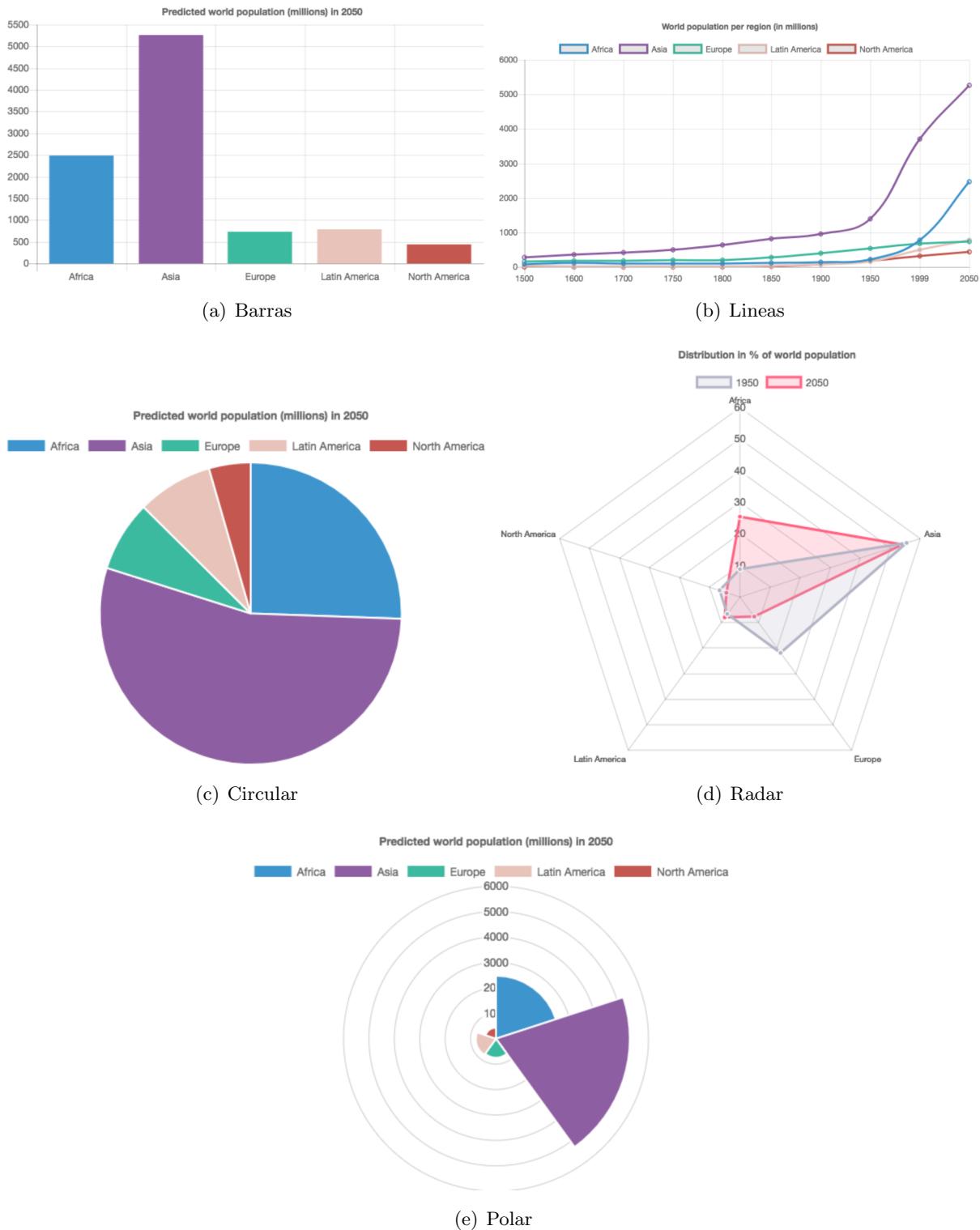


Figura 9: Graficas Chart.js (I)

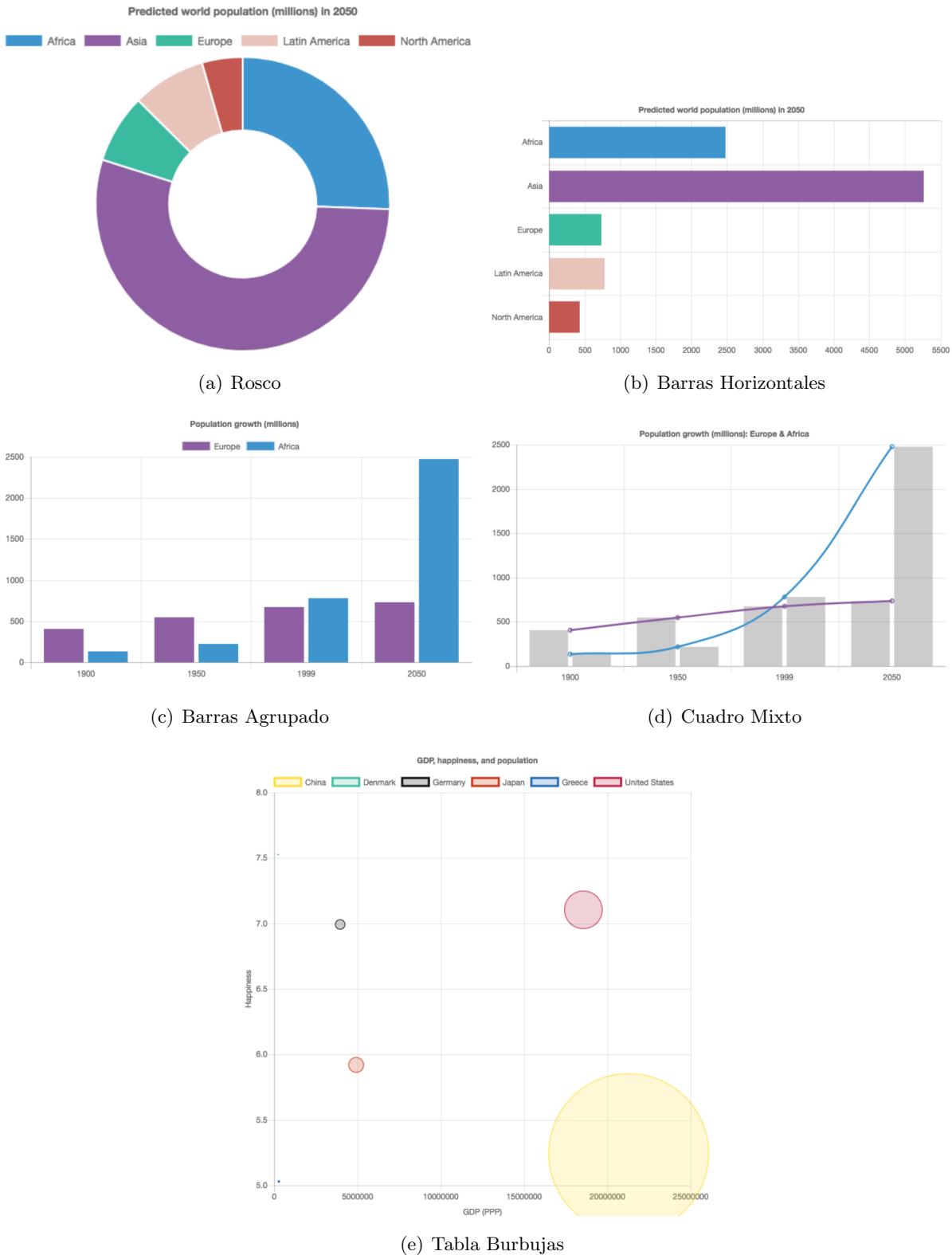


Figura 10: Gráficas Chart.js (II)

```

var data = {
  labels: ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"],
  datasets: [
    {
      label: 'Días de entrenamiento',
      data: [20, 50, 10, 35, 42, 5, 0],
      backgroundColor: [
        'rgba(100, 100, 100, 0.25)',
        'rgba(100, 100, 100, 0.25)',
        'rgba(100, 100, 100, 0.25)'
      ],
      borderColor: [
        'rgba(200,200,200,1)',
        'rgba(200,200,200,1)',
        'rgba(200,200,200,1)'
      ],
      borderWidth: 2
    }
  ];
}

```

Figura 11: Ejemplo de entrenamiento

	MAC	LINUX	WINDOWS
<i>iOS</i>	✓		
<i>Amazon Fire OS</i>	✓	✓	✓
<i>Android</i>	✓	✓	✓
<i>BlackBerry 10</i>	✓	✓	✓
<i>Navegador</i>	✓	✓	✓
<i>Ubuntu</i>	✓	✓	
<i>Windows Phone 8</i>			✓
<i>Windows</i>			✓
<i>Firefox OS</i>	✓	✓	✓

Figura 12: Plataformas de Ionic

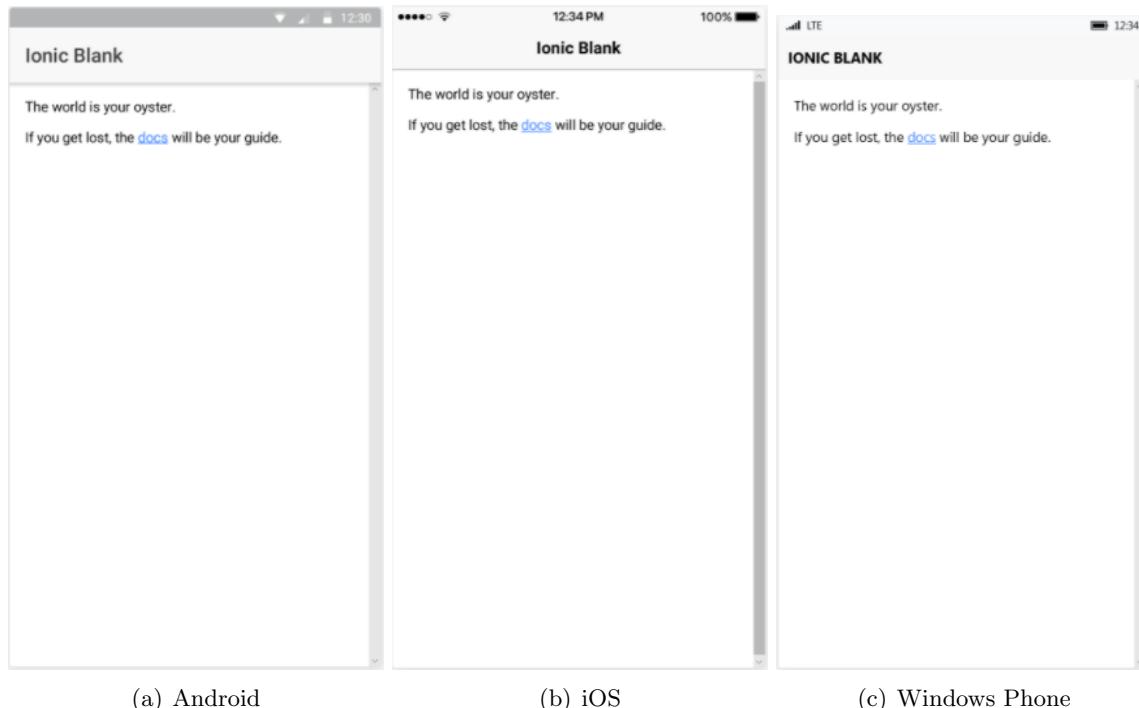


Figura 13: Tipo de plantilla: blank

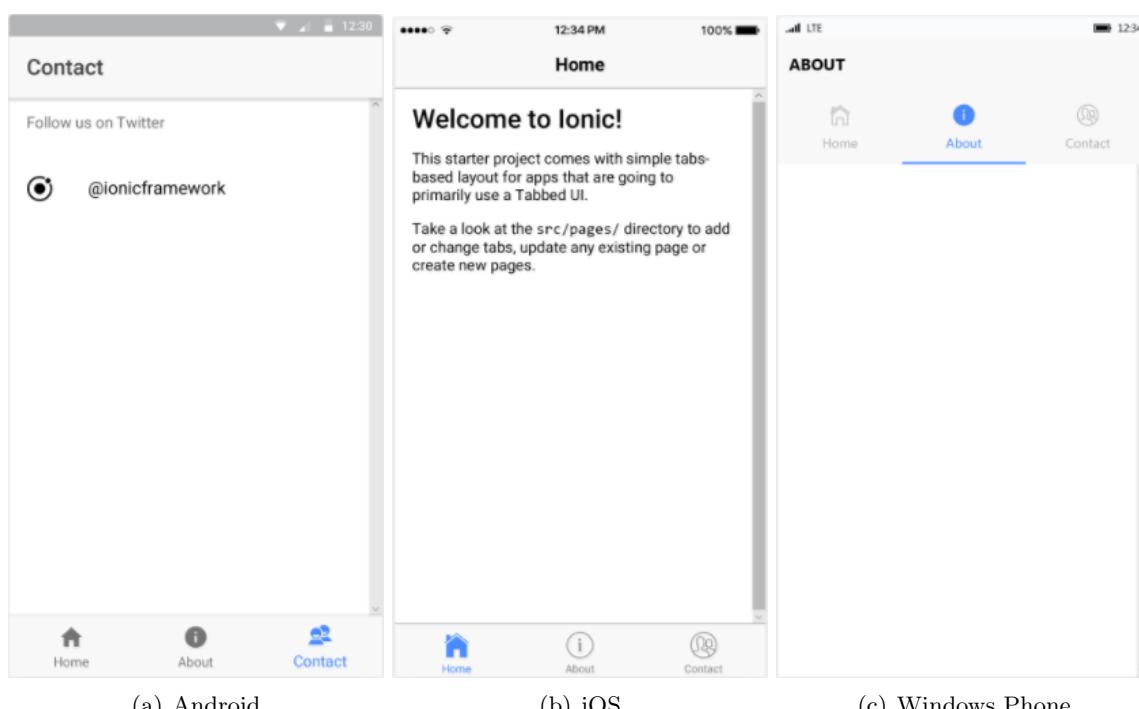
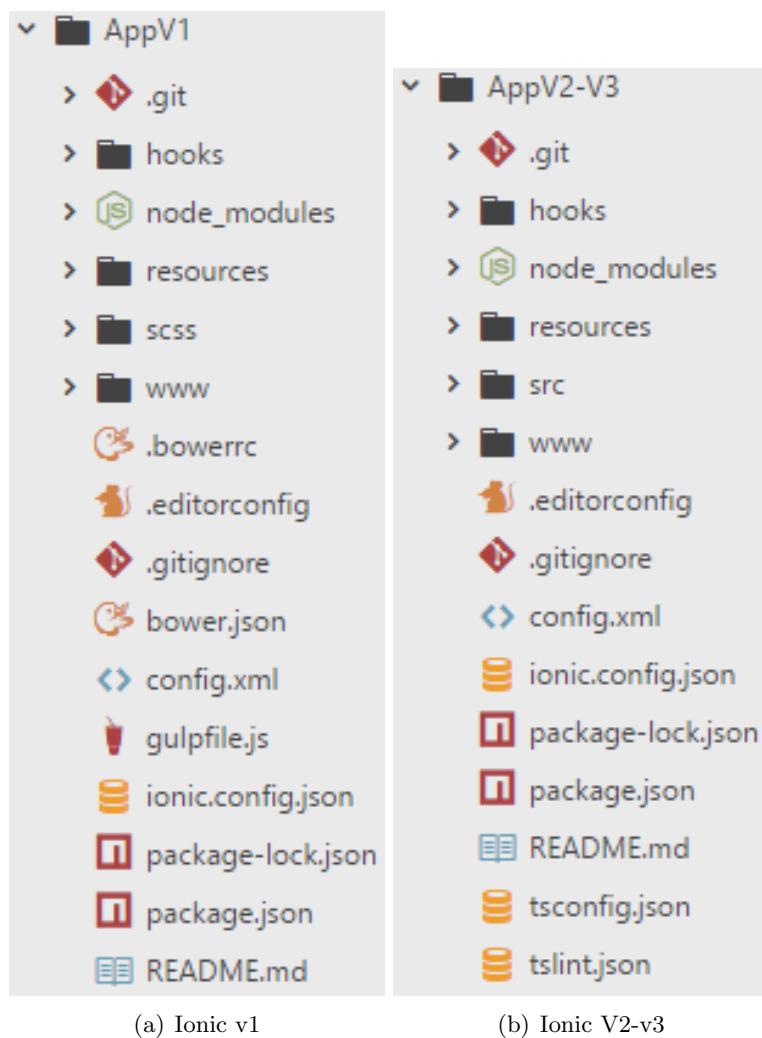


Figura 14: Tipo de plantilla: tabs



Figura 15: Tipo de plantilla: sidemenu



(a) Ionic v1

(b) Ionic V2-v3

Figura 16: Estructura en proyecto Ionic

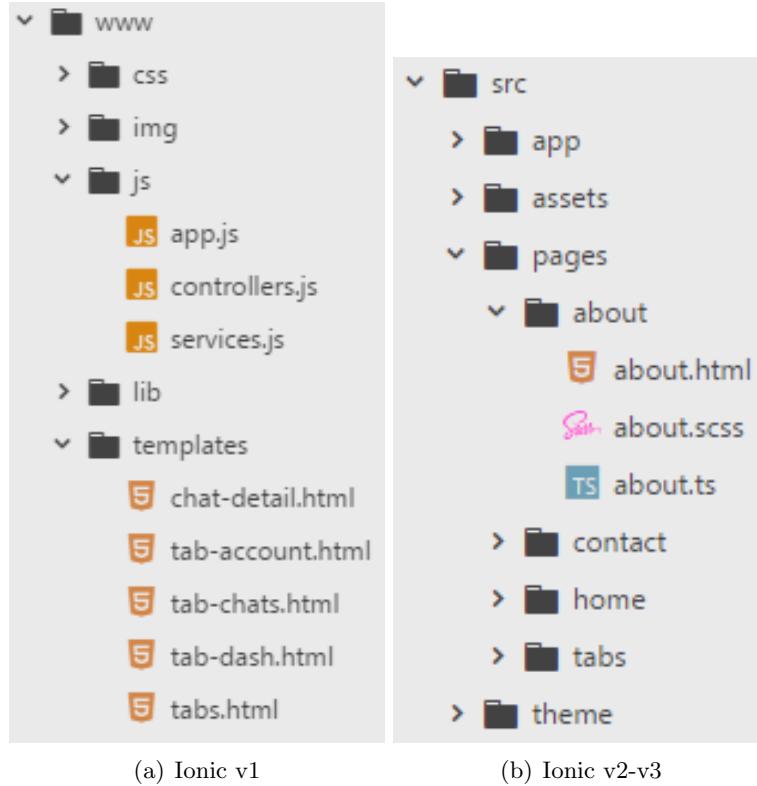


Figura 17: Lugar de trabajo

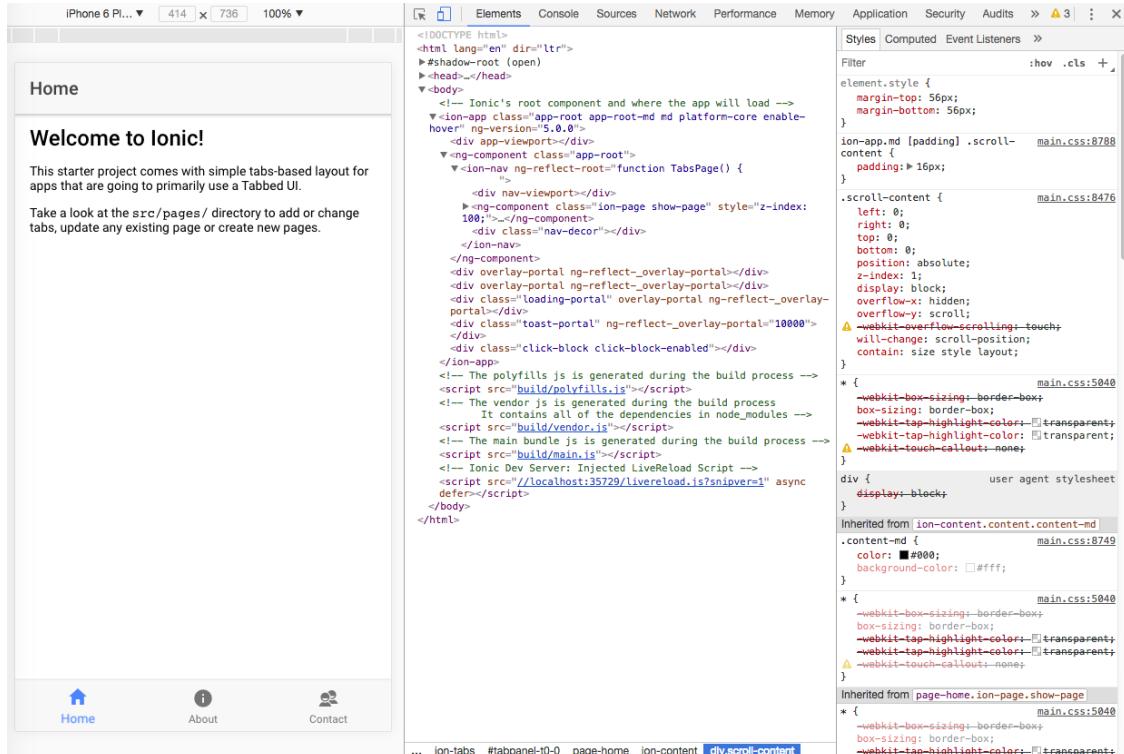


Figura 18: Ionic serve - Elements

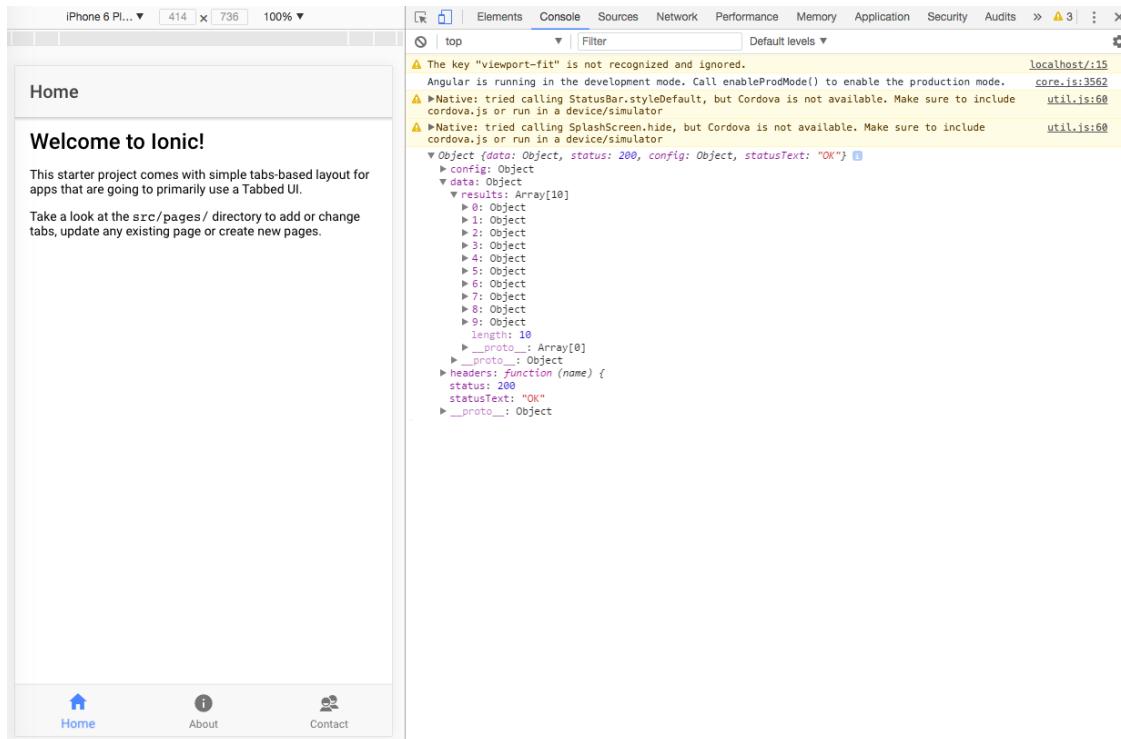


Figura 19: Ionic serve - Console

```

▼ Object {data: Object, status: 200, config: Object, statusText: "OK"} ⓘ
  ► config: Object
  ▼ data: Object
    ▼ results: Array[10]
      ► 0: Object
      ► 1: Object
      ► 2: Object
      ► 3: Object
      ► 4: Object
      ► 5: Object
      ► 6: Object
      ► 7: Object
      ► 8: Object
      ► 9: Object
      length: 10
      ► __proto__: Array[0]
    ► __proto__: Object
  ► headers: function (name) {
    status: 200
    statusText: "OK"
    ► __proto__: Object
  > |
```

Figura 20: Objects

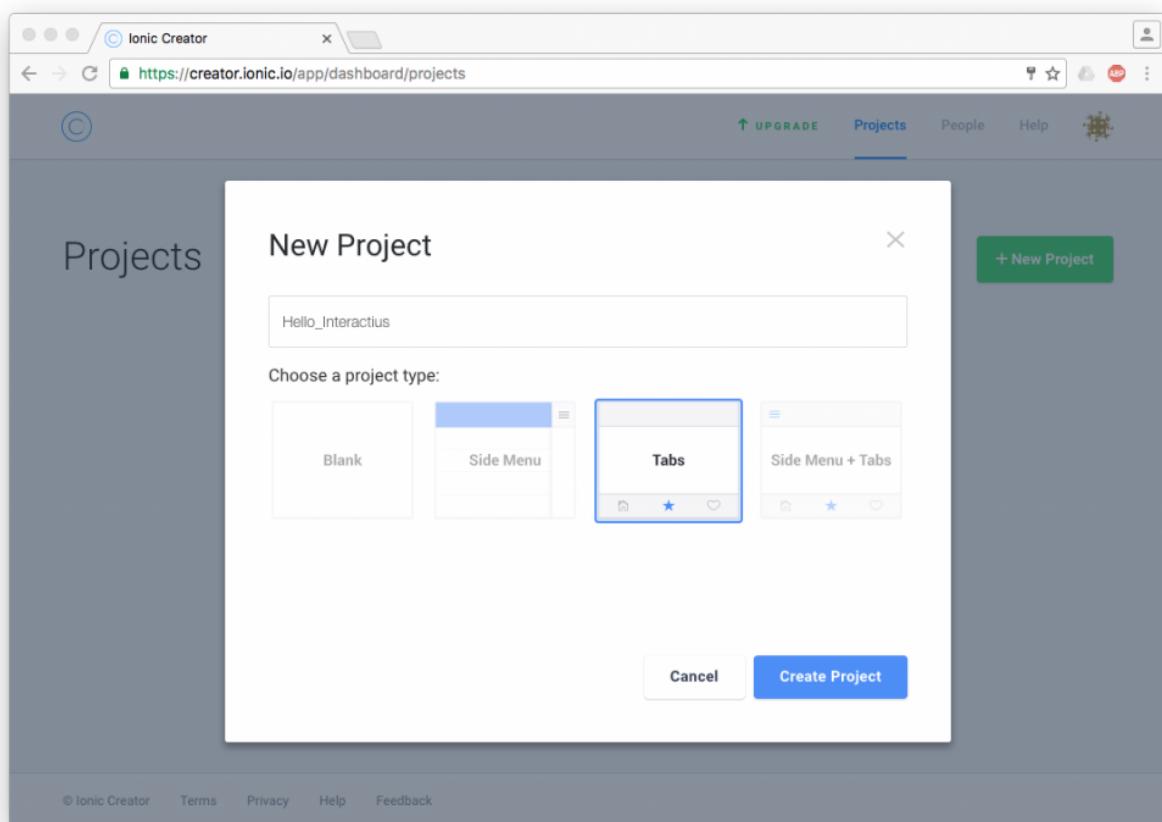


Figura 21: Nuevo proyecto en Ionic Creator

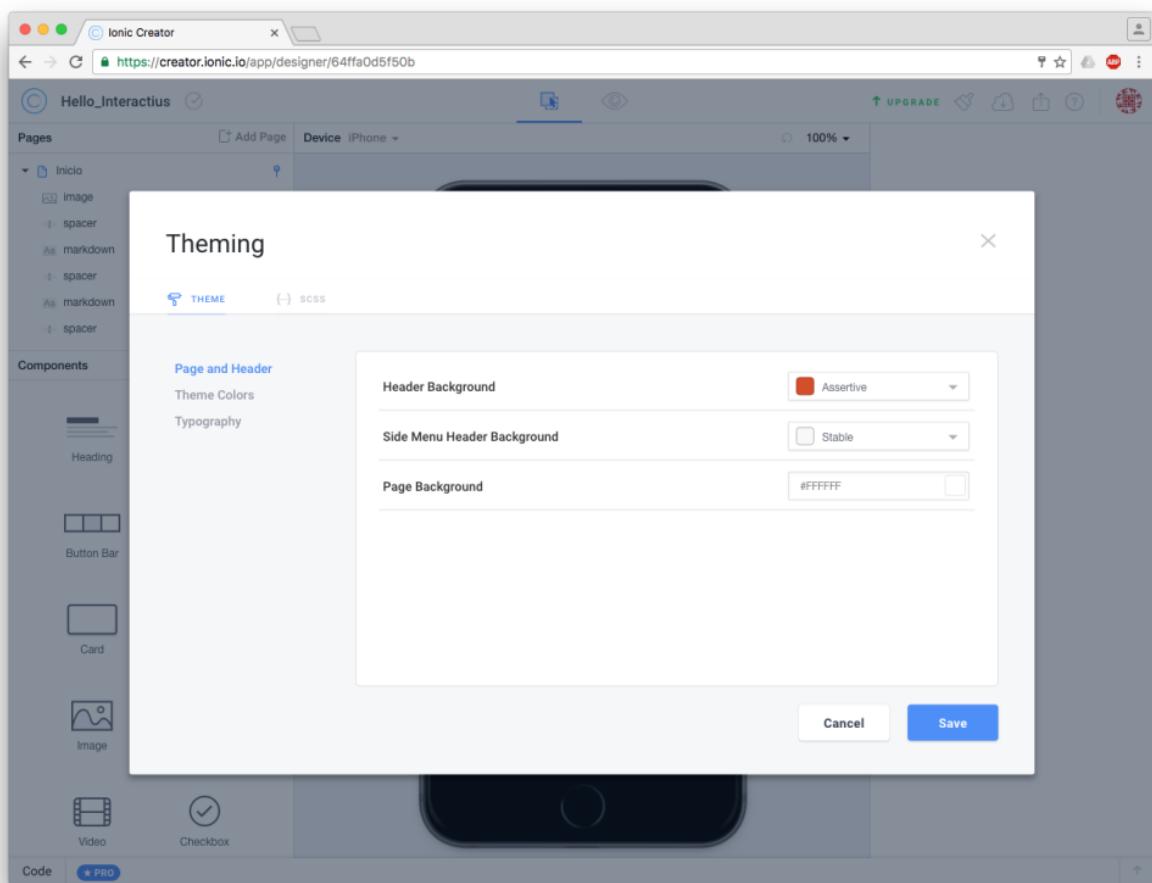


Figura 22: Cambios en el tema en Ionic Creator

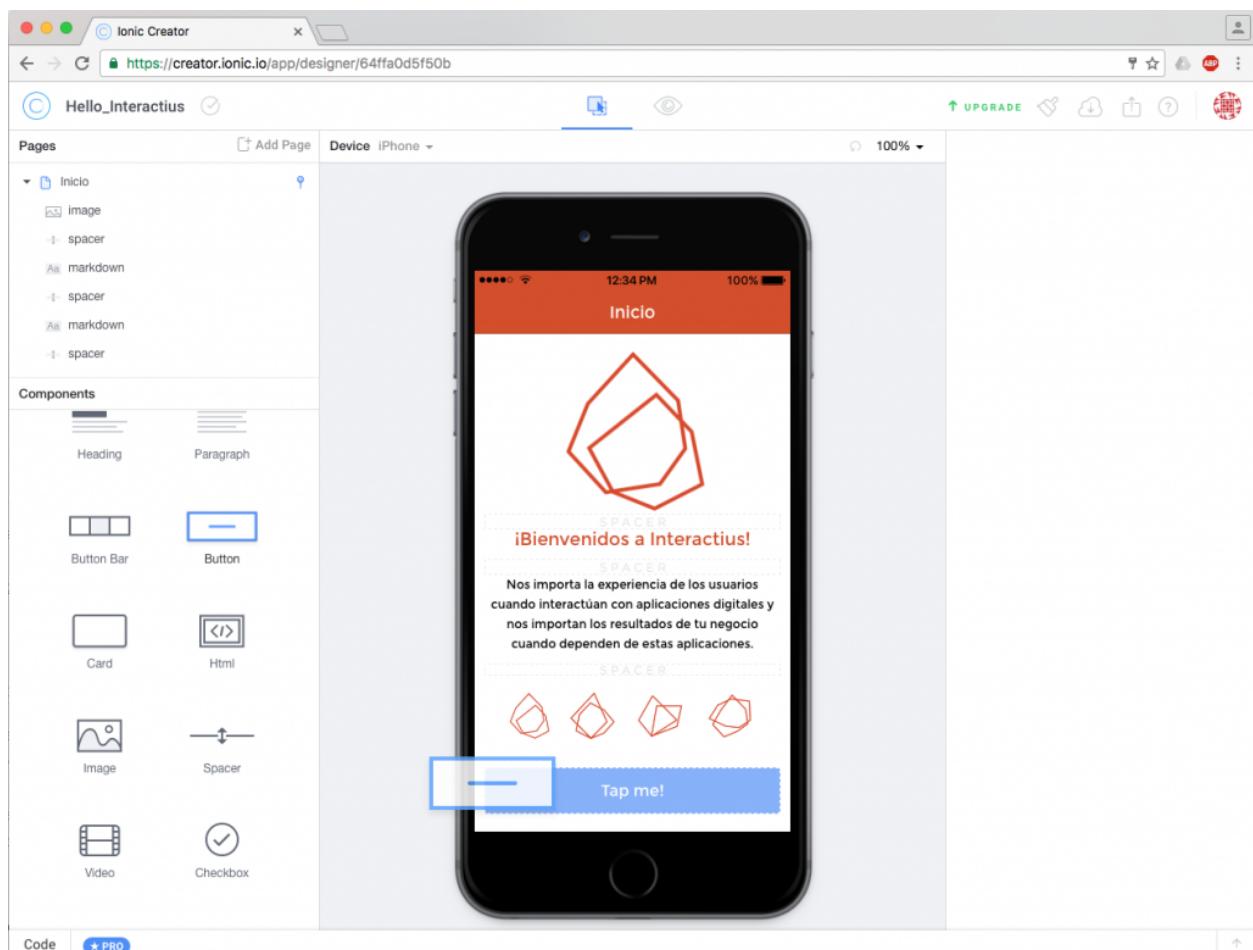


Figura 23: Interfaz de Ionic Creator

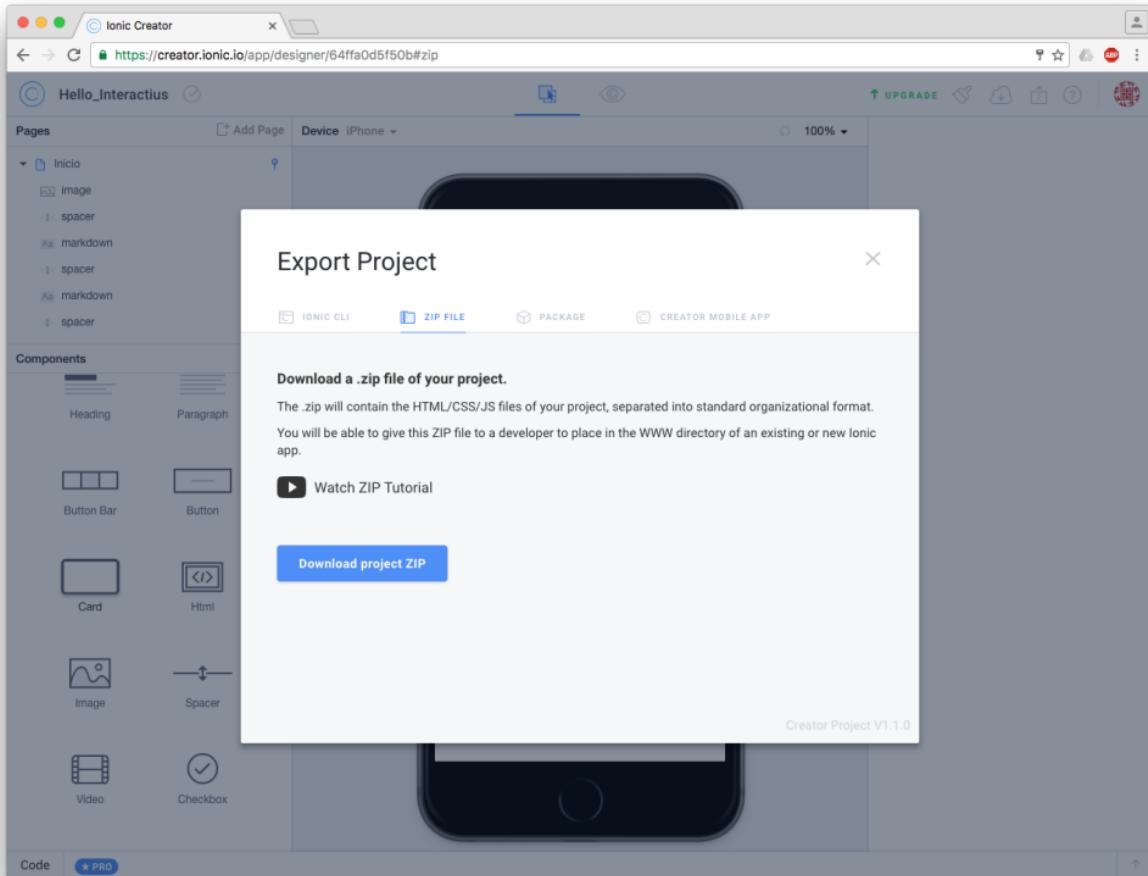


Figura 24: Exportar mi proyecto en Ionic Creator

```

src
  app
    app.component.ts
    app.html
    app.module.ts
    app.scss
    main.ts
  assets
  pages
  providers
  theme
  index.html
  manifest.json
  service-worker.js
23   import { GoogleMaps } from '@ionic-native/google-maps';
24   import { Geolocation } from '@ionic-native/geolocation';
25
26   @NgModule({
27     declarations: [
28       ...
29     ],
30     imports: [
31       ...
32     ],
33     bootstrap: [IonicApp],
34     entryComponents: [
35       ...
36     ],
37     providers: [
38       StatusBar,
39       SplashScreen,
40       {provide: ErrorHandler, useClass: IonicErrorHandler},
41       DbProvider,
42       SQLite,
43       Geolocation,
44       GoogleMaps
45     ]
46   })
47   export class AppModule {}
48
49
50
51
52
53
54
55
56
57

```

Figura 25: fichero: src/app/app.module.ts

```

1 import { Component } from '@angular/core';
2 import { NavController, AlertController } from 'ionic-angular';
3 import { Geolocation } from '@ionic-native/geolocation';
4 import { LatLng, Spherical } from '@ionic-native/google-maps';
5 import { DbProvider } from '../../../../../providers/db/db';
6 @Component({
7   selector: 'page-entrenamiento',
8   templateUrl: 'entrenamiento.html'
9 })
10 export class EntrenamientoPage {
11

```

Figura 26: fichero: src/pages/entrenamiento/entrenamiento.ts

Caso de uso: Iniciar una carrera

<i>Resumen</i>	El usuario ha de activar la ubicación GPS para poder realizar la carrera, posteriormente comenzará la carrera.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario debe tener activado la ubicación GPS. Y situarse en la vista “Entrenamiento”.
<i>Postcondición</i>	El usuario obtendrá todos los datos correspondientes al ejercicio realizado. <ul style="list-style-type: none"> 1. El usuario accede a la aplicación y accede a la vista “Entrenamiento” 2. EL usuario dará al botón inicio para empezar la carrera. 3. El cronómetro y el seguimiento GPS comenzarán. 4. La API de Google Maps irá calculando la distancia.
<i>Curso Normal</i>	<ul style="list-style-type: none"> 1. El usuario pulsa el botón iniciar carrera. 2. Aparece un mensaje de error: “no encendido GPS”.
<i>Curso Alternativo</i>	
<i>Observaciones</i>	Sin observaciones

Caso de uso: Parar una carrera

<i>Resumen</i>	El usuario detiene la carrera pulsando un botón.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario ha iniciado una carrera. Y situarse en la vista “Entrenamiento”.
<i>Postcondición</i>	El usuario detendrá la carrera. <ul style="list-style-type: none"> 1. El usuario accede a la aplicación y accede a la vista “Entrenamiento” 2. EL usuario dará al botón pausa. 3. El cronómetro, el seguimiento GPS y la API de Google Maps se detendrá.
<i>Curso Normal</i>	
<i>Curso Alternativo</i>	No existe curso alternativo.
<i>Observaciones</i>	El usuario detendrá la carrera y con ello no afectara a su marca el tiempo que este en pausa.

Caso de uso: Reanudar una carrera

<i>Resumen</i>	El usuario continua la carrera pulsando un botón.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario ha pausado una carrera. Y situarse en la vista "Entrenamiento".
<i>Postcondición</i>	El usuario detendrá la carrera.
<i>Curso Normal</i>	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y accede a la vista "Entrenamiento" 2. EL usuario dará al botón inicio. 3. El cronometro, el seguimiento GPS y la API de Google Maps continuarán tomando datos.
<i>Curso Alternativo</i>	No existe curso alternativo.
<i>Observaciones</i>	El usuario detendrá la carrera y con ello no afectara a su marca el tiempo que este en pausa.

Caso de uso: Finalizar una carrera

<i>Resumen</i>	El usuario decide que ha terminado su ejercicio físico y lo finaliza.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario ha iniciado una carrera. Y situarse en la vista "Entrenamiento".
<i>Postcondición</i>	El usuario finalizará la actividad física.
<i>Curso Normal</i>	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y accede a la vista "Entrenamiento" 2. EL usuario dará al botón finalizar. 3. Todos los procesos de seguimiento del ejercicio finalizaran. 4. Saltara una ventana flotante.
<i>Curso Alternativo</i>	No existe curso alternativo.
<i>Observaciones</i>	El usuario ha de estar seguro de que finalizara la carrera.

Caso de uso: Guardar una carrera

<i>Resumen</i>	El usuario decide que ha terminado su ejercicio físico y lo finaliza.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario ha finalizado una carrera. Y aparecerá una ventana emergente.
<i>Postcondición</i>	El usuario guardara la actividad física.
<i>Curso Normal</i>	<ol style="list-style-type: none"> 1. El usuario pulsara no para que se guarde. 2. El sistema guardara los datos de la carrera.
<i>Curso Alternativo</i>	El usuario pulsa no.
<i>Observaciones</i>	Sin observaciones.

Caso de uso: *No guardar una carrera*

<i>Resumen</i>	El usuario decide que ha terminado su ejercicio físico y lo finaliza.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario ha finalizado una carrera. Y aparecerá una ventana emergente.
<i>Postcondición</i>	El usuario no guardara la actividad física.
<i>Curso Normal</i>	<ol style="list-style-type: none"> 1. El usuario pulsara no para que se guarde. 2. El sistema guardara los datos de la carrera.
<i>Curso Alternativo</i>	El usuario pulsa si.
<i>Observaciones</i>	Sin observaciones.

Caso de uso: *Visualizar una carrera*

<i>Resumen</i>	El usuario podrá ver la carrera que ha realizado en una fecha concreta.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario ha de tener almacenado al menos una carrera.
<i>Postcondición</i>	El usuario podrá visualizar cualquier dato almacenado en la BD.
<i>Curso Normal</i>	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y en la vista “Historial”. 2. El usuario seleccionará la fecha que desea revisar. 3. El usuario seleccionará una hora de las aparecidas, estas hacen referencia a las carreras realizadas en dicha fecha. 4. Seleccionará la carrera que quiera. 5. Cambiará la vista y mostrará los datos de la carrera.
<i>Curso Alternativo</i>	No existe curso alternativo.
<i>Observaciones</i>	Sin observaciones

Caso de uso: *Visualizar progreso*

<i>Resumen</i>	El usuario podrá ver el progreso que ha realizado desde que empezó a realizar una actividad física.
<i>Actores</i>	Usuario
<i>Precondición</i>	El usuario ha de tener almacenado al menos una carrera.
<i>Postcondición</i>	El usuario podrá visualizar cualquier dato almacenado en la BD.
<i>Curso Normal</i>	<ol style="list-style-type: none"> 1. El usuario accede a la aplicación y en la vista “Estadística”. 2. Aparecerá una gráfica donde mostrará el avance.
<i>Curso Alternativo</i>	No existe curso alternativo.
<i>Observaciones</i>	Sin observaciones