



Aplicación para gestionar el entrenamiento de natación continua. SwimNCoach.

Manuel García Dolz

Máster Universitario de Desarrollo de Aplicaciones para Dispositivos Móviles

Tutor: Eduardo Martín Lineros

Fecha de entrega: 06/06/2018

© Manuel García Dolz

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	Aplicación para gestionar el entrenamiento de natación continua. SwinNCoach.
Nombre del autor:	Manuel García Dolz
Nombre del consultor:	Eduardo Martin Lineros
Fecha de entrega (mm/aaaa):	06/2018
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>
Resumen del Trabajo:	
<p>En la práctica de la natación continua como deporte se utiliza la velocidad y el ritmo como métricas para saber el desempeño que ha tenido un nadador en un cierto recorrido. Estas métricas se tienen en consideración para preparar pruebas deportivas y para mejorar de manera individual. Actualmente, el procedimiento más habitual de los entrenadores para gestionar las sesiones de entrenamiento con sus alumnos consiste en registrar los tiempos con un cronómetro y anotarlos de manera manual. Este proceso resulta tedioso y generalmente es complicado mantener una estructura organizada de los datos y obtener una visión global de la evolución de los alumnos a lo largo del tiempo. Para hacer más sencilla y eficiente la tarea de estos entrenadores se desarrolla en este proyecto un sistema que consta de una aplicación móvil y una aplicación web. La primera permite a cada entrenador mantener un registro de sus alumnos, clasificarlos en grupos y registrar los entrenamientos en tiempo real. La segunda permite tanto al entrenador como a los alumnos ver las estadísticas y resultados de las sesiones realizadas.</p>	

Abstract:

In the practice of continuous swimming, the speed and pace are used as metrics to know how a swimmer has performed on a certain course. These metrics are taken into consideration in order to prepare sport events and to improve on a personal level. Currently, the most common procedure used by coaches to manage the training sessions with their students is to record the times they get with a stopwatch and write them down manually. This process is tedious and it is usually difficult to maintain an organized data structure and obtain a global view of the evolution of students over time. To make the task of these coaches easier and more efficient, a system consisting of a mobile application and a web application is developed in this project. The first one allows each trainer to keep a record of his students, classify them in groups and record the training sessions in real time. The second one allows both the coach and the students to see the statistics and results of the sessions held.

Palabras clave:

natación, entrenador, aplicación, móvil, web

Índice

1. Introducción	11
1.1 Contexto y justificación del Trabajo	11
1.2 Objetivos del Trabajo	12
1.3 Enfoque y método seguido	12
1.4 Planificación del Trabajo	12
1.5 Breve resumen de productos obtenidos	15
1.6 Breve descripción de los otros capítulos de la memoria	15
2. Análisis estratégico	16
2.1 Introducción	16
2.2 Entorno de realización	16
2.3 Sistemas similares	16
2.3.1 SwimIO	16
2.3.2 Swim.com	17
2.3.3 SwimPB	18
2.3.4 MySwimPro	19
2.3.5 Entrenamiento de Natación	20
2.3.6 Social Swim	21
2.4 Análisis cuantitativo	22
2.5 Análisis cualitativo	25
2.6 Síntesis	26
2.7 Tecnologías a emplear	26
2.8 Conclusiones	27
3. Especificación de requisitos	28
3.1 Introducción	28
3.1.1 Propósito	28
3.1.2 Ámbito del Sistema	28
3.1.3 Definiciones, acrónimos y abreviaturas	29
3.2 Descripción general	29
3.2.1 Perspectiva del producto	29
3.2.2 Características de los usuarios	29
3.2.3 Funciones del producto	30
3.2.4 Restricciones	31
3.2.5 Suposiciones y dependencias	31
3.2.6 Requisitos futuros	32
3.3 Requisitos específicos	32
3.3.1 Interfaces externas	32
3.3.2 Funciones	33
3.3.3 Requisitos de rendimiento	37
3.3.4 Restricciones de diseño	37
3.3.5 Atributos del Sistema	37
3.4 Conclusiones	38
4. Diseño	39
4.1 Introducción	39
4.2 Especificación conceptual	39
4.3 Especificación formal	40
4.3.1 Capa de presentación	40
4.3.2 Capa de persistencia	47
4.3.3 Capa de negocio	51

4.4 Conclusión	57
5. Implementación	59
5.1 Introducción	59
5.2 Implementación	59
5.2.1 Aplicación móvil	59
5.2.1.1 Autenticación de usuarios	62
5.2.1.2 Añadir un alumno.....	64
5.2.1.3 Registro de tiempos de los alumnos durante el transcurso de una sesión de entrenamiento en tiempo real.....	67
5.2.2 Aplicación web.....	68
5.2.2.1 Dibujar gráfica del ritmo de una sesión con varios alumnos	72
5.3 Implantación	73
5.4 Conclusión	74
6. Conclusiones	75
6.1 Trabajo realizado	75
6.2 Dificultades y soluciones	75
6.3 Aportaciones	76
6.4 Trabajo futuro	77
7. Bibliografía	78

Ilustraciones

Ilustración 1: Diagrama de Gantt de la planificación del trabajo.....	13
Ilustración 2: Interfaz del sistema similar SwimIO.....	17
Ilustración 3: Interfaz del sistema similar Swim.com.....	18
Ilustración 4: Interfaz del sistema similar SwimPB.....	19
Ilustración 5: Interfaz del sistema similar MySwimPro	20
Ilustración 6: Interfaz del sistema similar Entrenamiento de Natación	21
Ilustración 7: Interfaz del sistema similar Social Swim.....	22
Ilustración 8: Diagrama de casos de uso del sistema.....	30
Ilustración 9: Diagrama de especificación conceptual del sistema	39
Ilustración 10: Diagramas de clases - Capa de presentación	41
Ilustración 11: Bosquejo de la aplicación móvil.....	42
Ilustración 12: Bosquejo de la ventana de selección de usuario de la aplicación web.....	44
Ilustración 13: Bosquejo de la ventana de inicio de sesión del alumno en la aplicación web	45
Ilustración 14: Bosquejo de la ventana de inicio de sesión del entrenador en la aplicación web	45
Ilustración 15: Bosquejo de la ventana de visión de resultados del alumno en la aplicación web	46
Ilustración 16: Bosquejo de la ventana de visión de resultados del entrenador en la aplicación web	46
Ilustración 17: Diagrama entidad-relación representativo de la base de datos	48
Ilustración 18: diagrama de la estructura de la base de datos de Firebase.....	50
Ilustración 19: Diagrama de clases - Lógica y persistencia.....	52
Ilustración 20: Diagrama de secuencia - Ver estadísticas (CU01, CU02, CU03)	53
Ilustración 21: Diagrama de secuencia - Añadir grupo (CU04)	54
Ilustración 22: Diagrama de secuencia - Eliminar grupo (CU04).....	54
Ilustración 23: Diagrama de secuencia - Añadir alumno al grupo (CU04)	55
Ilustración 24: Diagrama de secuencia - Eliminar alumno del grupo (CU04)....	55
Ilustración 25: Diagrama de secuencia - Registrar sesión de entrenamiento (CU05)	56
Ilustración 26: Diagrama de secuencia - Empezar la sesión de entrenamiento (CU06)	56
Ilustración 27: Diagrama de secuencia - Acciones durante el transcurso de una sesión de entrenamiento (CU06).....	57
Ilustración 28: Diagrama de secuencia - Registrar el último tiempo en una sesión de entrenamiento (CU06).....	57
Ilustración 29: Capturas de pantalla de la aplicación móvil 1	60
Ilustración 30: Capturas de pantalla de la aplicación móvil 2.....	61
Ilustración 31: Proveedores servicios de inicio de sesión activos en Firebase	62
Ilustración 32: Captura de pantalla de la aplicación web - elegir rol	69
Ilustración 33: Captura de pantalla de la aplicación web - inicio de sesión del alumno	69
Ilustración 34: Captura de pantalla de la aplicación web - inicio de sesión del entrenador	70
Ilustración 35: Captura de pantalla de la aplicación web - estadísticas del alumno 1 ..	70
Ilustración 36: Captura de pantalla de la aplicación web - estadísticas del alumno 2 ..	71

Ilustración 37: Captura de pantalla de la aplicación web - estadísticas del entrenador 1	71
Ilustración 38: Captura de pantalla de la aplicación web - estadísticas del entrenador 2	72

Tablas

Tabla 1: Planificación de tareas del proyecto	14
Tabla 2: Características del sistema similar SwimIO	17
Tabla 3: Características del sistema similar Swim.com	17
Tabla 4: Características del sistema similar SwimPB	18
Tabla 5: Características del sistema similar MySwimPro	19
Tabla 6: Características del sistema similar Entrenamiento de Natación	20
Tabla 7: Características del sistema similar Social Swim	22
Tabla 8: Comparativa de características cuantitativas de los sistemas similares	24
Tabla 9: Comparativa de características cualitativas de los sistemas similares	25
Tabla 10: Características del sistema	26
Tabla 11: Características de los usuarios	30
Tabla 12: Casos de uso del sistema	31
Tabla 13: Características del requisito RQ01	32
Tabla 14: Características del requisito RQ02	33
Tabla 15: Características del requisito RQ03	33
Tabla 16: Características del requisito RQ4	34
Tabla 17: Características del requisito RQ05	34
Tabla 18: Características del requisito RQ06	35
Tabla 19: Características del requisito RQ07	35
Tabla 20: Características del requisito RQ08	36
Tabla 21: Características del requisito RQ10	36
Tabla 22: Características del requisito RQ11	36
Tabla 23: Características del requisito RQ13	36
Tabla 24: Características del requisito RQ12	37
Tabla 25: Relación entre características, casos de uso y requisitos	38

Código

Código 1: Permiso de acceso a internet.....	62
Código 2: Inicializar las variables de la actividad de inicio y llamada al método que inicia la session	63
Código 3: Autenticación de usuario en la aplicación móvil	63
Código 4: Guardar un entrenador en la base de datos.....	64
Código 5: Opciones de inicio de sesión y registro	64
Código 6: Instanciación de las variables de la clase de autenticación de Firebase	65
Código 7: Creación de nuevo alumno como usuario con autenticación	66
Código 8: Registro de tiempos de los usuarios durante el transcurso de una sesión de entrenamiento.....	68
Código 9: Botón para finalizar sesión de entrenamiento	68
Código 10: Crear table de datos y añadir filas y columnas para mostrar en el gráfico de ritmo	73
Código 11: Especificar las opciones del gráfico de ritmo	73
Código 12: Dibujar gráfico de líneas del ritmo en el contenedor HTML especificado.....	73

1. Introducción

1.1 Contexto y justificación del Trabajo

El desarrollo de la actividad de la natación como deporte lleva con nosotros desde principios del siglo XIX. Habiendo varias modalidades y diversas formas de practicar la natación, una de ellas es la natación continua, que consiste en nadar de manera continuada durante cierto periodo de tiempo o hasta que se recorra determinada distancia. Es frecuente realizar rutas en aguas abiertas de varios kilómetros para los amantes de esta modalidad, pero la natación continua también se practica en espacios cerrados, como cualquier piscina, a modo de entrenamiento y para mejorar la resistencia física de un nadador de cualquier otra modalidad.

En la mayoría de modalidades, se tienen en cuenta la velocidad y los tiempos como métricas fundamentales para medir y valorar el rendimiento del nadador. Históricamente, el registro de estas métricas se ha realizado mediante herramientas de carácter rudimentario, como pueden ser relojes y cronómetros, que no tienen por qué haber sido diseñados directamente para este fin.

Actualmente existen instrumentos más modernos y específicamente diseñados para la natación, como cronómetros para entrenadores, pero su coste es elevado y siguen dependiendo de herramientas externas para registrar las marcas y llevar un seguimiento continuo del nadador en entrenamientos sucesivos. Por ejemplo, ¿Cómo se registran hoy los tiempos en una serie larga de natación continua? El entrenador ha de tomar los tiempos de cada uno de los largos de cada nadador por separado. Después anotar o registrar esos datos y hacer lo mismo con cada uno de los nadadores que corresponda. Además de llevar el control independiente de cada nadador, el entrenador ha de registrar los tiempos de manera manual, lo que supone la dedicación de tiempo adicional más allá de la sesión de entrenamiento.

En este contexto surge la evolución de la tecnología y de la ingeniería informática, capaz de diseñar herramientas que faciliten la labor de los entrenadores haciendo más sencillo el registro de los datos de entrenamientos de sus alumnos y que incluso aumenten el nivel, calidad y precisión de la información y el detalle de la actividad. Esto supone un incremento de la base de conocimiento general del desarrollo de la actividad, lo que beneficia tanto al entrenador, que le resulta más sencillo y práctico, y está más informado, como a sus alumnos, que pueden recibir un mejor asesoramiento por parte de su entrenador y conocer mejor su desempeño.

Para dar forma a esta idea, en este proyecto se desarrolla una aplicación móvil mediante la cual un entrenador es capaz de añadir alumnos, organizarlos en grupos, crear sesiones de entrenamiento y registrar sus resultados en tiempo real, es decir, conforme se desarrolla la actividad. Estos datos se almacenan en una base de datos y son posteriormente visibles por el entrenador y sus alumnos mediante una página web, donde pueden consultar sus estadísticas.

1.2 Objetivos del Trabajo

El objetivo principal del proyecto es desarrollar un sistema que permita a los entrenadores gestionar sus alumnos, registrar sesiones de entrenamiento e introducir los resultados en tiempo real para posteriormente poder ver las estadísticas. Esto permite al entrenador hacer un análisis y autoevaluar sus entrenamientos planteando cambios o mejoras de cara a futuras sesiones. Además, los alumnos al ser capaces de visualizar el desarrollo de su actividad a lo largo del tiempo pueden marcarse nuevos retos. Este objetivo global se desarrolla a través de otros objetivos específicos:

- Analizar los sistemas que presentan ámbitos o soluciones similares para poder definir las características generales del sistema para su especificación.
- Establecer los requisitos del sistema para su desarrollo.
- Diseñar el sistema de manera que se satisfagan todos los requisitos planteados.
- Implementar el diseño y evaluarlo comprobando que se cubren los requisitos.

1.3 Enfoque y método seguido

El sistema objeto de este trabajo busca mejorar los métodos actuales para el registro de entrenamientos de natación. Actualmente existen diversas aplicaciones que sirven para registrar sesiones personales de un usuario de manera individual, pero no así para entrenadores. Por tanto, el proyecto tiene como meta desarrollar un producto nuevo y original, basándose y considerando las alternativas actuales.

Para comenzar se realiza un análisis estratégico con el fin de contemplar el estado del arte, y ver qué características presentes en otros sistemas resultan de interés y qué otras no presentes han de incluirse. Una vez se definen, se puede pasar a determinar cuáles son los contextos de uso, los tipos de usuario y los casos de uso de nuestra aplicación, pudiendo especificar los requisitos necesarios que ha cumplir nuestro sistema. A continuación se establece el diseño del producto, profundizando cada una de las capas: presentación, negocio y persistencia. Con ello, es posible adentrarse a la implementación de la aplicación. Por último el sistema es sometido a evaluación, de la que se obtienen unas conclusiones y se presentan propuestas futuras de ampliación.

1.4 Planificación del Trabajo

El trabajo total del proyecto se ha repartido en tareas, a su vez divididas en subtareas. A cada una de ellas se le dedica una serie de horas. Se han excluido los días festivos que coinciden con fin de semana ya que no se pretende trabajar en el proyecto durante estos días. El número de horas por día dedicadas al trabajo serán 2 como máximo, repartidas entre las tareas que se solapan. Para la representación de la planificación del trabajo y la división de tareas se ha utilizado un diagrama de Gantt que se muestra a continuación.

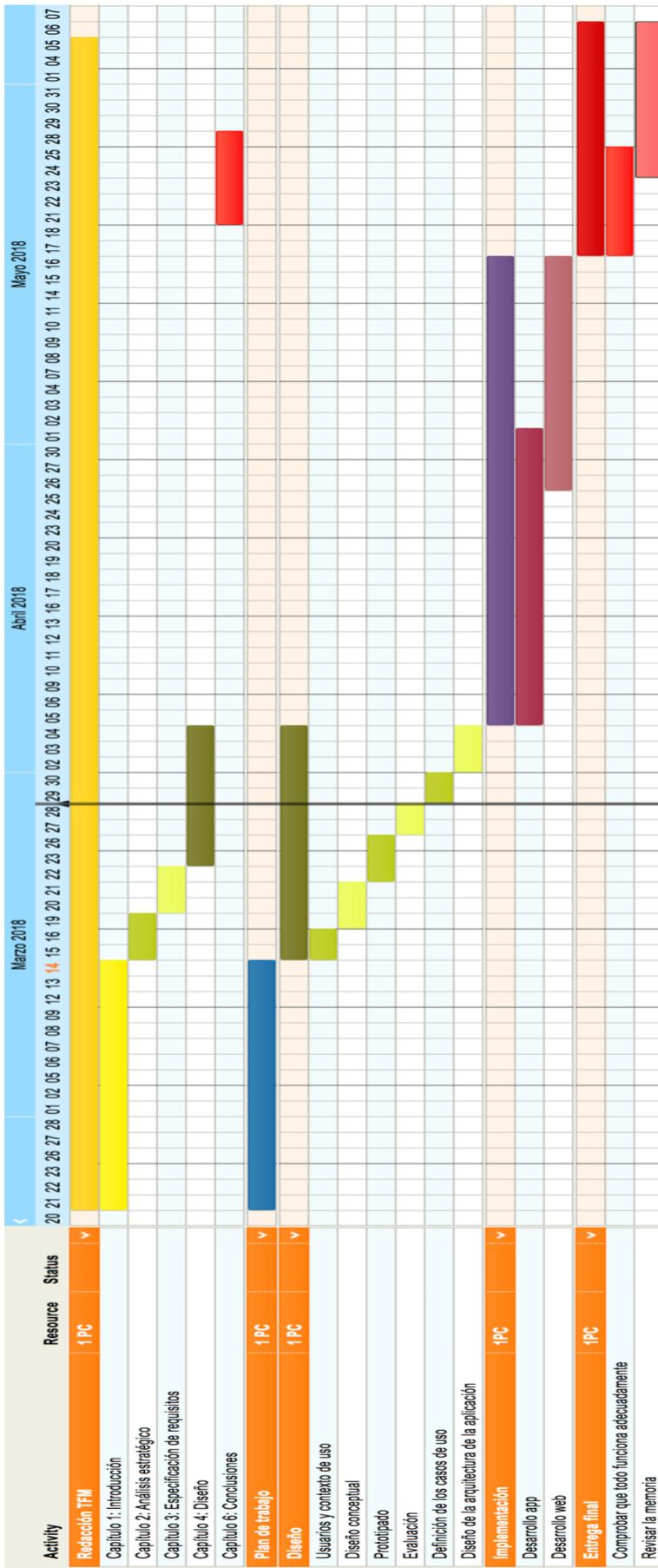


Ilustración 1: Diagrama de Gantt de la planificación del trabajo

Para clarificar la planificación de las tareas y su dedicación, se ha realizado la siguiente tabla donde cada columna describe lo siguiente:

- Tarea: Referencia a la tarea en cuestión. Se utiliza la nomenclatura TAxx.
- Subtarea: Referencia a la subtarea en cuestión. Se utiliza la misma nomenclatura que para las tareas: TAxx.
- Descripción: Se describe de manera más detallada el la tarea.
- Horas: Número de horas dedicadas a la tarea.

Tarea	Subtarea	Descripción	Horas
TA1		Redacción TFM	45
	TA11	Capítulo 1: Introducción	16
	TA12	Capítulo 2: Análisis estratégico	3
	TA13	Capítulo 3: Especificación de requisitos	3
	TA14	Capítulo 4: Diseño	9
	TA15	Capítulo 5: Implementación	11
	TA16	Capítulo 6: Conclusiones	3
TA2		Diseño	15
	TA21	Usuarios y contexto de uso	2
	TA22	Diseño conceptual	3
	TA23	Prototipado	3
	TA24	Evaluación	2
	TA25	Definición de los casos de uso	2
	TA26	Diseño de la arquitectura de la aplicación	3
TA3		Implementación	49
	TA31	Desarrollo app	34
	TA32	Desarrollo web	15
TA4		Entrega final	24
	TA41	Comprobar que todo funciona adecuadamente	8
	TA42	Revisar la memoria	16

Tabla 1: Planificación de tareas del proyecto

Podemos ver que hay 4 tareas principales que componen el desarrollo completo del proyecto. La dedicación en horas de las mismas suma 133. Este es el número total de horas que se estiman para realización del trabajo.

1.5 Breve resumen de productos obtenidos

Los productos resultantes del proyecto son dos:

- Una aplicación móvil diseñada para entrenadores de natación capaz de gestionar alumnos y sesiones de entrenamiento.
- Una página web en la que tanto alumnos como entrenadores pueden ver sus resultados y estadísticas a lo largo del tiempo.

1.6 Breve descripción de los otros capítulos de la memoria

El documento se estructura en seis capítulos que describen de manera detallada las fases de desarrollo del sistema. El contenido de cada capítulo, exceptuando la introducción, se describe a continuación.

Capítulo 2: Análisis estratégico

Una de las primeras fases a considerar en el desarrollo del proyecto consiste en la elaboración de un análisis cualitativo y cuantitativo de los sistemas similares. Este análisis nos muestra la situación del mercado, así como los puntos fuertes y débiles de los distintos sistemas. Gracias a ello se determinan las características que debe incluir el sistema a desarrollar.

Capítulo 3: Especificación de requisitos

Para representar el sistema de manera formal, se realiza una especificación de requisitos utilizando el estándar IEE 830. A raíz de las características determinadas en el capítulo dos se procede a considerar las funcionalidades y requisitos que debe incluir y satisfacer respectivamente el sistema.

Capítulo 4: Diseño

Se muestra un diagrama conceptual que explica el funcionamiento global del sistema y cómo están relacionadas sus partes. A raíz de ello y siguiendo el modelo de desarrollo software por capas, se describe el diseño de la capa de presentación, lógica y de persistencia.

Capítulo 5: Implementación

Tras el análisis estratégico, la especificación de requisitos y el diseño finalmente se concluye el desarrollo del sistema con la implementación. Se muestra el resultado final de las aplicaciones mediante capturas de pantalla además de incluir las líneas de código que son más relevantes.

Capítulo 6: Conclusiones

Por último, se establecen las conclusiones del proyecto. Se indican las dificultades encontradas durante el desarrollo y la manera de resolverlas, las aportaciones realizadas y las posibles acciones futuras de mejora del sistema.

2. Análisis estratégico

2.1 Introducción

En este capítulo se realiza un primer análisis de la situación y el entorno en la que se encuentra el ámbito de desarrollo del proyecto.

Se lleva a cabo una exploración de las distintas herramientas, en concreto aplicaciones móviles, que se incluyen en el campo de la natación y que pueden ofrecer una serie de servicios comunes al campo del proyecto. Dadas las características que presentan dichas aplicaciones se extraen y se definen las propias del proyecto.

2.2 Entorno de realización

La práctica de la natación continua no parece estar directamente relacionada con la tecnología ni con la informática. Sin embargo, cada día son más las aplicaciones móviles y dispositivos como relojes inteligentes o pulseras de actividad que ayudan al usuario a llevar sus registros de entrenamiento de natación, entre otros deportes.

A lo largo de este capítulo se realiza un análisis de este tipo de sistemas. Son probados y utilizados. Una vez analizados se pueden determinar las características que ha de tener el sistema.

2.3 Sistemas similares

A continuación, se presentan las aplicaciones móviles más populares relacionadas con la natación que pueden encontrarse para las plataformas móviles iOS y Android y que suponen un punto de partida para considerar y plantear las características que debe incluir el sistema que se desarrolla en el proyecto.

2.3.1 SwimIO

Nombre	SwimIO
Referencia	http://www.helloswimio.com/
Desarrollador	Active in Time Ltd.
Tamaño (MB)	99.3 MB
Versión	3.7.0

Descripción	SwimIO es una aplicación de fitness para nadadores con la que el usuario puede establecer y alcanzar fácilmente objetivos, registrar entrenamientos, unirse a retos y encontrar piscinas cercanas. Asimismo, permite la interacción entre usuarios pudiendo compartir sus resultados, enviar comentarios, y darle me gusta a los entrenamientos.
--------------------	--

Tabla 2: Características del sistema similar SwimIO



Ilustración 2: Interfaz del sistema similar SwimIO

2.3.2 Swim.com

Nombre	Swim.com
Referencia	https://www.swim.com/
Desarrollador	Swim.com
Tamaño (MB)	79.6 MB
Versión	1.28
Descripción	Swim.com es una aplicación para nadadores con la que el usuario puede registrar sus entrenamientos, contactar y competir con otros nadadores y encontrar piscinas o clubs de natación cercanos. Además, incluye la integración con diversos smartwatches como los distintos modelos de Apple Watch, Pebble y dispositivos con Android Wear.

Tabla 3: Características del sistema similar Swim.com

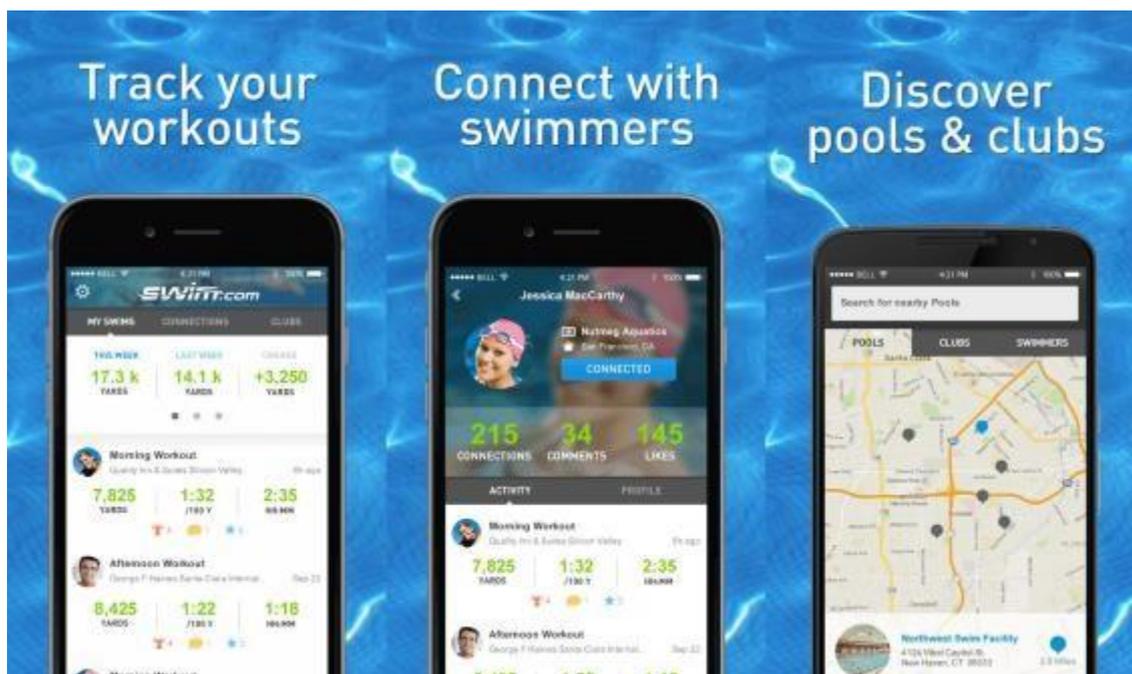


Ilustración 3: Interfaz del sistema similar Swim.com

2.3.3 SwimPB

Nombre	SwimPB
Referencia	https://play.google.com/store/apps/details?id=ie.lucca.swimtimes&hl=es
Desarrollador	cowen042
Tamaño (MB)	9.6 MB
Versión	3.2.0
Descripción	Swim PB permite mantener un registro histórico de los tiempos de un nadador (o un grupo de nadadores). Para cada tipo de ejercicio, es posible guardar el tiempo empleado en el entrenamiento y posteriormente ver una gráfica de la evolución del nadador.

Tabla 4: Características del sistema similar SwimPB

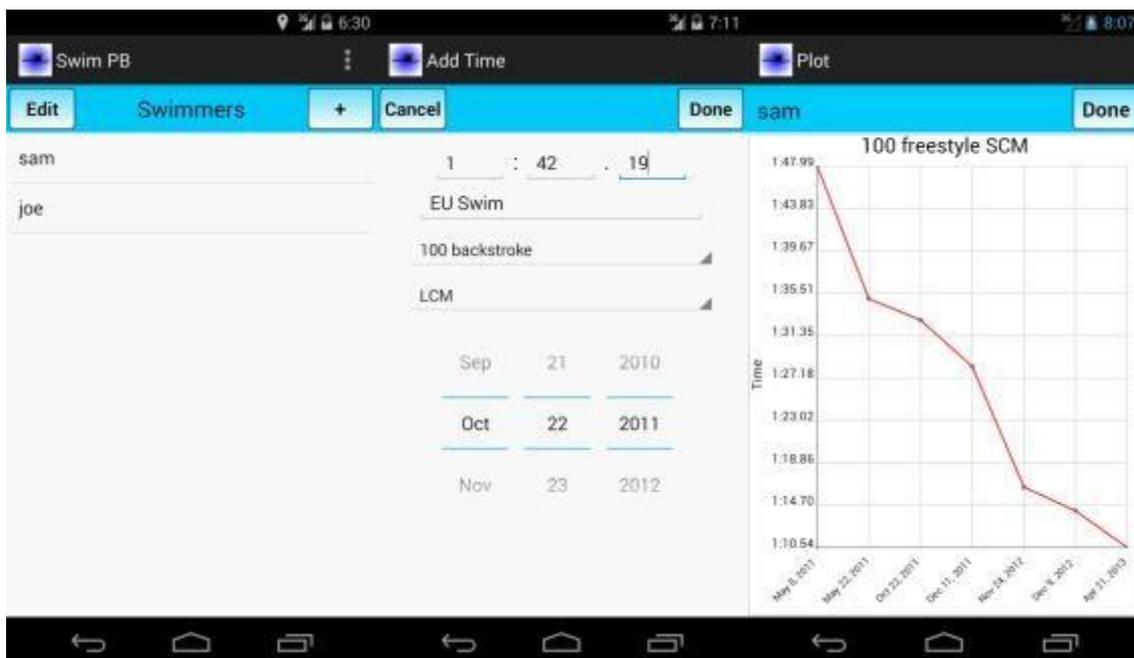


Ilustración 4: Interfaz del sistema similar SwimPB

2.3.4 MySwimPro

Nombre	MySwimPro
Referencia	http://myswimpro.com/
Desarrollador	MySwimPro
Tamaño (MB)	82.1 MB
Versión	4.3.2
Descripción	Se trata de una aplicación que además de registrar los entrenamientos de natación es capaz de crear estadísticas que muestran la evolución del usuario. Tiene también un directorio con cientos de rutinas de entrenamientos y ejercicios que puede seleccionar el usuario. Además, permite exportar los entrenamientos a PDF para imprimirlos y llevarlos a la piscina. También es posible la interacción con otros nadadores, dando me gusta y comentando sus entrenamientos.

Tabla 5: Características del sistema similar MySwimPro



Ilustración 5: Interfaz del sistema similar MySwimPro

2.3.5 Entrenamiento de Natación

Nombre	Entrenamiento de Natación
Referencia	https://goo.gl/Cemq6a
Desarrollador	App_Lab
Tamaño (MB)	91.2 MB
Versión	1.05
Descripción	En esta aplicación se encuentra un compendio de rutinas de entrenamiento y ejercicios de los distintos estilos de natación, así como vídeos explicativos sobre cómo mejorar las técnicas de cada estilo y la respiración.

Tabla 6: Características del sistema similar Entrenamiento de Natación

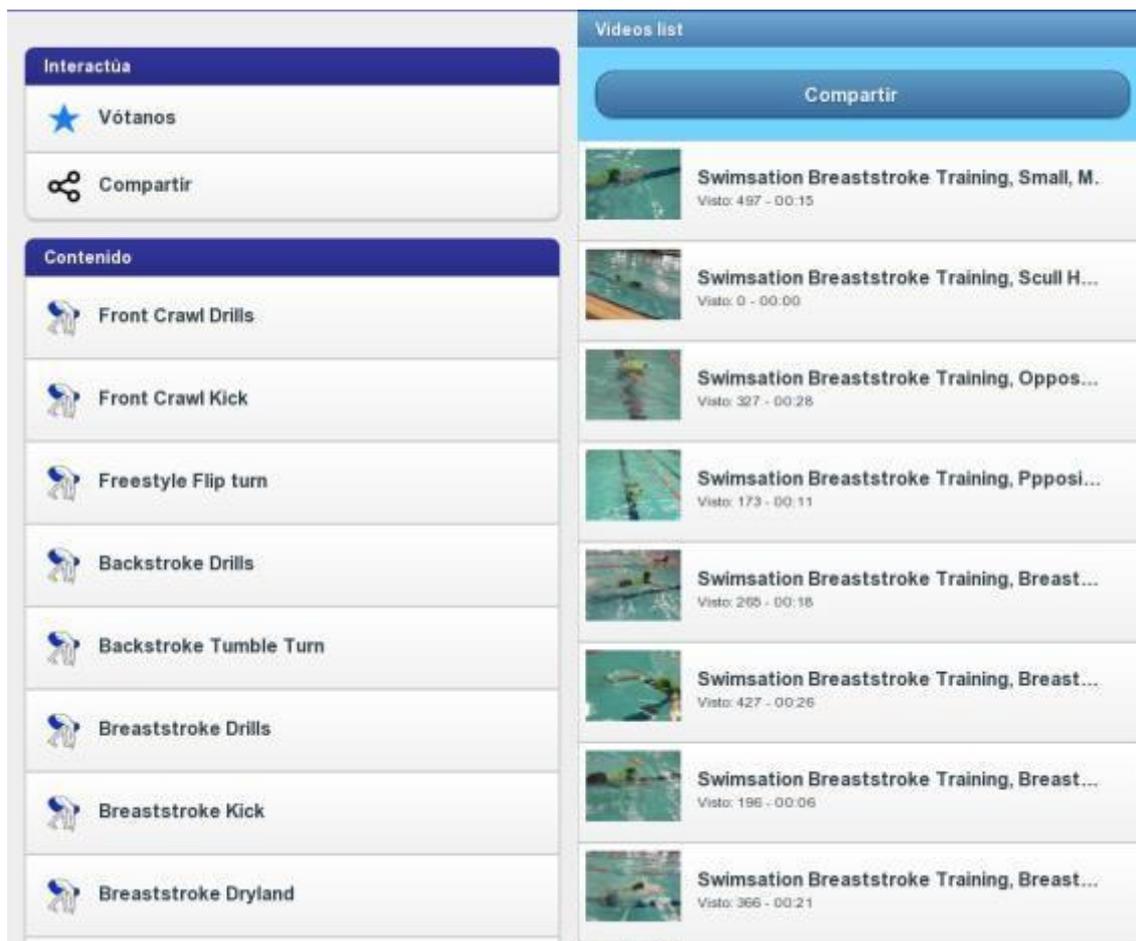


Ilustración 6: Interfaz del sistema similar Entrenamiento de Natación

2.3.6 Social Swim

Nombre	Social Swim
Referencia	http://www.socialswim.es/
Desarrollador	David Martínez Gallardo
Tamaño (MB)	19.6 MB
Versión	1.0.9
Descripción	Se trata de una aplicación que incluye una serie de características que pueden ser útiles para los nadadores habituales. Se pueden registrar los entrenamientos, así como las marcas realizadas en competiciones. Se mantiene un listado actualizado de las próximas pruebas en aguas abiertas. Es posible consultar las mínimas de diferentes campeonatos regionales, nacionales e internacionales, así como acceder a la normativa de los mismos. Recomienda tiendas especializadas de material deportivo y además incluye la posibilidad de interacción entre usuarios compartiendo entrenamientos y resultados de entrenamientos con la posibilidad de comentarlos y darles me gusta.

Tabla 7: Características del sistema similar Social Swim



Ilustración 7: Interfaz del sistema similar Social Swim

2.4 Análisis cuantitativo

En la recopilación de sistemas similares se han considerado aquellos que comparten algunos puntos e ideas que se abordan en este proyecto y que están relacionados directamente con la práctica de la natación. Este compendio nos muestra información relevante para el desarrollo del proyecto por lo que se ha realizado un profundo análisis con el fin de comparar los puntos y características observables y objetivas que comparten y diferencian a cada uno de ellos.

Las características analizadas son las siguientes:

- Disponibilidad en iOS: Describe si la aplicación está disponible para la plataforma iOS.
- Disponibilidad en Android: Describe si la aplicación está disponible para la plataforma Android.
- Registro de entrenamientos: Describe si se da soporte al registro de distintas rutinas de ejercicio.
- Establecimiento de objetivos: Describe si se da soporte a establecer una serie de metas u objetivos a cumplir.
- Búsqueda de piscinas cercanas: Describe si se da soporte a la localización de piscinas basada en la ubicación del usuario.
- Compatibilidad con smartwatches: Describe si se incluye la sincronización de las rutinas de entrenamiento con algunos modelos de smartwatches del mercado.

- Integración con apps de social media: Describe si se da soporte a la integración de la app con Twitter/Facebook u otras redes sociales.
Registro de varios perfiles de usuario: Describe si se da soporte al registro de las rutinas de ejercicios a más de un usuario.
- Rutinas de entrenamiento predefinidas: Describe si se da soporte a seguir una serie de rutinas de entrenamiento predefinidas en la aplicación.
- Vídeos de ejemplo: Describe si se incluyen vídeos demostrativos de cómo realizar determinadas técnicas de natación.
- Estadísticas de la actividad del usuario: Describe si se da soporte a la obtención de estadísticas de la actividad del usuario.
- Exportación de entrenamientos: Describe si se da soporte a exportar las rutinas de entrenamiento a otros formatos como PDF fuera de la aplicación.
- Registro de competiciones: Describe si se da soporte al registro de las marcas obtenidas en competiciones.
- Listado de pruebas: Describe si se incluye de manera actualizada un listado de las próximas competiciones de natación a nivel regional, nacional e internacional.
- Consulta de mínimas: Describe si incluye la posibilidad de consultar las marcas mínimas para varios campeonatos regionales, nacionales e internacionales.
- Recomendación de tiendas deportivas: Describe si da recomendaciones sobre tiendas deportivas.
- Orientado a nadadores: Describe si la aplicación está orientada a un perfil de usuario de entrenador.
- Orientado a Entrenadores: Describe si la aplicación está orientada a un perfil de usuario de entrenador.
- Disponibilidad en inglés: Describe si la aplicación está disponible en el idioma inglés.
- Disponibilidad en español: Describe si la aplicación está disponible en el idioma español.
- Integración con apps externas: Describe si la aplicación permite la integración de los datos recogidos con aplicaciones externas de salud y deporte como Google Fit, Salud, entre otras.
- Precio: Indica cuál el precio de la aplicación en euros.
- Tamaño: Indica cuál es el tamaño de la aplicación en MegaBytes.

Para poder observar de manera clara y esquemática las diferencias y similitudes entre los sistemas, se ha establecido una comparativa con las distintas características analizadas que se recoge en la tabla 8.

	SwimIO	Swim.com	SwimPB	MySwimPro	Entrenamiento de Natación	Social Swim
iOS	✓	✓	✓	✓	✗	✓
Android	✗	✓	✓	✓	✓	✓
Registro de entrenamientos	✓	✓	✓	✓	✗	✓
Establecer objetivos	✓	✗	✗	✗	✗	✗
Encontrar piscinas cercanas	✓	✓	✗	✗	✗	✗
Compatibilidad con smartwatches	✓	✓	✗	✗	✗	✓
Social (compartir, comentar, me gusta)	✓	✓	✗	✓	✗	✓
Registro de varios perfiles	✗	✗	✓	✗	✗	✗
Rutinas de entrenamiento predefinidas	✗	✓	✗	✓	✓	✗
Videos de ejemplo	✗	✗	✗	✓	✓	✗
Estadísticas de la actividad del usuario	✓	✓	✓	✓	✗	✗
Exportar entrenamientos	✗	✗	✗	✓	✗	✓
Registro de competiciones	✗	✗	✗	✗	✗	✓
Listado de pruebas	✗	✗	✗	✗	✗	✓
Consulta de mínimas	✗	✗	✗	✗	✗	✓
Compatibilidad piscinas	✓	✓	✓	✓	✓	✓
Compatibilidad aguas abiertas	✓	✓	✓	✓	✓	✓
Recomendación tiendas	✗	✗	✗	✗	✗	✓
Orientado a Nadadores	✓	✓	✗	✓	✓	✓
Orientado a Entrenadores	✗	✗	✓	✗	✗	✗
Disponible en inglés	✓	✓	✓	✓	✓	✓
Disponible en español	✗	✗	✗	✗	✓	✓
Integración con app externas (runkeeper, runtastic, iOS Health, Google Fit...)	✓	✗	✗	✓	✗	✗
Precio	Gratis	Gratis	0,99€	Gratis (Freemium)	Gratis	Gratis
Tamaño (MB)	99.3	79.6	9.6	82.1	91.2	19.6

Tabla 8: Comparativa de características cuantitativas de los sistemas similares

Como podemos observar, la mayoría de sistemas están disponibles en las plataformas iOS y Android, permiten registrar entrenamientos y muestran las

estadísticas del usuario. Sin embargo, la mayor parte de ellos están orientados a nadadores y no a entrenadores.

2.5 Análisis cualitativo

Además de las características objetivas y observables que presenta cada sistema, también tienen otras subjetivas y relativas pero que son importantes de igual manera. Es por ello que se ha realizado una valoración cualitativa de las características que se han considerado relevantes. Se ha empleado una puntuación de escala de valores de 1 a 5, donde cada valor representa lo siguiente:

- 1 - Nada adecuado
- 2 - Poco adecuado
- 3 - Normal
- 4 - Adecuado
- 5 - Muy adecuado

Las características que se han analizado son:

- **Facilidad de uso (usuarios):** Describe el nivel de dificultad con el que se encuentra el usuario a la hora de realizar las distintas acciones que permite la aplicación.
- **Diseño:** Describe el nivel de diseño visual y estratégico de la aplicación.
- **Funcionalidad:** Describe el número de funcionalidades que presenta la aplicación.
- **Complejidad (de la app):** Describe el nivel de complejidad que presenta la aplicación considerando su desarrollo.
- **Navegabilidad:** Describe el nivel de dificultad para navegar entre los distintos menús de la aplicación.
- **Velocidad:** Describe el nivel de rapidez con la que reacciona la aplicación ante las acciones que realiza el usuario.

Para poder observar de manera clara y esquemática las diferencias y similitudes entre los sistemas, se ha diseñado una tabla comparativa con las distintas características analizadas. Véase la tabla 9.

	SwimIO	Swim.com	SwimPB	MySwimPro	Entrenamiento de Natación	Social Swim
Facilidad de uso (usuarios)	4	4	2	4	2	2
Diseño	5	4	2	4	1	1
Funcionalidad	4	4	3	4	3	5
Complejidad (de la app)	4	4	1	4	1	3
Navegabilidad	4	4	3	4	3	2
Velocidad	4	4	4	4	5	4

Tabla 9: Comparativa de características cualitativas de los sistemas similares

Los sistemas propuestos presentan un tiempo de reacción bastante bueno ante la interacción del usuario. Sin embargo, algunos de ellos presentan un diseño y una complejidad deficientes.

2.6 Síntesis

Una vez realizado el análisis y habiendo contemplado y considerado las distintas características que presentan las aplicaciones existentes, es posible extraer y concretar aquellas que se consideren oportunas para la aplicación. A continuación, se detallan en la tabla 10 las características utilizando la nomenclatura CAxx, donde x son valores numéricos que permiten la numeración de las mismas.

Referencia	Descripción
CA01	Capacidad de registrar sesiones de entrenamiento.
CA02	Capacidad de registrar las marcas de varios nadadores simultáneamente.
CA03	Mostrar estadísticas de los nadadores sobre una sesión de entrenamiento.
CA04	Mostrar el histórico de un nadador a lo largo de distintas sesiones.
CA05	Disponibilidad de la aplicación en los idiomas español e inglés.
CA06	Crear y editar un grupo de nadadores.

Tabla 10: Características del sistema

2.7 Tecnologías a emplear

En este apartado se tienen en consideración las plataformas y tecnologías de trabajo en la elaboración del proyecto.

Considerando la tabla en la que se reflejan las características objetivas que comparten las distintas aplicaciones, se puede observar que, en general, las aplicaciones están disponibles para las plataformas iOS y Android de igual manera. Pese a ello, el número de usuarios que utilizan Android es mayor y, por tanto, la disponibilidad de la aplicación sería más amplia. Por esto, además de por la disponibilidad de recursos se ha optado por desarrollar la aplicación para esta plataforma.

Para el desarrollo de la aplicación se ha optado por utilizar el IDE Android Studio y el kit SDK, ambos en la última versión (3.1), ya que es una herramienta muy extendida y se trata del entorno de desarrollo oficial. Android Studio nos proporciona un entorno especialmente dedicado para el desarrollo

de la aplicación, así como un emulador que nos permite realizar pruebas sobre la aplicación directamente.

En la aplicación se utilizará además un módulo externo para cargar y mostrar la imagen del usuario: Picasso.

Como también se va a desarrollar un cliente web para ver las estadísticas de las sesiones, se piensa hacer uso la herramienta Google Charts para mostrar los datos de manera gráfica.

Además, se empleará la plataforma de desarrollo Firebase para almacenar las cuentas de los usuarios, así como los datos de la aplicación que posteriormente serán accesibles desde la web. También se utilizará como servicio de hosting para la web.

2.8 Conclusiones

En este capítulo se ha abordado el análisis necesario para conocer el entorno sobre el que se fundamenta el desarrollo del proyecto con el fin de concretar qué características debería incluir la aplicación a desarrollar. Para ello se han definido también las herramientas convenientes a emplear.

Se ha realizado un recorrido por las distintas aplicaciones de natación que se encuentran disponibles para las plataformas móviles existentes. Algunas de ellas ofrecen funcionalidades interesantes pero la mayoría no son completas, ofrecen un contenido pobre y sobretodo no están orientadas al entrenador, sino al nadador. De ahí que pueda mejorarse el diseño y el contenido de lo que nos ofrecen las aplicaciones actuales y desarrollar una aplicación dedicada a los entrenadores que les permita llevar el registro y control de todos sus alumnos nadadores de una manera sencilla y efectiva.

Para el desarrollo de la aplicación se ha optado por la plataforma Android y el entorno de desarrollo Android Studio junto con el kit SDK ya que nos permite el desempeño de la implementación de una manera estructurada. Además, para la gestión y almacenamiento de usuarios y sus datos se utilizará la plataforma Firebase, y para mostrar las estadísticas en la web se utilizará el servicio Google Charts.

3. Especificación de requisitos

3.1 Introducción

Una vez realizado el estudio y análisis de sistemas similares en el capítulo 2 y habiendo definido las características que tendrá el sistema planteado, así como la tecnología a emplear en su desarrollo se procede a una definición formal del sistema en cuestión. Para ello se utiliza el estándar IEEE 380 “Especificación de requisitos de software”.

Se considerará el propósito y el ámbito del sistema, se definirán los términos relevantes, así como sus acrónimos y/o abreviaturas, se incluirán las referencias utilizadas y por último se presentará una visión general del producto.

3.1.1 Propósito

La especificación de requisitos supone un medio para determinar de manera amplia y completa la descripción del sistema que se va a desarrollar.

3.1.2 Ámbito del Sistema

Al sistema se le ha denominado “SwimNCoah”. Y teniendo en cuenta los contextos de uso, podemos definir qué podrá hacer y qué no.

Qué hará:

- Registrar sesiones de entrenamiento de natación continua
- Guardar las marcas de los alumnos
- Crear grupos de alumnos
- Mostrar estadísticas al entrenador de las sesiones de entrenamiento de los alumnos
- Mostrar estadísticas al alumno de sus sesiones de entrenamiento

Qué no hará:

- Registrar sesiones de entrenamiento específicos

A continuación, se especifican los beneficios que proporciona el sistema a los usuarios.

El sistema supone una evolución sobre la manera tradicional de recoger tiempos y marcas de natación continua. Además, incrementa la base de conocimiento general del desarrollo de la actividad ya que los datos recogidos se utilizan como fuente de análisis.

El sistema facilita la labor de los entrenadores ya que el registro de datos de entrenamiento de los alumnos se realiza de una manera cómoda y ordenada.

Además, el entrenador es capaz de agrupar a sus alumnos y visualizar sus estadísticas, de manera que pueda considerar los datos y plantear mejor las futuras sesiones de entrenamiento. También resulta beneficioso para los alumnos, capaces de ver sus estadísticas y marcas para así contemplar su evolución y mejorar.

3.1.3 Definiciones, acrónimos y abreviaturas

Definiciones:

- Sistema: El proyecto en conjunto, incluyendo la aplicación móvil y la aplicación web.
- Alumno: Nadador que es tutelado por un entrenador de natación.
- Entrenador: Se trata de una persona que entrena a uno o más nadadores.
- Perfil: Conjunto de datos que se refieren al usuario del sistema.
- Marca: Tiempo que tarda un nadador en realizar determinado recorrido.
- Sesión de entrenamiento: Representa el entrenamiento realizado por uno o más alumnos dado un día, una hora, una longitud de piscina y una duración determinada.

Acrónimos:

- BD: Base de datos del sistema.

3.2 Descripción general

3.2.1 Perspectiva del producto

El sistema en cuestión es un producto independiente que no forma parte de otro proyecto. Está desarrollado de manera individual y su descripción completa y detallada está presente en el documento.

3.2.2 Características de los usuarios

En el sistema se describen dos tipos de usuarios y cada uno de ellos representa un rol con unas funciones diferentes. Los tipos de usuarios son el alumno y el entrenador.

A continuación, se presenta la tabla 11 que define el papel que desarrolla cada usuario en el sistema. Cada columna representa un usuario y las filas describen lo siguiente:

- Tipo de usuario: Referencia al usuario en cuestión.
- Actividades: Papel que desarrolla el usuario en el sistema.
- Nivel educacional: Nivel de educación del usuario.
- Experiencia: Experiencia del usuario en la actividad de natación.
- Experiencia técnica: Experiencia del usuario con la utilización de sistemas similares.

Tipo de usuario	Alumno	Entrenador
Actividades	Ver los resultados de sus entrenamientos	Registrar las sesiones de entrenamiento y llevar un control de sus alumnos
Nivel educacional	Indiferente	Estudios de natación
Experiencia	Indiferente	Alta
Experiencia técnica	Indiferente	Indiferente

Tabla 11: Características de los usuarios

3.2.3 Funciones del producto

Con el fin de presentar las funcionalidades principales del sistema se muestra a continuación un diagrama de casos de usos en la ilustración 7. De esta manera se contemplan los tipos de usuario que utilizarán el sistema y las acciones que pueden realizar.

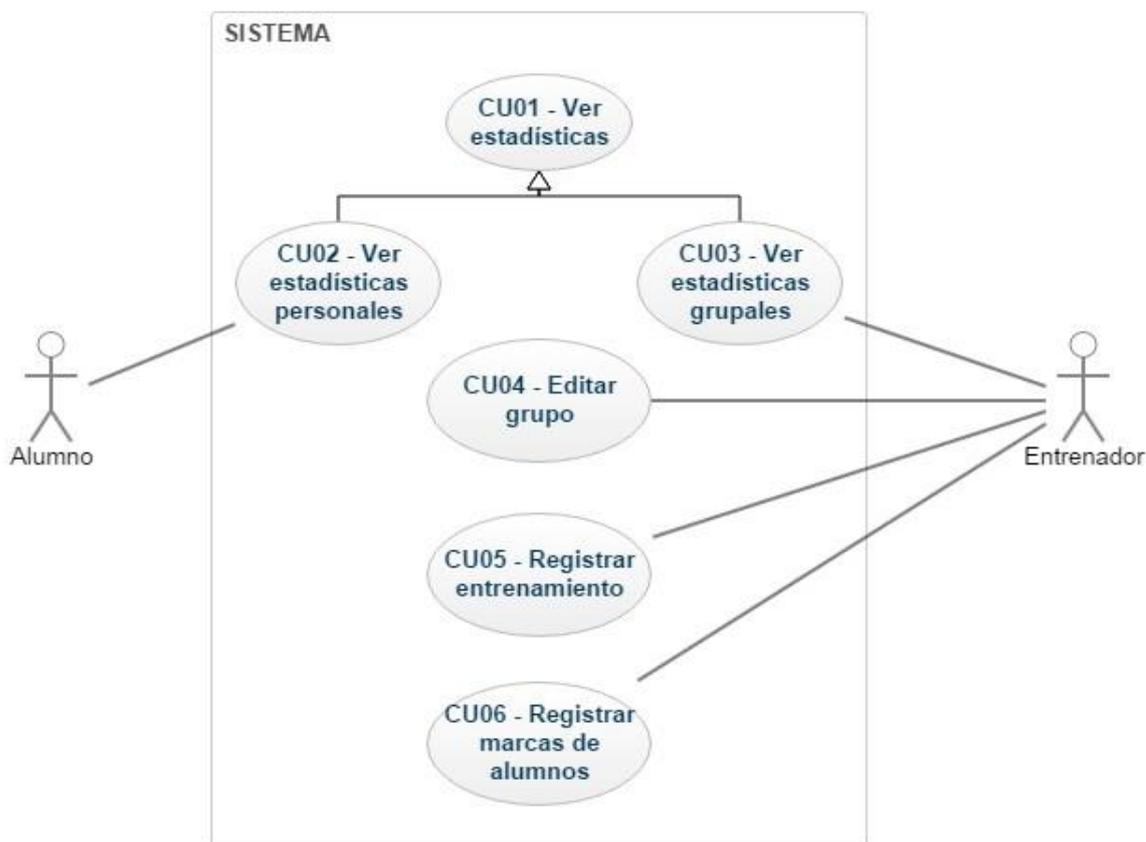


Ilustración 8: Diagrama de casos de uso del sistema

Para precisar el significado de los casos de uso descritos en el diagrama anterior se va a utilizar una tabla describiendo cada caso de uso, donde cada columna define lo siguiente:

- Caso de uso: Referencia al caso de uso que se está tratando en cuestión. Se utiliza la nomenclatura CUxx.
- Descripción: Se describe de manera más detallada el caso de uso.
- Característica: Indica la característica del sistema de la cual proviene el caso de uso. Estas características están descritas en la tabla 12.

Caso de uso	Descripción	Característica
CU01	Ver las estadísticas de las sesiones de entrenamiento.	CA04, CA03
CU02	El alumno puede ver las estadísticas personales de sus sesiones de entrenamiento.	CA04
CU03	El entrenador puede ver las estadísticas de las sesiones de entrenamiento de los grupos.	CA03
CU04	El entrenador puede editar grupos de alumnos.	CA05, CA06
CU05	El entrenador puede registrar sesiones de entrenamiento.	CA01, CA05
CU06	El entrenador puede registrar las marcas de los alumnos en una sesión de entrenamiento.	CA02, CA05

Tabla 12: Casos de uso del sistema

3.2.4 Restricciones

El sistema presenta diversas restricciones a tener en cuenta en su desarrollo. Principalmente es necesario disponer de un equipo cuyos recursos sean suficientes para ejecutar Android Studio. Además, es recomendable disponer de un dispositivo android, ya sea un smartphone o una tablet donde realizar las pruebas correspondientes sobre el sistema. Por último, el acceso a internet también es un requisito imprescindible para la interacción de la aplicación con el servidor web y la base de datos de Firebase.

3.2.5 Suposiciones y dependencias

La aplicación móvil está diseñada para funcionar en Android. Si se trasladara el sistema para funcionar también en iOS quizá fuera necesario replantear algunos de los requisitos presentes en este documento.

Para el correcto funcionamiento de la aplicación se presupone además que un alumno ha de tener un entrenador que registre las sesiones de entrenamiento. Asimismo, el entrenador ha de tener como mínimo un alumno para poder utilizar todas las funcionalidades de la aplicación. El número de alumnos influye en las estadísticas de los resultados de las sesiones de entrenamiento ya que un mayor

número de alumnos implica un resultado más representativo del nivel de cada uno.

3.2.6 Requisitos futuros

Algunas de las posibles ampliaciones que se podrían incluir en el sistema en versiones posteriores son:

- Desarrollo del sistema para iOS
- Incluir el registro de entrenamientos específicos considerando los distintos estilos de natación
- Mostrar resultados y estadísticas de los alumnos en la aplicación móvil

3.3 Requisitos específicos

En este punto se incluyen los requisitos del sistema con un mayor nivel de detalle de manera que el diseño del sistema pueda adaptarse de la mejor forma posible y satisfaga los requisitos. Cada uno de los requisitos describe un comportamiento externo del sistema, perceptible por los usuarios.

Los requisitos se recogen en forma de tabla, una por cada requisito, donde las filas de cada una describen lo siguiente:

- Referencia: Referencia por la cual se identifica cada requisito. Se utiliza la nomenclatura RQxx.
- Nombre: Nombre asignado al requisito.
- Tipo: Se diferenciará entre requisito de interfaz, funcional, rendimiento, diseño u otro.
- Descripción: Explicación del requisito.
- Prioridad: Se diferenciará entre alta, media o baja.
- Caso de uso: Referencia del requisito al caso de uso correspondiente.
- Rol: Representa el tipo de usuario que necesita o se ve afectado por el requisito.

3.3.1 Interfaces externas

Referencia	RQ01
Nombre	Diseño adaptable.
Tipo	Interfaz
Descripción	La interfaz de la aplicación tiene que ser adaptable a las pantallas de los distintos dispositivos donde pueda ser utilizada.
Prioridad	Media
Caso de uso	CU04, CU05, CU06
Rol	Alumno/Entrenador

Tabla 13: Características del requisito RQ01

3.2.2 Funciones

Referencia	RQ02
Nombre	Ver estadísticas individuales de las sesiones de entrenamiento.
Tipo	Funcional
Descripción	<p>Cada alumno ha de poder visualizar sus propias estadísticas a lo largo de distintas sesiones de entrenamiento. Se mostrarán mediante:</p> <ul style="list-style-type: none"> - Gráfico de barras de tiempo (min) - Gráfico de barras de distancia (m) - Gráfico de barras de velocidad media (km/h) <p>Además, para cada sesión se mostrará:</p> <ul style="list-style-type: none"> - Indicador del tiempo total empleado (min) - Indicador de su velocidad media (km/h) - Indicador de su ritmo medio (s/100m) - Gráfico de líneas del ritmo (s/100m) - Tabla del ritmo
Prioridad	Alta
Caso de uso	CU02
Rol	Alumno

Tabla 14: Características del requisito RQ02

Referencia	RQ03
Nombre	Registrar sesiones de entrenamiento
Tipo	Funcional
Descripción	El entrenador ha de poder registrar sesiones de entrenamiento. Esto es que puede marcar una fecha, una hora, la longitud de la piscina y la duración del entrenamiento e incluir a un grupo de alumnos para realizar la sesión.
Prioridad	Alta
Caso de uso	CU05
Rol	Entrenador

Tabla 15: Características del requisito RQ03

Referencia	RQ04
Nombre	Crear grupos de alumnos
Tipo	Funcional
Descripción	El entrenador puede seleccionar una serie de alumnos y formar un grupo con ellos. Este grupo podrá utilizarse para realizar un registro de entrenamiento.
Prioridad	Alta
Caso de uso	CU04
Rol	Entrenador

Tabla 16: Características del requisito RQ4

Referencia	RQ05
Nombre	Registrar las marcas de los alumnos en las sesiones de entrenamiento.
Tipo	Funcional
Descripción	El entrenador es capaz de ir registrando las marcas de los alumnos en una sesión de entrenamiento de manera simultánea. Cada vez que un alumno llega al borde de la piscina el entrenador pulsa en la pantalla de la aplicación sobre la vista que representa al alumno y se registra su marca.
Prioridad	Alta
Caso de uso	CU06
Rol	Entrenador

Tabla 17: Características del requisito RQ05

Referencia	RQ06
Nombre	Ver estadísticas grupales de las sesiones de entrenamiento.
Tipo	Funcional
Descripción	<p>El entrenador ha de poder visualizar las estadísticas grupales a lo largo de distintas sesiones de entrenamiento. Se mostrarán mediante:</p> <ul style="list-style-type: none"> - Gráfico de barras de tiempo (min) - Gráfico de barras de distancia (m) - Gráfico de barras de velocidad media (km/h) - Gráfico de barras de ritmo medio (s/100m) <p>Además, para cada sesión se mostrará:</p> <ul style="list-style-type: none"> - Indicador de la duración de la sesión (min) - Indicador de la velocidad media de cada alumno (km/h) - Indicador del ritmo medio de cada alumno (s/100m) - Gráfico de líneas del ritmo de cada alumno (s/100m) - Tabla del ritmo medio
Prioridad	Alta
Caso de uso	CU03
Rol	Entrenador

Tabla 18: Características del requisito RQ06

Referencia	RQ07
Nombre	Autenticación de los usuarios
Tipo	Funcional
Descripción	Tanto el entrenador como el alumno han de autenticar su información mediante un inicio de sesión con usuario y contraseña para poder acceder a las funcionalidades de la aplicación y de la página web respectivamente.
Prioridad	Alta
Caso de uso	CU02, CU03, CU04, CU05, CU06
Rol	Alumno/Entrenador

Tabla 19: Características del requisito RQ07

Referencia	RQ08
Nombre	Reordenación de los alumnos
Tipo	Funcional
Descripción	Cuando el entrenador está registrando las marcas de los alumnos durante el transcurso de una sesión de entrenamiento será posible reordenar los alumnos.

Prioridad	Media
Caso de uso	CU06
Rol	Entrenador

Tabla 20: Características del requisito RQ08

Referencia	RQ10
Nombre	Modificar grupos de alumnos
Tipo	Funcional
Descripción	El entrenador puede seleccionar un grupo de alumnos existente y añadir nuevos alumnos o eliminarlos.
Prioridad	Alta
Caso de uso	CU04
Rol	Entrenador

Tabla 21: Características del requisito RQ10

Referencia	RQ11
Nombre	Eliminar grupos de alumnos
Tipo	Funcional
Descripción	El entrenador puede seleccionar un grupo de alumnos y eliminarlo.
Prioridad	Alta
Caso de uso	CU04
Rol	Entrenador

Tabla 22: Características del requisito RQ11

Referencia	RQ13
Nombre	Sincronización de resultados de la sesión de entrenamiento
Tipo	Funcional
Descripción	Los resultados de las sesiones de entrenamiento realizadas han de subirse al servidor web de manera automática.
Prioridad	Alta
Caso de uso	CU05
Rol	Entrenador

Tabla 23: Características del requisito RQ13

3.3.3 Requisitos de rendimiento

Referencia	RQ12
Nombre	Retardo mínimo de registro de llegadas
Tipo	Rendimiento
Descripción	La aplicación debe tener un retardo máximo de 1 segundo para registrar las marcas de dos o más alumnos que llegan al mismo tiempo.
Prioridad	Alta
Caso de uso	CU06
Rol	Entrenador

Tabla 24: Características del requisito RQ12

3.3.4 Restricciones de diseño

El diseño de la aplicación debe soportar que esta sea utilizable en dispositivos que como mínimo tengan instalada la versión de Android 4.4 y que tengan un tamaño mínimo de pantalla de 4 pulgadas. Además, ha de ser compatible con todas las versiones de Android existentes entre Android 4.4 y Android 7.0.

3.3.5 Atributos del Sistema

- **Fiabilidad:** El sistema se somete a una serie de pruebas que verifican que cada una de las funcionalidades se desarrollan de manera adecuada. Pese a ello no se puede garantizar una fiabilidad del 100% dado el gran número de factores a tener en cuenta como el modelo del dispositivo utilizado, entre otros.
- **Mantenibilidad:** El sistema es objeto de mantenimiento y revisión de cara a futuras correcciones y actualizaciones.
- **Portabilidad:** El sistema solamente es accesible a los dispositivos que utilizan el sistema operativo Android. Pese a ello, esto es posible en millones de teléfonos inteligentes y tabletas.
- **Seguridad:** Los datos de los usuarios se encuentran almacenados en la base de datos del servidor y solamente son accesibles mediante un sistema de autenticación de usuario y contraseña.

Como se menciona en el apartado 3.2.2 “Características de Usuario”, existen dos tipos de usuario: el entrenador y el alumno. Las tareas que puede realizar cada uno de ellos se recogen en el apartado “Funciones del producto” y se corresponden con los casos de uso con los que se relaciona cada actor.

Para garantizar que los usuarios están autorizados a realizar cada tarea se emplea un mecanismo de seguridad de autenticación de email y contraseña asociada. De esta manera el entrenador podrá acceder a las funciones incluidas en su rol y el alumno podrá hacer lo propio.

3.4 Conclusiones

En este capítulo se ha realizado la especificación de requisitos del sistema siguiendo el estándar IEEE 380. Conociendo las características que se presentaron en el capítulo dos y mediante la utilización de un diagrama de casos de uso se han podido concretar los requisitos específicos que debe satisfacer el sistema y que hay que implementar. En la siguiente tabla se recoge la relación entre las características, los casos de uso y los requisitos:

Requisito	Casos de Uso	Características
RQ01	CU04, CU05, CU06	CA01, CA02, CA05, CA06
RQ02	CU02	CA04
RQ03	CU05	CA01, CA05
RQ04	CU04	CA05, CA06
RQ05	CU06	CA02, CA05
RQ06	CU03	CA03
RQ07	CU02, CU03, CU04, CU05, CU06	CA04, CA03, CA06, CA05, CA01, CA02
RQ08	CU06	CA02, CA05
RQ09	CU02, CU03	CA03, CA04
RQ10	CU04	CA05, CA06
RQ11	CU04	CA05, CA06
RQ12	CU06	CA02, CA05
RQ13	CU05	CA01, CA05

Tabla 25: Relación entre características, casos de uso y requisitos

4. Diseño

4.1 Introducción

Una vez realizada la especificación de requisitos en el capítulo anterior, es necesario especificar el diseño formal del sistema. Esto se abordará en el presente capítulo y para ello se utilizará el lenguaje de modelado UML.

En primer lugar, se realiza una especificación conceptual del sistema que representa, de manera genérica, su estructura y funcionamiento. Más adelante se especificarán las partes que lo componen, separándolas por capas diferenciadas: la capa de persistencia, la de lógica y la de presentación. Todo ello se muestra mediante diagramas e imágenes.

4.2 Especificación conceptual

Para visualizar la estructura y el funcionamiento global del sistema, se presenta la ilustración 8.

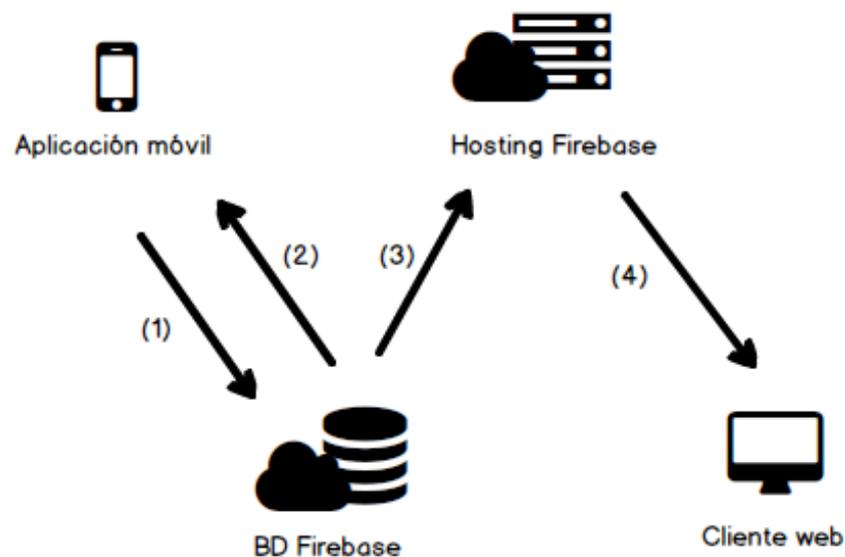


Ilustración 9: Diagrama de especificación conceptual del sistema

Un usuario con rol de entrenador interactúa con la aplicación móvil. Desde ella es capaz de agregar alumnos, crear y editar grupos, crear y editar sesiones de entrenamiento y registrar las marcas de los alumnos en cada sesión. Estos datos se envían y se escriben en la base de datos (1). Dicha aplicación también es capaz de leer la base de datos (2), pues todos los datos que se almacenan están asociados al usuario y desde cualquier dispositivo en que utilice la aplicación puede acceder a ellos. Asimismo, el servidor web puede leer los datos de la base de datos (3) y mostrar así las estadísticas de las sesiones de entrenamiento ante un usuario autenticado que mediante un navegador acceda a él (4).

4.3 Especificación formal

En este apartado se aborda la formalización de las tres capas de implementación del sistema. Primero se presenta la capa de presentación, posteriormente la de persistencia y por último la capa de negocio.

4.3.1 Capa de presentación

En este apartado se muestran una serie de bocetos de lo que supone la capa de presentación, es decir la que ve el usuario y con la cual interactúa. Cada ilustración se encuentra acompañada de la explicación correspondiente para entender mejor el diseño de la interfaz. En primer lugar, se presentan los bosquejos de la aplicación móvil y en segundo lugar los de la aplicación web.

Aplicación móvil

Para un diseño adecuado de la capa, antes de comentar en profundidad la interfaz se va a mostrar un diagrama de clases con las clases que participan en la capa de presentación para ver su interrelación. Este lo encontramos en la ilustración 9. Como podemos observar, se utilizan actividades y fragmentos de Android para distribuir las vistas.

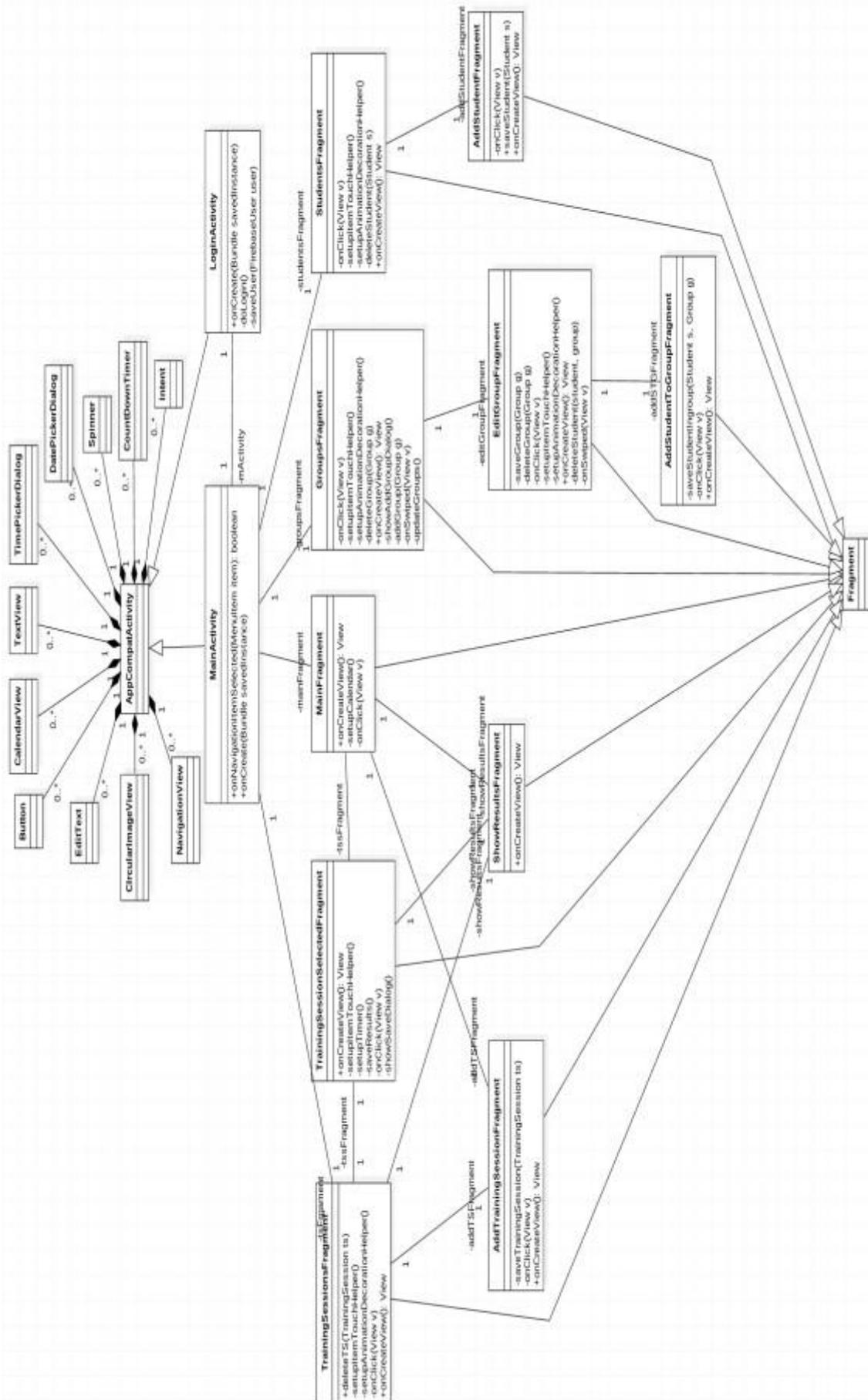


Ilustración 10: Diagramas de clases - Capa de presentación

La interfaz de la aplicación trata de ser sencilla y cómoda para el usuario. En la ilustración 10 puede observarse de qué manera evoluciona la aplicación conforme el usuario la utiliza, es decir, el árbol de navegación.

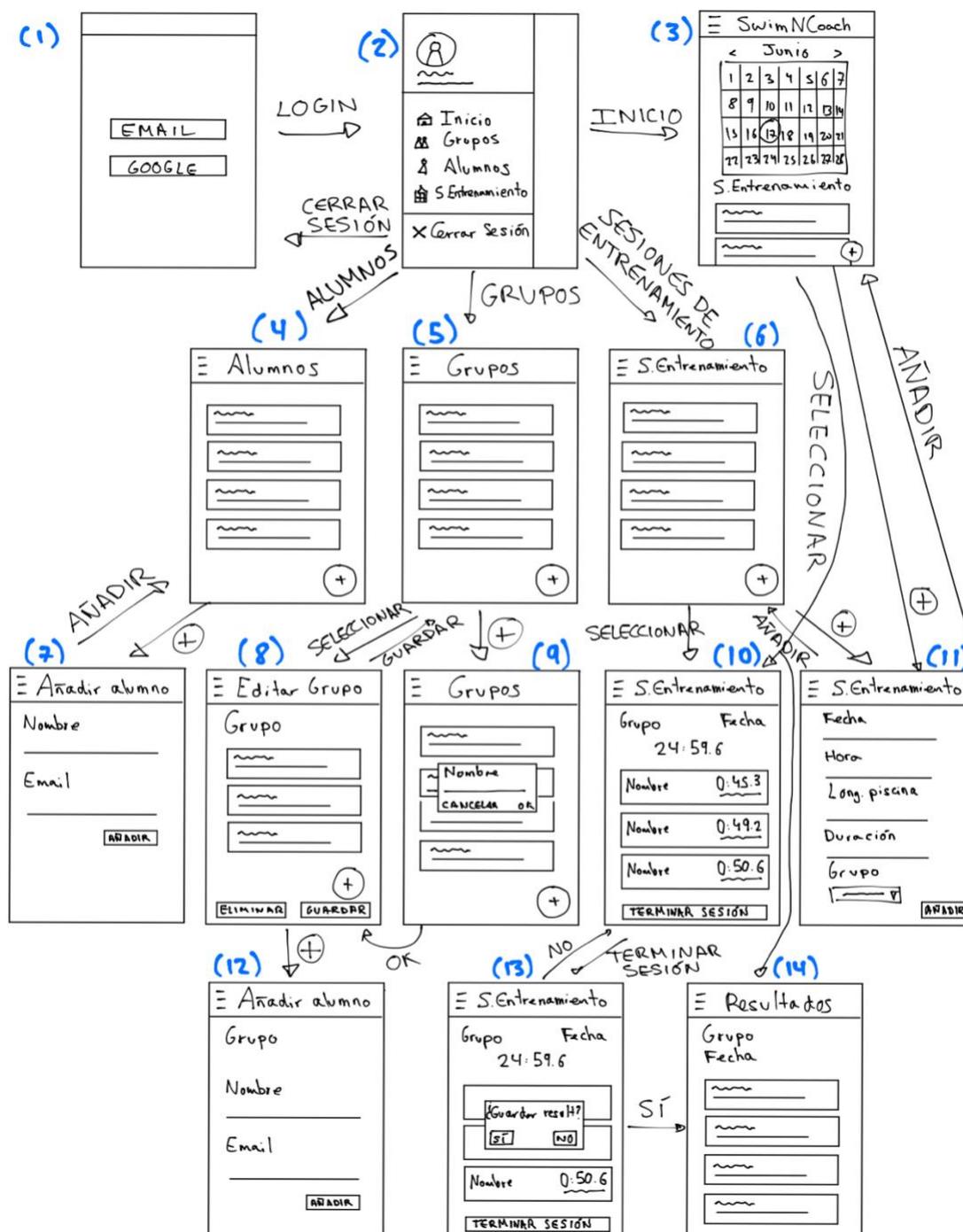


Ilustración 11: Bosquejo de la aplicación móvil

La primera vez que se abre la aplicación el usuario se encuentra con una ventana de inicio de sesión (1). Desde ella puede acceder con email y contraseña o a través de su cuenta de Google a la ventana principal (2). Desde esta el entrenador puede acceder a las secciones más importantes de la aplicación:

(3): Desde esta ventana el entrenador puede ver un calendario por el que puede navegar y ver las sesiones de entrenamiento que tiene programadas o que ya se han realizado cada uno de los días. Desde aquí, puede añadir una sesión de entrenamiento para el día seleccionado (11) pulsando el botón +, empezar a registrar tiempos en una sesión de entrenamiento seleccionada que no se ha realizado todavía (10) o ver los resultados de una sesión seleccionada que ya se ha realizado (14).

(4): Desde esta ventana el entrenador ve todos los alumnos que tiene registrados. Puede añadir nuevos alumnos (7) pulsando el botón + o eliminarlos deslizando hacia la izquierda sobre el alumno que se desea eliminar.

(5): Desde esta ventana el entrenador puede ver los grupos que tiene registrados. Puede añadir nuevos grupos (9) pulsando el botón +. Además, puede editar los grupos (8) para añadir nuevos alumnos al grupo (12) y eliminar alumnos del grupo deslizando hacia la izquierda sobre el alumno que se desee borrar. También puede borrar un grupo deslizando hacia la izquierda sobre el grupo y desde el botón ELIMINAR en la ventana de editar grupo (8).

(6): Desde esta ventana el entrenador puede ver las sesiones de entrenamiento que tiene registradas. Puede añadir nuevas sesiones (11) pulsando el botón +, ver los resultados de una sesión ya realizada (14) seleccionándola o acceder a la ventana para comenzar a registrar los tiempos de los alumnos (10) si se selecciona y no se ha realizado todavía. Desde esta última ventana (10) el entrenador puede ir marcando los tiempos de cada alumno al realizar un largo de piscina pulsando sobre el alumno correspondiente. Además, puede reorganizar en la lista el orden de los alumnos manteniendo pulsado uno de ellos y deslizándolo arriba o abajo. Una vez termine el tiempo de la sesión o si se pulsa el botón TERMINAR SESIÓN se pregunta al usuario si dese guardar los resultados (13), de ser así se muestran los resultados de la sesión (14).

Hace falta señalar que, con respecto al inicio de sesión, si el usuario todavía no tiene una cuenta asociada al sistema se le guía a través de un proceso para hacerlo donde debe establecer una contraseña si opta por el método de email o donde debe aceptar los permisos necesarios para utilizar su cuenta de Google si elige el método de Google. También es necesario mencionar que todos los campos que se han de rellenar por el usuario se comprueba que son válidos antes de poder introducirlos.

Aplicación web

Con respecto a la aplicación web también se pretende mantener un diseño sencillo e intuitivo. A continuación, en las ilustraciones 12-16 donde se recogen las distintas ventanas por las que se puede mover el usuario.

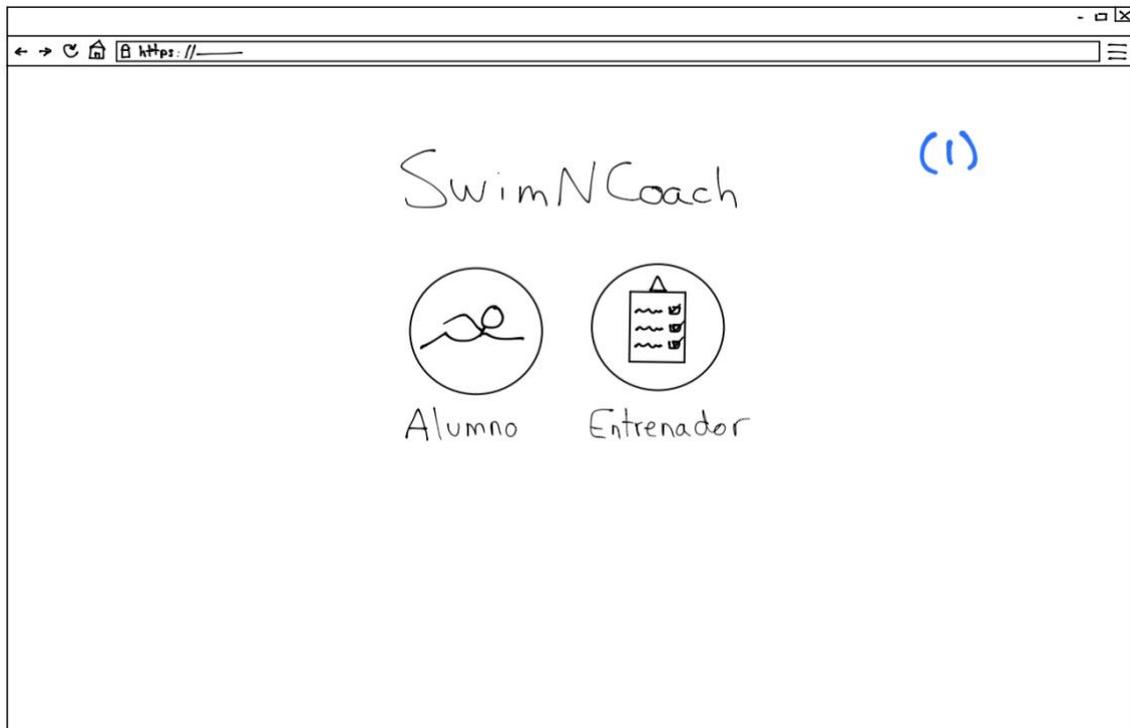


Ilustración 12: Bosquejo de la ventana de selección de usuario de la aplicación web



Ilustración 13: Bosquejo de la ventana de inicio de sesión del alumno en la aplicación web

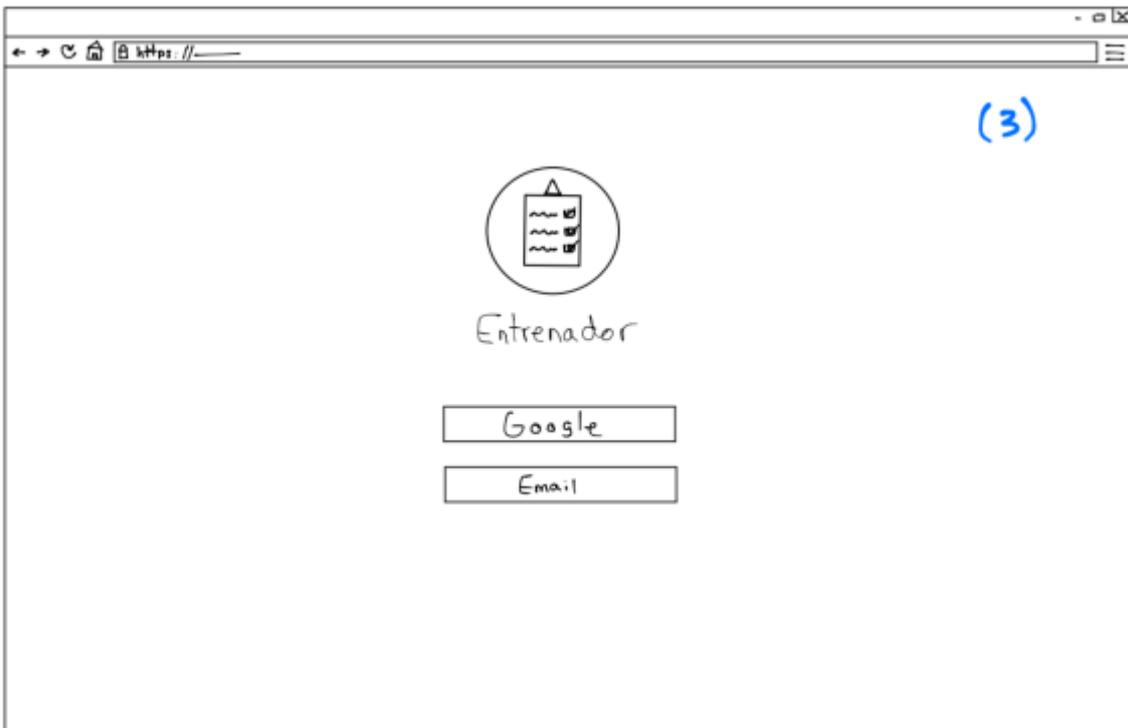


Ilustración 14: Bosquejo de la ventana de inicio de sesión del entrenador en la aplicación web

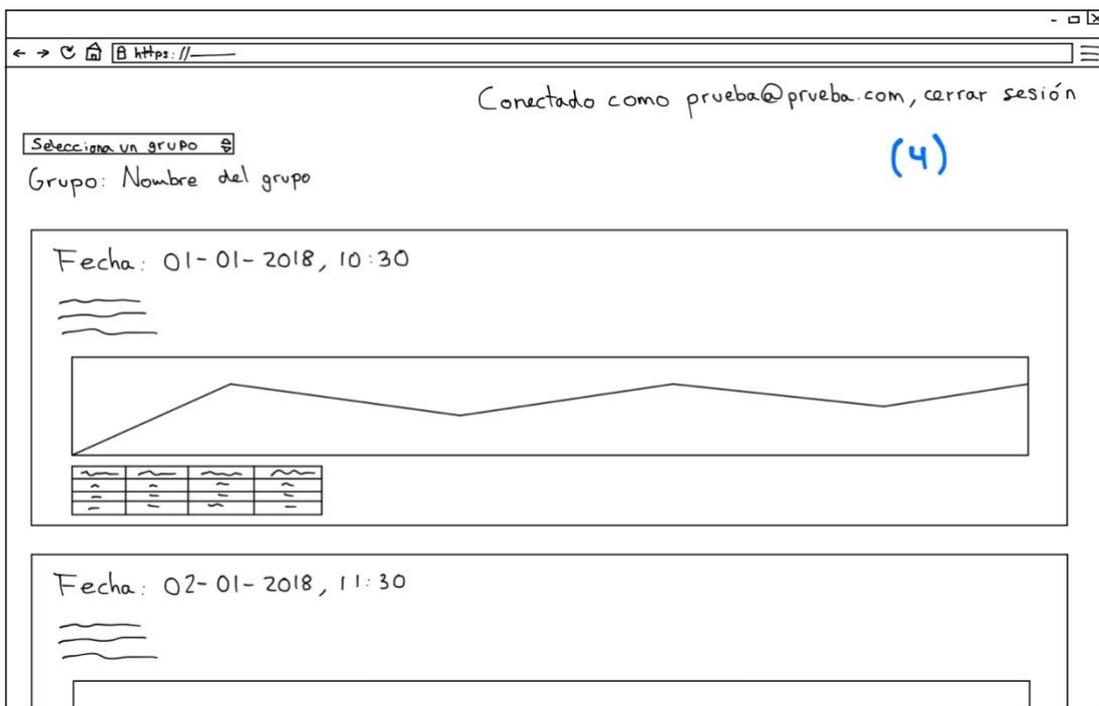


Ilustración 15: Bosquejo de la ventana de visión de resultados del alumno en la aplicación web

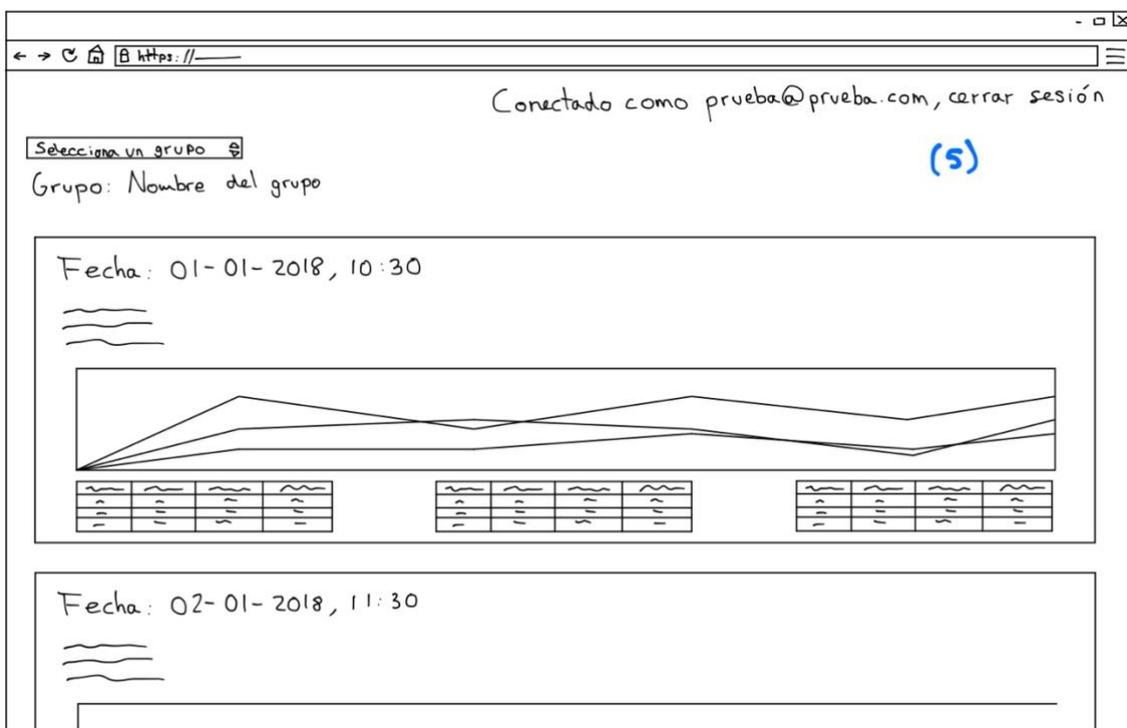


Ilustración 16: Bosquejo de la ventana de visión de resultados del entrenador en la aplicación web

En la ventana de selección de usuario (1), el alumno o entrenador deben seleccionar su rol. Esto les lleva a la ventana de inicio de sesión de alumno (2) y la ventana de inicio de sesión del entrenador (3) respectivamente. El alumno solamente puede acceder mediante email y contraseña y el entrenador puede hacerlo mediante email o cuenta de Google. Esto se debe a que las cuentas de

los alumnos son creadas por el entrenador cuando registra a los alumnos en la aplicación móvil, para lo que necesita sus emails.

Una vez el usuario se ha autenticado, dependiendo de si es alumno o profesor se acceden a las siguientes ventanas:

(4): Si es un alumno. Puede ver los detalles de cada sesión de entrenamiento (fecha, hora, duración, longitud de la piscina) y sus resultados individuales (velocidad media, ritmo medio, tiempo total, distancia total, gráfico de líneas de ritmo y tabla de ritmo). Además, al final puede ver un resumen total de sus resultados a lo largo del tiempo. Se incluye un gráfico de barras de distancia total, otro de tiempo total, otro de ritmo medio y otro de velocidad media. En principio ve las estadísticas de todas las sesiones de entrenamiento que ha realizado independientemente del grupo al que pertenezca durante la sesión, pero si ha participado en varios grupos puede seleccionar entre ellos para poder ver sus estadísticas por grupo. Si desea cerrar sesión y volver a la pantalla de selección de usuario (1), puede hacerlo pulsando “cerrar sesión”, arriba a la derecha.

(5): Si es un entrenador. Puede ver los detalles de cada sesión de entrenamiento (fecha, hora, duración, longitud de la piscina) y los resultados individuales de cada alumno (velocidad media, ritmo medio, tiempo total, distancia total) además de un gráfico de líneas comparativo del ritmo, así como una tabla de ritmo por cada alumno. Además, al final puede ver un resumen total de los resultados de las sesiones a lo largo del tiempo. Se incluye un gráfico de barras de tiempo total, otro de distancia total, otro de ritmo medio y otro de velocidad media. En principio ve las estadísticas de todas las sesiones de entrenamiento que ha gestionado independientemente de los grupos. Pero puede seleccionar entre los grupos que dirige para ver las estadísticas por grupo. Si desea cerrar sesión y volver a la pantalla de selección de usuario (1), de la misma manera que se ha explicado anteriormente para el alumno, el entrenador puede pulsar “cerrar sesión”.

4.3.2 Capa de persistencia

Para estructurar la capa de persistencia se utiliza la base de datos de Firebase. Se trata de una base de datos NoSQL que organiza los datos de manera distinta a las clásicas bases de datos relacionales. En lugar de tablas, se utilizan objetos de tipo JSON para almacenar la información.

Para una mejor comprensión de la estructura de los datos, se presenta en la ilustración 16 un diagrama entidad-relación de la base de datos y posteriormente se indican los cambios realizados a la hora de utilizar el modelo no relacional.

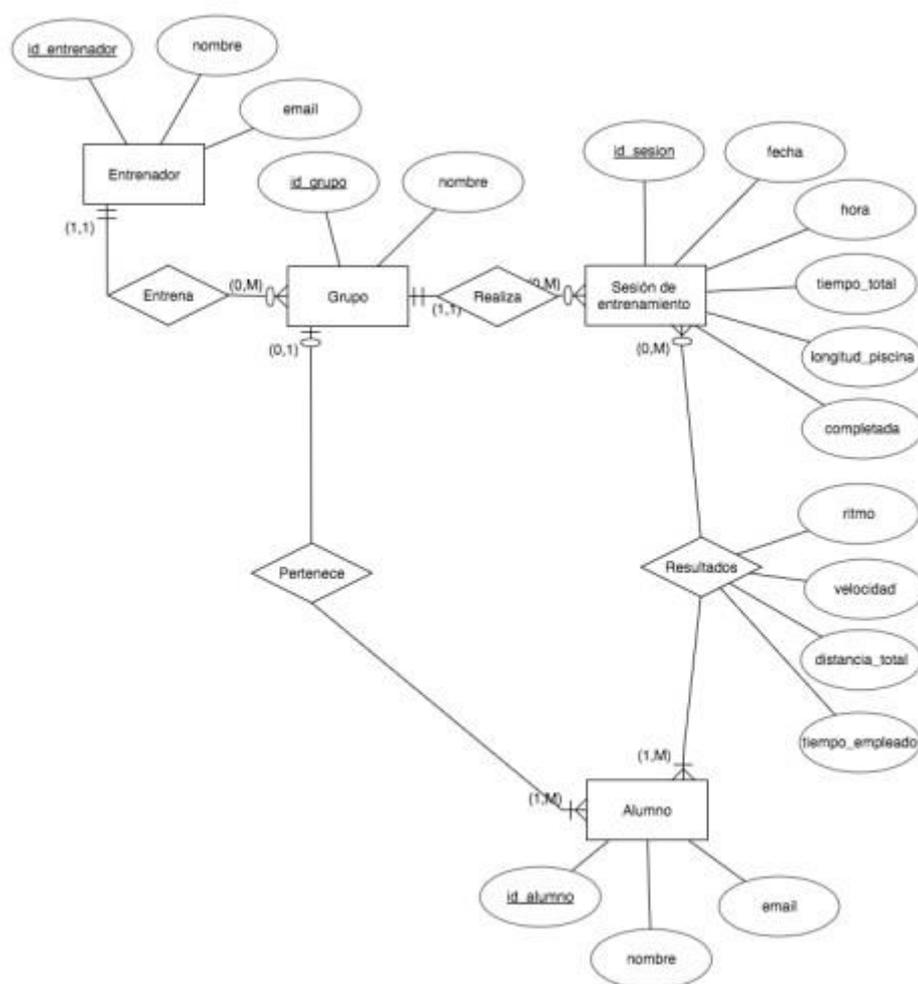


Ilustración 17: Diagrama entidad-relación representativo de la base de datos

La información que se almacena en la base de datos del diagrama se recoge a continuación:

Entrenador: Se trata del usuario principal del sistema. Tiene los siguientes atributos:

- id_entrenador: identificador del entrenador
- nombre: nombre del entrenador
- email: email del entrenador

Grupo: Consiste en una agrupación de alumnos. Tiene los siguientes atributos:

- id_grupo: identificador del grupo
- nombre: nombre del grupo

Alumno: Nadador que es tutelado por el entrenador. Tiene los siguientes atributos:

- id_alumno: identificador del alumno
- nombre: nombre del alumno

- email: email del alumno

Sesión de entrenamiento: Consiste en un entrenamiento definido por el entrenador. Tiene los siguientes atributos:

- fecha: fecha de la sesión de entrenamiento
- hora: hora de la sesión de entrenamiento
- tiempo_total: duración de la sesión de entrenamiento
- longitud_piscina: longitud del largo de la piscina
- completada: indicador de si la sesión ya se ha realizado o no

Resultados: Se corresponde con los resultados obtenidos por los alumnos en las sesiones de entrenamiento. Tiene los siguientes atributos:

ritmo: ritmo que el alumno

- velocidad: velocidad media del alumno en la sesión de entrenamiento
- distancia_total: distancia total recorrida por el alumno durante la sesión de entrenamiento
- tiempo_employado: tiempo empleado por el alumno al realizar la sesión de entrenamiento

El diagrama de entidad-relación de la ilustración 16 se convierte en un objeto JSON para representarlo adecuadamente en la base de datos de Firebase. Esto se representa en la ilustración 17.

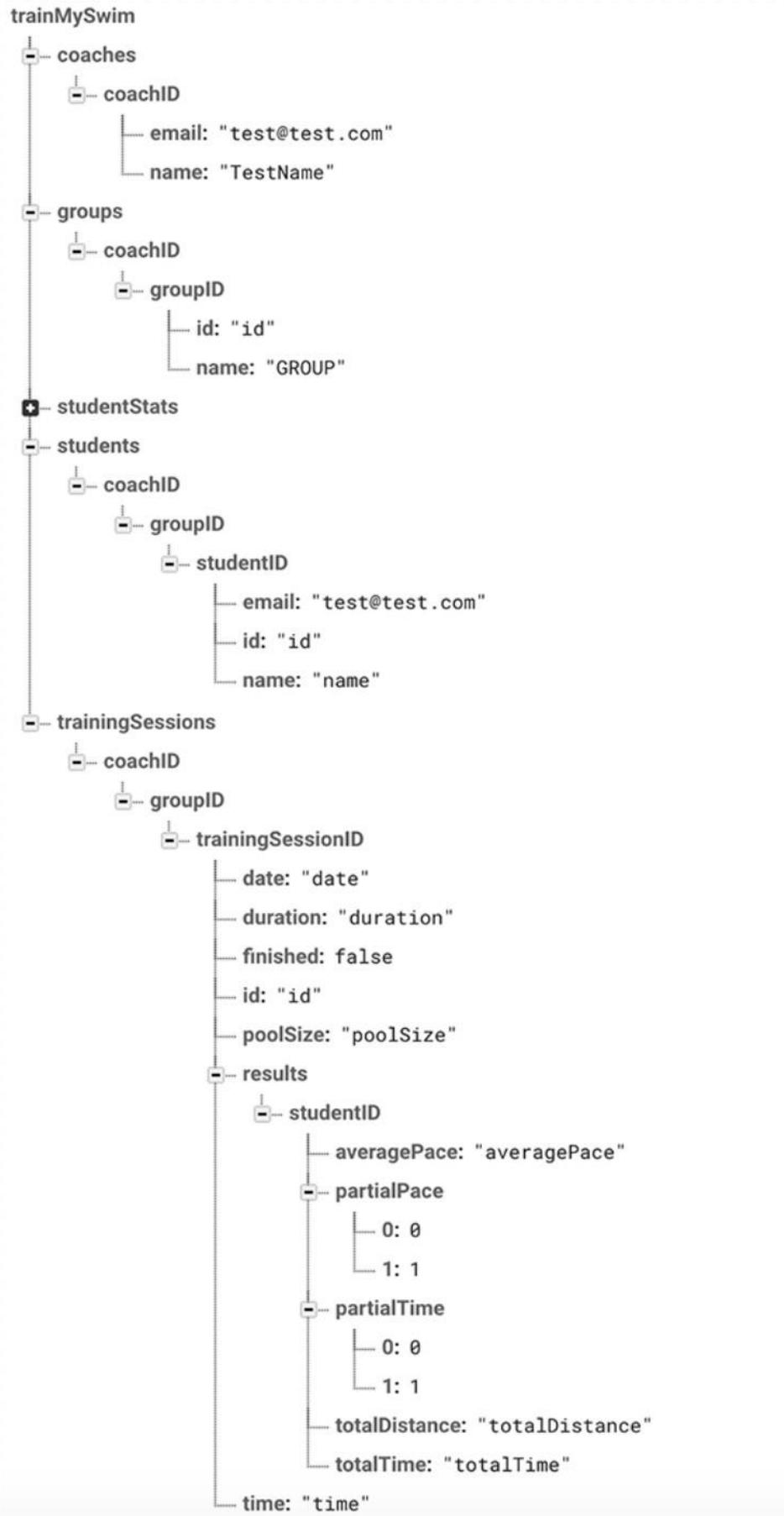


Ilustración 18: diagrama de la estructura de la base de datos de Firebase

Los nodos principales son coaches, groups, students y trainingSessions. Estos se corresponden con entrenador, grupo, alumno y sesión de entrenamiento.

La estructura de los datos se ha modificado ligeramente con el fin de optimizar las llamadas para recuperar datos. En este tipo de base de datos se debe aplicar un proceso de desnormalización y separar los posibles datos anidados en distintos caminos de manera que se pueda acceder a ellos eficientemente en llamadas separadas conforme se necesiten.

Por ejemplo, con esta estructura si el entrenador quisiera acceder a los datos de un alumno en una sesión de entrenamiento concreta, a través del nodo trainingSessions se puede acceder al nodo de los resultados del alumno para la sesión específica y descargar solamente esa información.

4.3.3 Capa de negocio

En este apartado se muestra el desarrollo de la lógica de la aplicación, cuyos aspectos más relevantes se representan mediante diagramas de secuencia. Pero primero se presenta un diagrama de clases que incluye aquellas clases implicadas en la lógica y la persistencia de la aplicación. Este diagrama nos ayudará a visualizar mejor la relación y el funcionamiento de las mismas. Se puede encontrar en la ilustración 18.

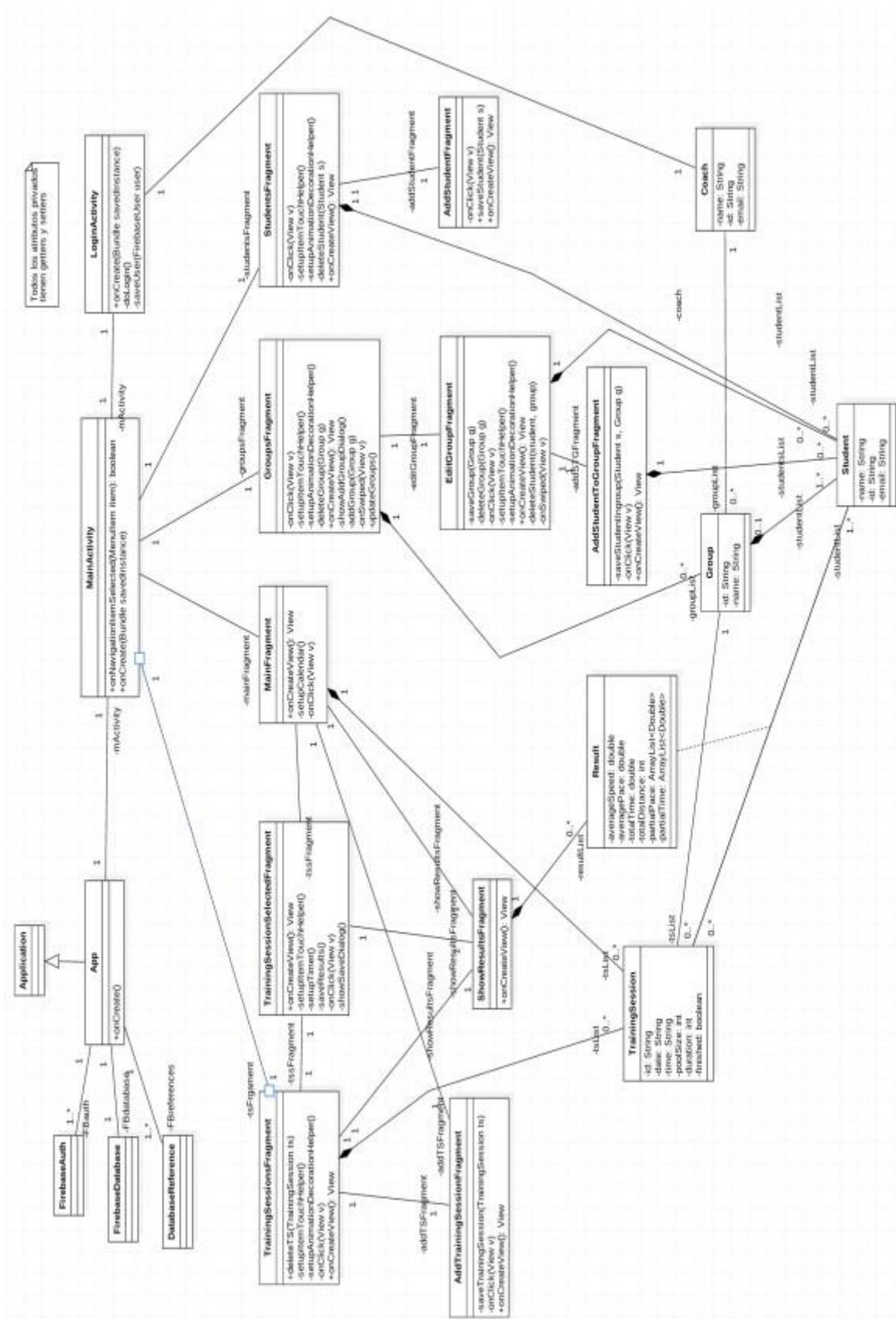


Ilustración 19: Diagrama de clases - Lógica y persistencia

Se han elaborado los diagramas de los casos de uso establecidos en el apartado 3.2.3.

El primer caso de uso recoge la acción de ver estadísticas (CU01). El segundo y el tercero descienden del primero ya que implican ver estadísticas personales (CU02) y ver estadísticas grupales (CU03), uno de ellos relacionado con el rol de alumno y el otro con el de profesor. Debido a la similitud en el diseño se mostrará en un único diagrama el primer caso de uso (CU01) y englobará a los otros dos. Esto se recoge en la ilustración 20.

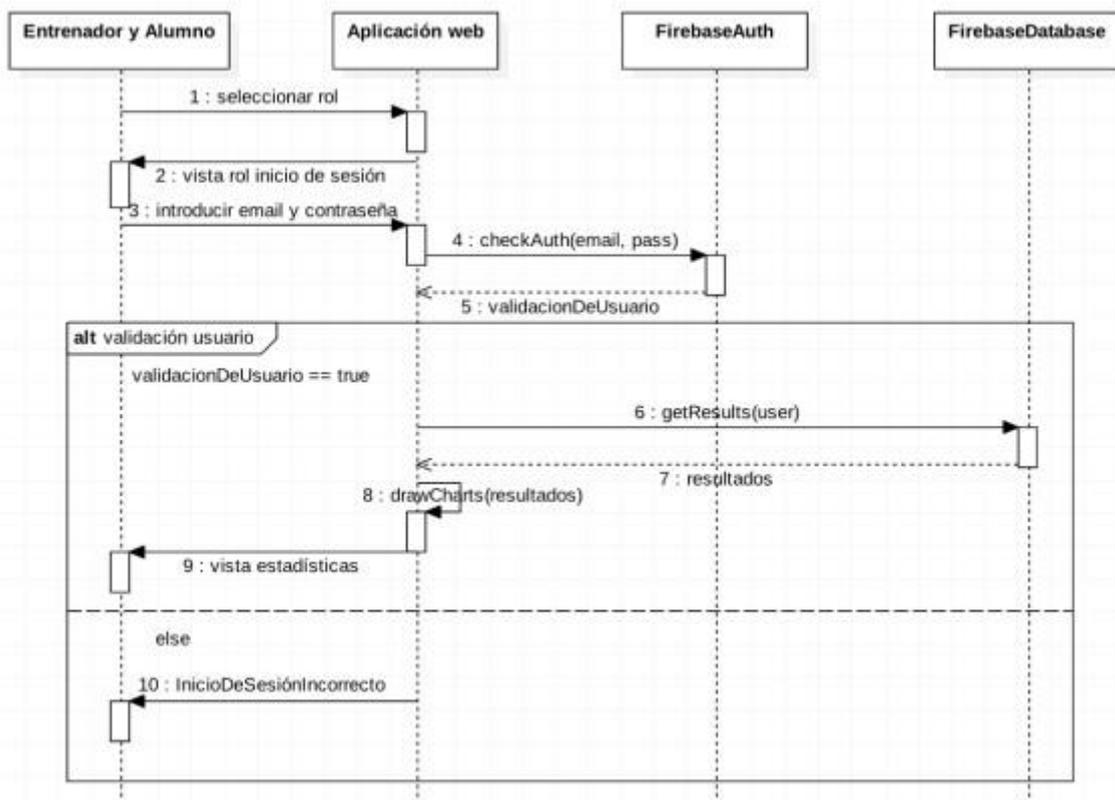


Ilustración 20: Diagrama de secuencia - Ver estadísticas (CU01, CU02, CU03)

El caso de uso correspondiente a la edición de grupos (CU04) se subdivide en cuatro acciones: añadir grupo, eliminar grupo, añadir alumno al grupo y eliminar alumno del grupo. Para una mayor claridad se representan las acciones en diagramas separados que se corresponden con las ilustraciones 21, 22, 23 y 24.

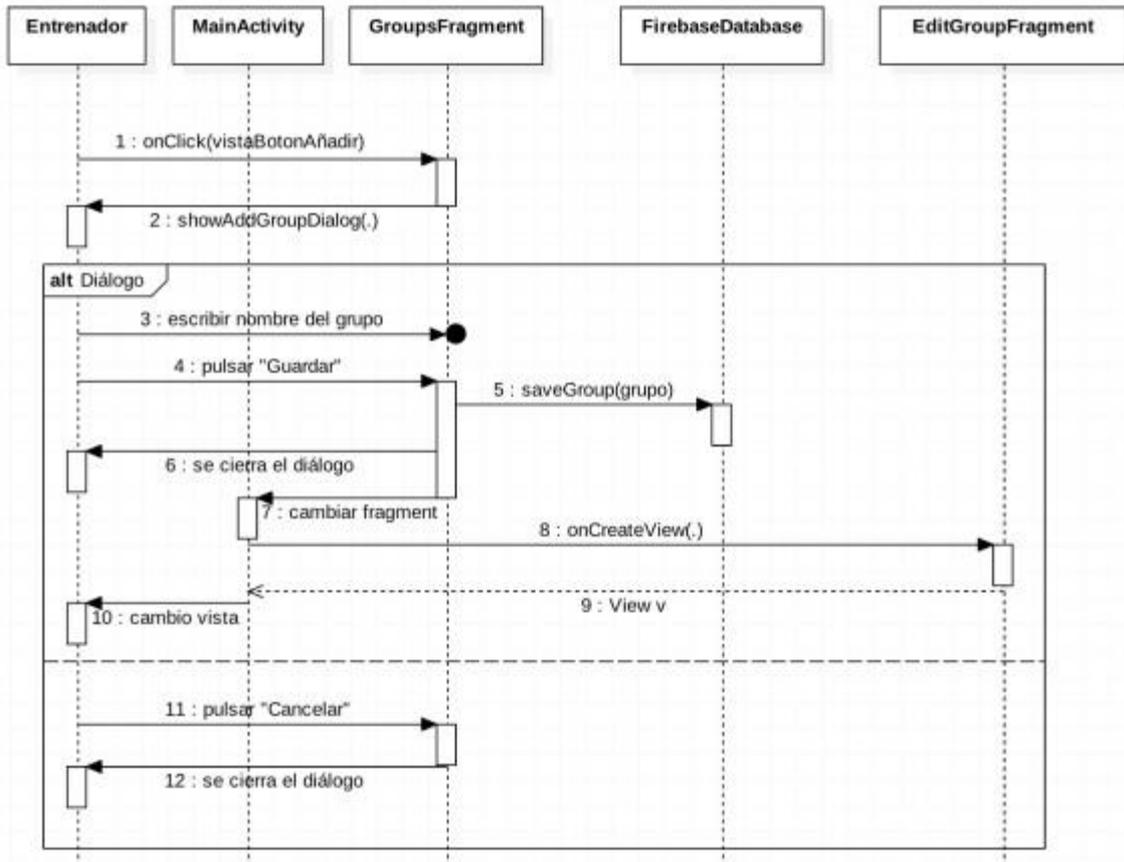


Ilustración 21: Diagrama de secuencia - Añadir grupo (CU04)

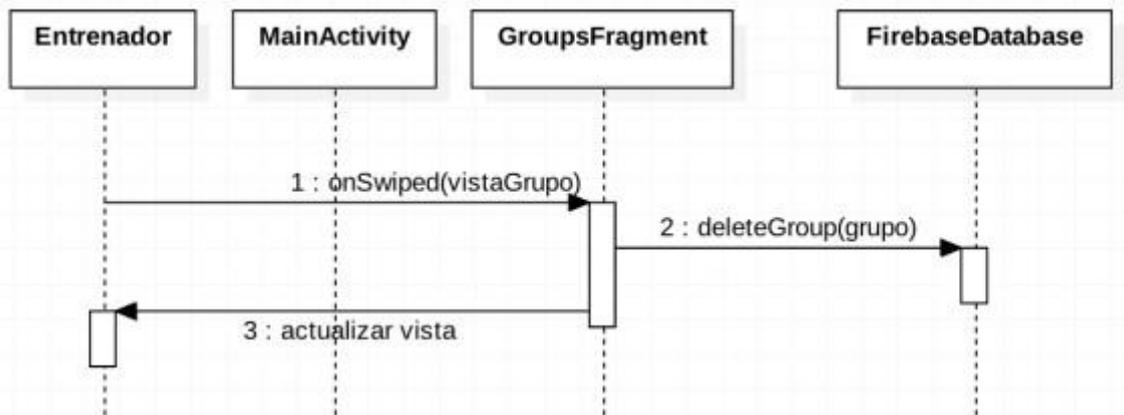


Ilustración 22: Diagrama de secuencia - Eliminar grupo (CU04)

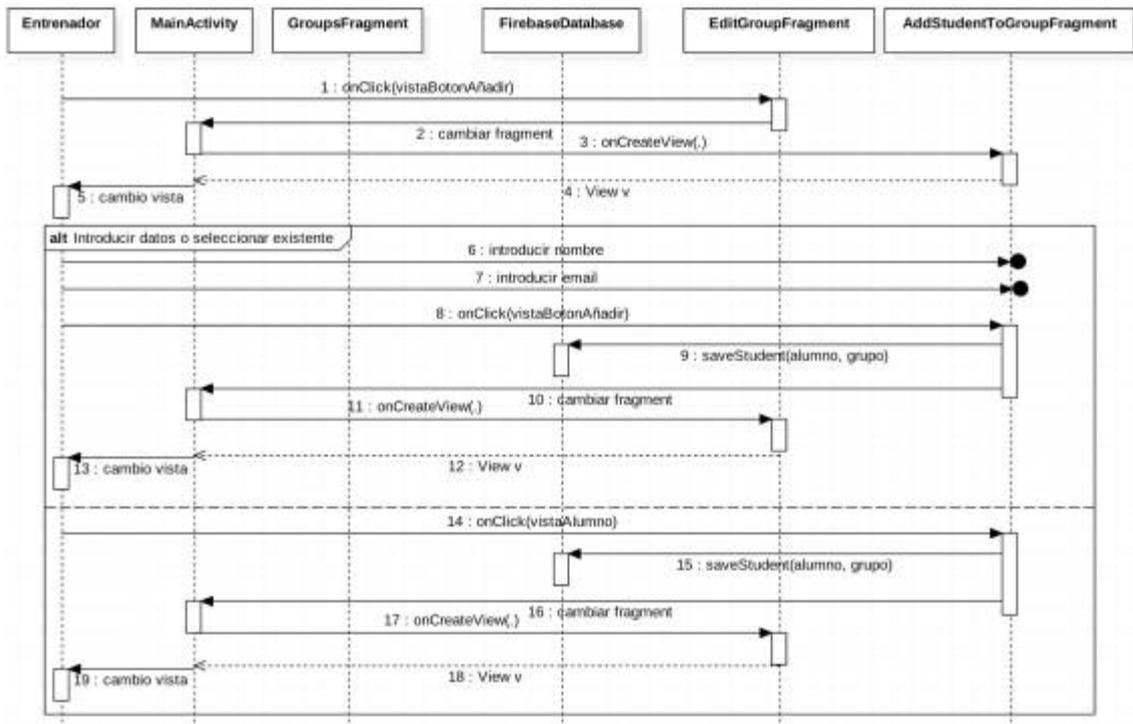


Ilustración 23: Diagrama de secuencia - Añadir alumno al grupo (CU04)

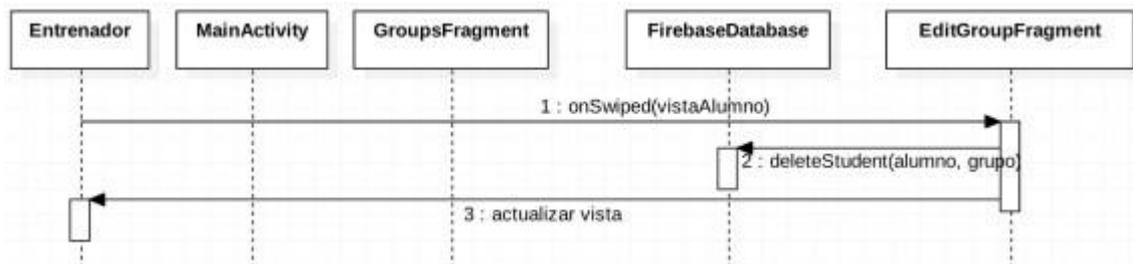


Ilustración 24: Diagrama de secuencia - Eliminar alumno del grupo (CU04)

El caso de uso CU05 comprende el registro de una sesión de entrenamiento y se representa en el diagrama de secuencia de la ilustración 25.

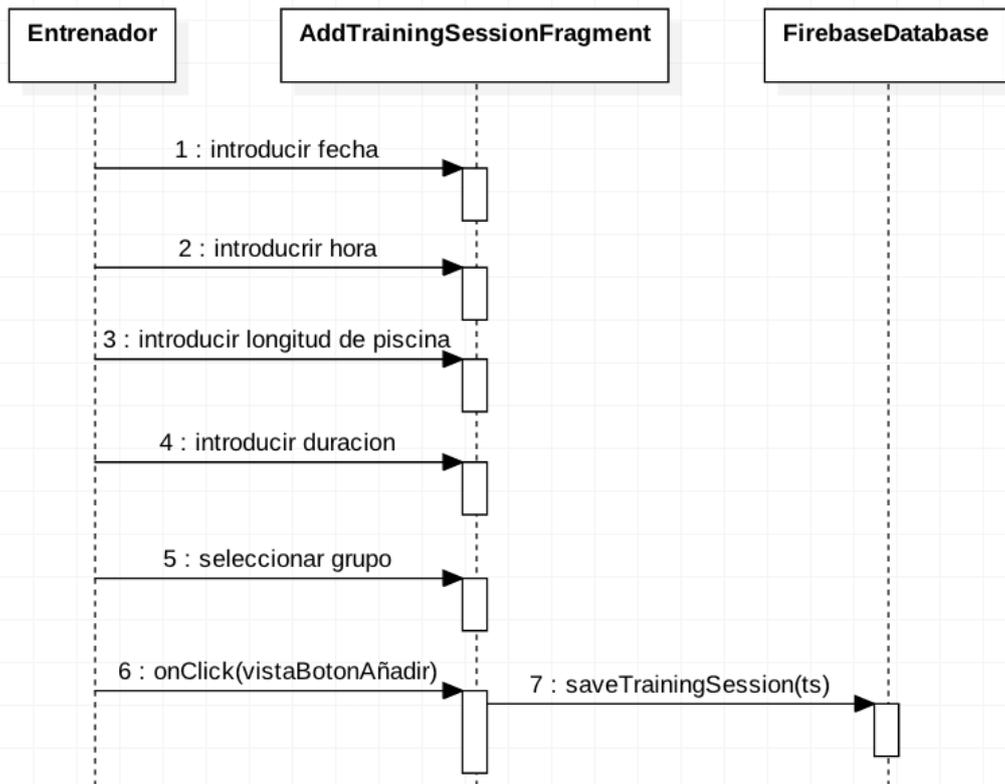


Ilustración 25: Diagrama de secuencia - Registrar sesión de entrenamiento (CU05)

A continuación, se representan los diagramas de secuencia correspondientes al caso de uso que implica el registro de marcas de alumnos (CU06). Se va a representar un diagrama de secuencia que determina cómo empezar la sesión de entrenamiento, el siguiente nos indica cómo registrar los tiempos y cómo finalizar la sesión mientras está en transcurso. Por último, se mostrará cómo registrar el último tiempo tras haber terminado la duración de la sesión.



Ilustración 26: Diagrama de secuencia - Empezar la sesión de entrenamiento (CU06)

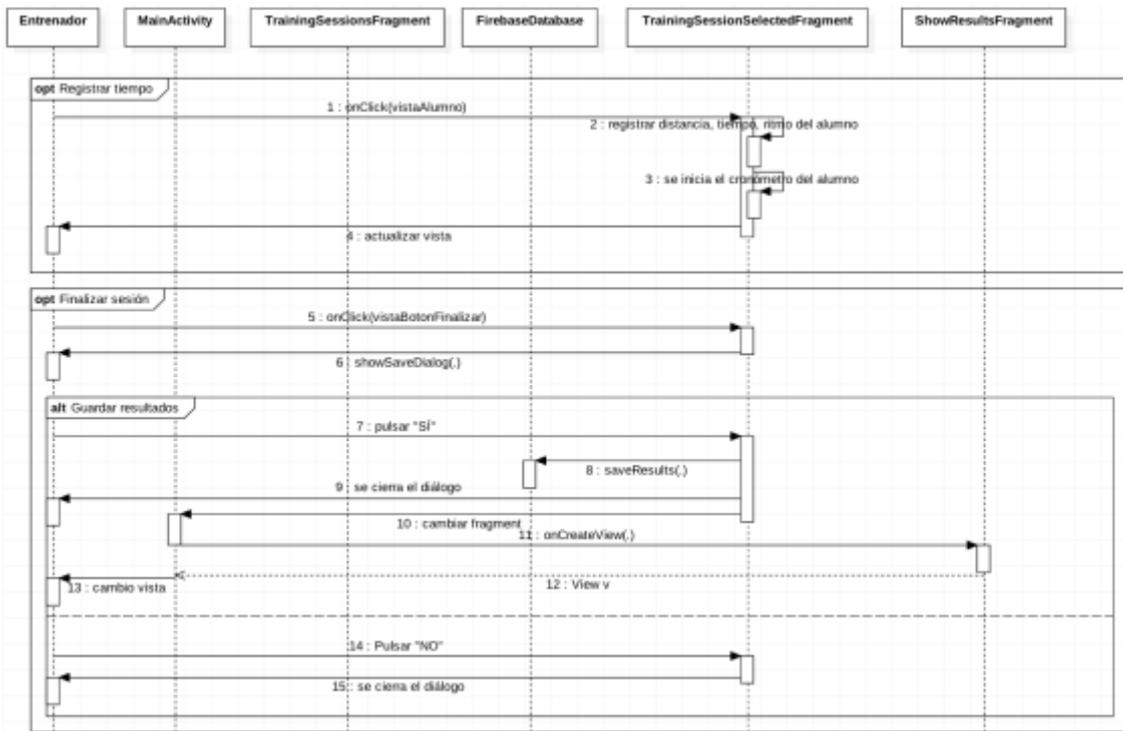


Ilustración 27: Diagrama de secuencia - Acciones durante el transcurso de una sesión de entrenamiento (CU06)

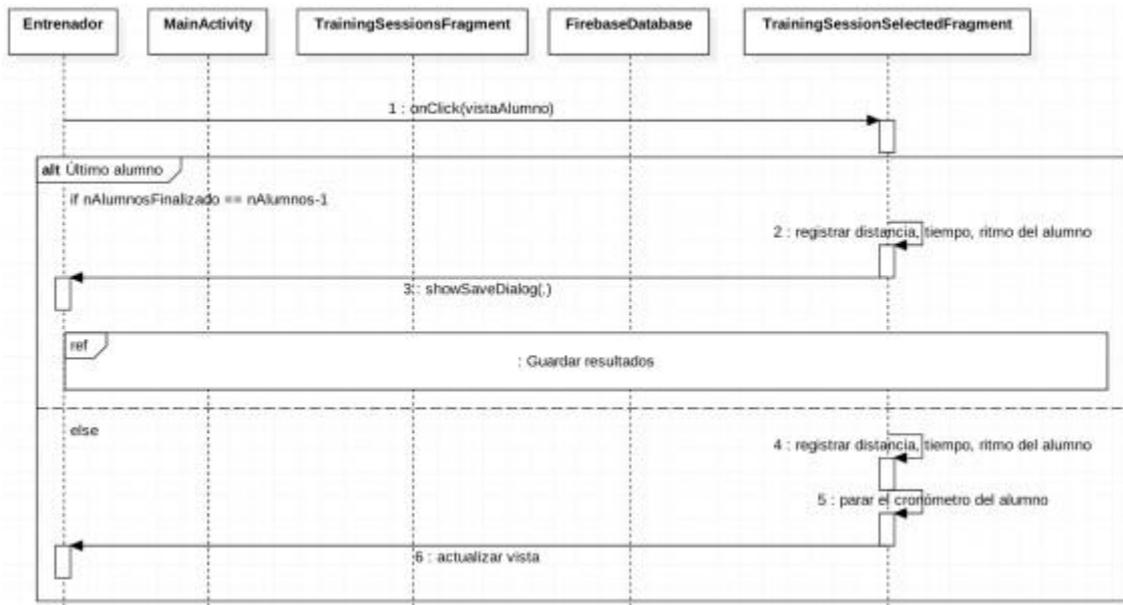


Ilustración 28: Diagrama de secuencia - Registrar el último tiempo en una sesión de entrenamiento (CU06)

4.4 Conclusión

En este capítulo se ha especificado el diseño del sistema. Para ello se ha comenzado con un esquema conceptual y posteriormente se ha ido

desgranando el diseño en la estructura por capas, profundizando en cada una de ellas.

En la capa de presentación se han incluido bocetos de la aplicación móvil y de la aplicación web, además de un diagrama con la interrelación de todas las clases que intervienen en esta capa. Con respecto a la persistencia se ha incluido un esquema de entidad-relación de la base de datos, para más tarde transformarlo una base de datos no relacional, de la cual se han comentado sus ventajas y el modelo de estructura de datos sobre el que se fundamenta. En cuanto a la capa de negocio, se ha mostrado un diagrama de las relaciones entre las clases de la capa persistencia y de lógica, además se han desarrollado una serie de diagramas de secuencia que recogen todos los casos de uso planteados en el capítulo 3. Con el diseño ya establecido se puede proceder a la implementación.

Además, se ha cumplido adecuadamente con la planificación del trabajo establecida en el apartado 1.4 de la introducción. Las tareas se encuentran al día y el desarrollo del proyecto progresa adecuadamente.

5. Implementación

5.1 Introducción

Una vez concretado el diseño del sistema se procede a la exposición de su implementación e implantación.

Con respecto a la implementación se describe con mayor detalle y de manera complementaria al apartado 2.7, las tecnologías y herramientas empleadas en el desarrollo. Además, también se incluyen capturas de pantalla del sistema en funcionamiento, así como las líneas de código más relevantes para el desempeño de las funciones principales de la aplicación.

En el apartado de implantación se describen todos los requisitos necesarios para la correcta instalación y despliegue el sistema.

5.2 Implementación

La descripción de la implementación del sistema se estructura en dos apartados diferenciados. Uno de ellos comprende la implementación de la aplicación móvil y el otro la de la aplicación web.

5.2.1 Aplicación móvil

Para desarrollar la aplicación móvil se utiliza Android Studio y su SDK, el entorno de desarrollo oficial de Android, diseñado por Google y que posee todas las herramientas necesarias para poder elaborar el código, incluyendo un simulador donde poder probar la aplicación a lo largo de su producción. El lenguaje de programación utilizado es Java.

Además, se utilizan los servicios de autenticación y base de datos de Firebase. El primero se emplea para guardar y gestionar los usuarios y el segundo para almacenar toda la información correspondiente a cada entrenador (alumnos, grupos, sesiones de entrenamiento, resultados, etc.)

También se utiliza el módulo Picasso para poder cargar la imagen del usuario cuando se registra con su cuenta de Google y el módulo CircularImageView para mostrarla dentro de un círculo.

A continuación, con el fin de validar la implementación en base al diseño, se presentan unas capturas de pantalla de la aplicación que se recogen en las ilustraciones 29 y 30 y que se pueden comparar con el diseño que aparece en el apartado 4.3.1, en la ilustración 11.

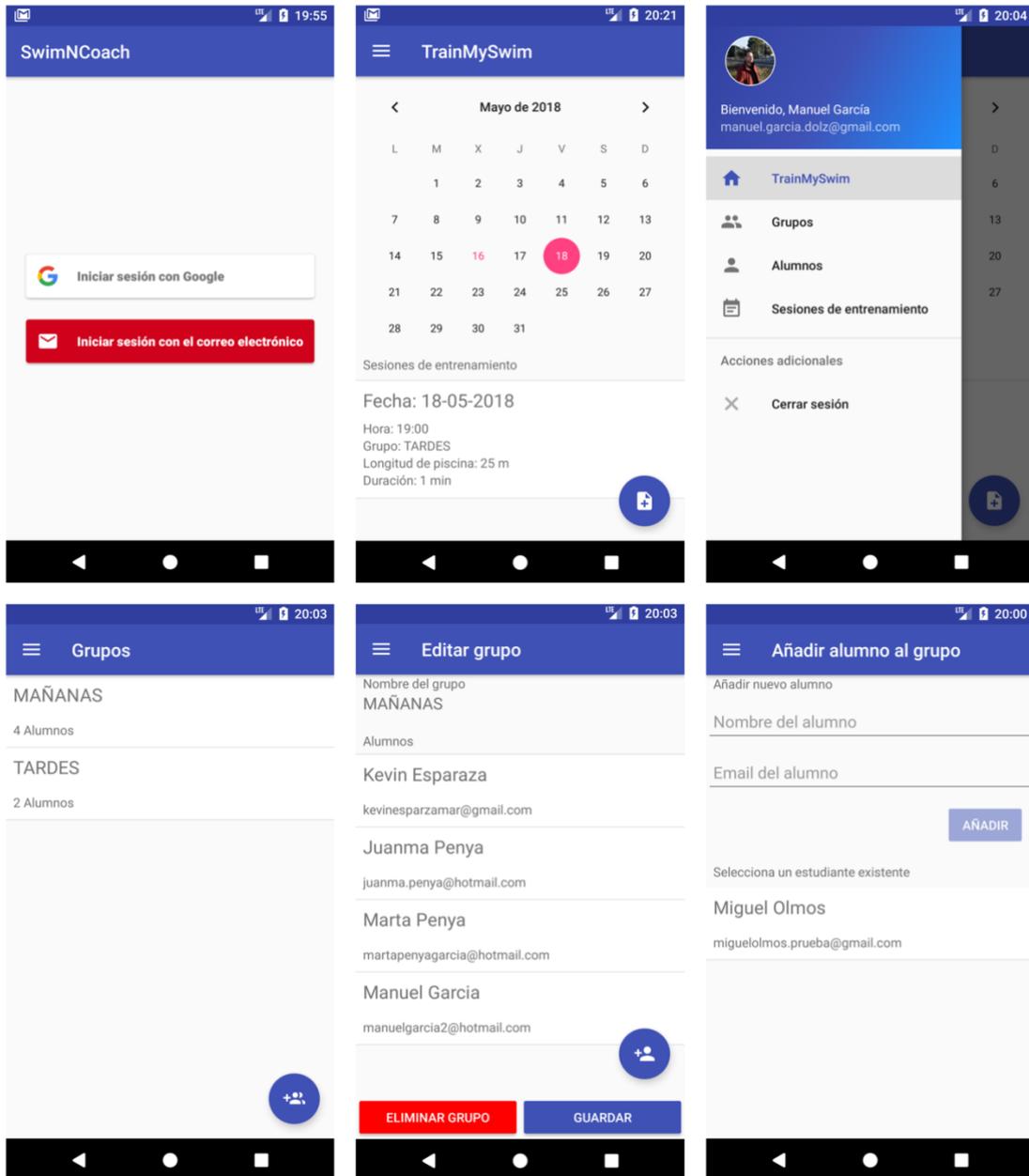


Ilustración 29: Capturas de pantalla de la aplicación móvil 1

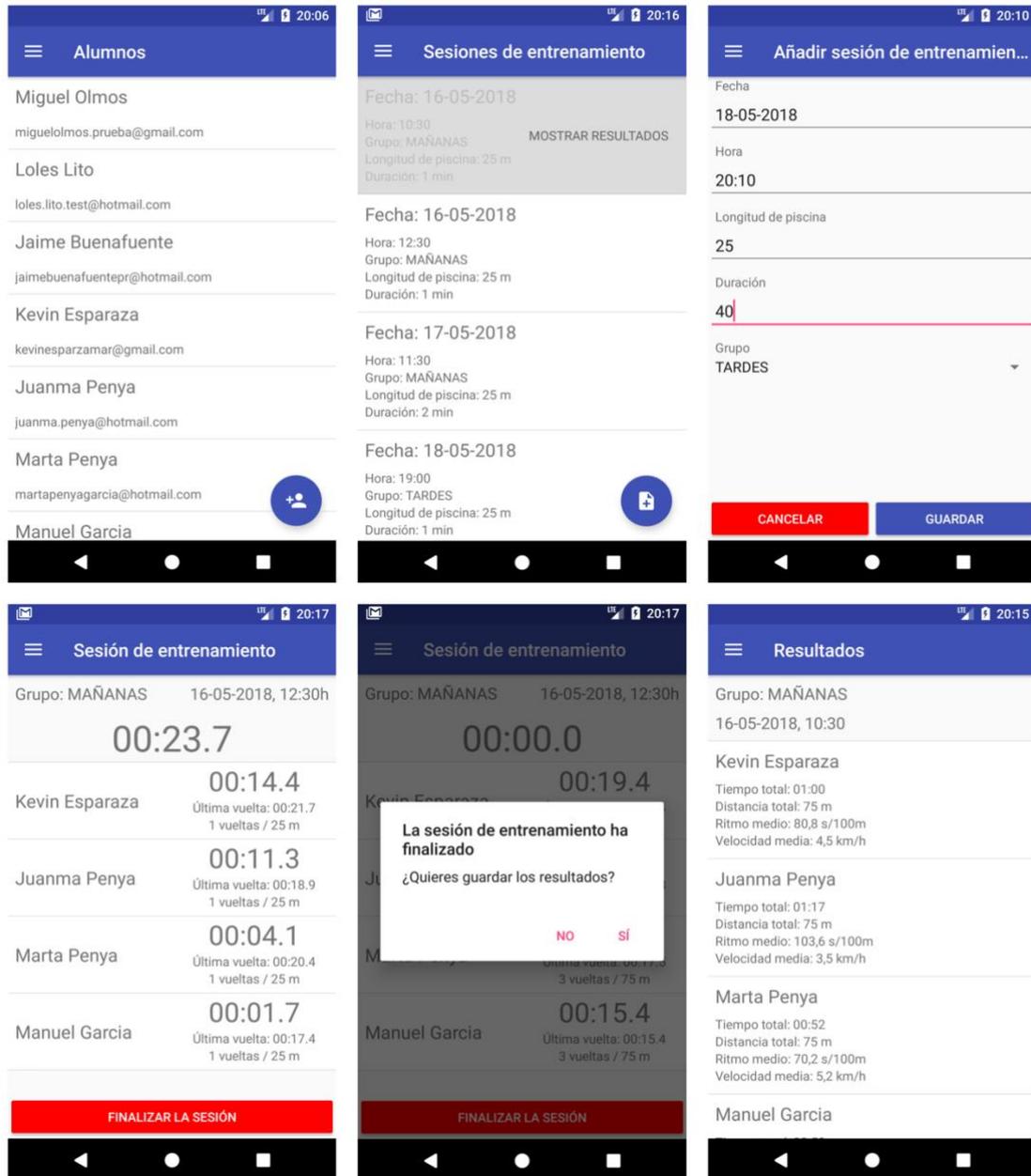


Ilustración 30: Capturas de pantalla de la aplicación móvil 2

Además de mostrar las interfaces gráficas presentes en la aplicación resulta conveniente destacar aquellos fragmentos de código que son relevantes para el adecuado funcionamiento de la aplicación. Por ello se procede a recopilarlos junto con una pequeña explicación de la función que desempeñan.

Dado que se van a introducir pequeños fragmentos de código, se intentará mantener la indentación en la medida de lo posible.

Como se ha utilizado la plataforma Firebase para gestionar la autenticación de los usuarios además de otros servicios como el alojamiento de la web y la base de datos, en primer lugar, hay que añadir Firebase a nuestra aplicación. El proceso es muy intuitivo y la documentación que hay presente en la página oficial es muy detallada. Para poder utilizar la autenticación mediante correo

electrónico/contraseña y Google hay que habilitar los dos métodos de inicio de sesión desde la consola de Firebase como puede verse en la ilustración 30.

Proveedores de inicio de sesión	
Proveedor	Estado
 Correo electrónico/contraseña	Habilitada
 Teléfono	Inhabilitado
 Google	Habilitada

Ilustración 31: Proveedores servicios de inicio de sesión activos en Firebase

Además, es necesario mencionar que en la aplicación se dispone de una clase Application donde se encuentran las referencias a la base de datos y la autenticación de Firebase y son accesibles de manera global en la aplicación.

5.2.1.1 Autenticación de usuarios

Una vez Firebase está integrado en nuestra aplicación, nos aseguramos de poder tener acceso a internet añadiendo en el AndroidManifest.xml

```
<manifest ...>  
    <uses-permission android:name="android.permission.INTERNET"/>  
</manifest>
```

Código 1: Permiso de acceso a internet

En la actividad que se encarga del inicio de sesión se incluye el código 2, donde se obtienen la referencia de la autenticación de firebase y se llama al método doLogin() que se encarga de llevar a cabo la autenticación.

```
private static final int RC_SIGN_IN = 123;  
public static final String USER_PREFERENCES = "userPreferences";  
private FirebaseAuth auth;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
auth = ((Aplicacion) getApplicationContext()).getAuth();
doLogin();
}

```

Código 2: Inicializar las variables de la actividad de inicio y llamada al método que inicia la sesión

El método doLogin() que se incluye en el código 3 se encarga de comprobar que el usuario esté correctamente autenticado. Si es así guarda sus datos y los almacena mediante el método saveUser(FirebaseUser user) en la base de datos de Firebase. Después lanza la actividad principal.

```

private void doLogin() {
    FirebaseUser currentUser = auth.getCurrentUser();
    if (currentUser != null) {
        ((App) getApplicationContext()).initReferences();
        saveUser(currentUser);

        //...

        //Launch MainActivity
        Intent i = new Intent(this, MainActivity.class);
        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP
            | Intent.FLAG_ACTIVITY_NEW_TASK
            | Intent.FLAG_ACTIVITY_CLEAR_TASK); startActivity(i);

    } else {
        signInOptions();
    }
}

```

Código 3: Autenticación de usuario en la aplicación móvil

En el código 4 se incluye el método saveUser(FirebaseUser user) que guarda los datos de un usuario entrenador en la base de datos. Con este ejemplo podemos ver cómo se escribe en la base de datos de Firebase. Para introducir

el resto de datos que se necesiten durante la utilización de la aplicación la metodología es la misma.

```
private void saveUser(final FirebaseUser user) {  
    DatabaseReference coachReference=((App)  
getApplicationContext()).getCoachesReference().child(user.getUid());  
    coachReference.setValue(new Coach(user.getDisplayName(),  
user.getEmail()));  
}
```

Código 4: Guardar un entrenador en la base de datos

En el código 5 se muestra el método signInOptions, que se llama cuando el usuario no está autenticado, y se le muestra las dos opciones de inicio de sesión o registro.

```
private void signInOptions() {  
    startActivityForResult(AuthUI.getInstance()  
        .createSignInIntentBuilder().setProviders(Arrays.asList(  
            new  
AuthUI.IdpConfig.Builder(AuthUI.EMAIL_PROVIDER).build(),  
            new  
AuthUI.IdpConfig.Builder(AuthUI.GOOGLE_PROVIDER).build()))  
        .setIsSmartLockEnabled(true).build(), RC_SIGN_IN);  
}
```

Código 5: Opciones de inicio de sesión y registro

5.2.1.2 Añadir un alumno

Aunque viendo el ejemplo anterior donde se guarda un usuario en la base de datos pueda parecer que añadir un alumno sea una tarea sencilla, se ha decidido mostrar este caso ya que tiene algo de complejidad.

Si se pretende que los alumnos que añade el entrenador puedan consultar sus estadísticas de manera autónoma, es necesario crearlos como usuarios que puedan igualmente autenticarse para que solo ellos puedan ver sus datos. A la hora de crear nuevos usuarios, Firebase complica un poco la tarea ya que un usuario no puede crear otro sin cerrar sesión e iniciar sesión como el nuevo

usuario. Pese a ello, se ha podido encontrar la manera de conseguirlo. La solución consiste en tener dos instancias de la clase de autenticación de Firebase `FirebaseAuth`. Una de ellas está en todo momento asociada al entrenador que utiliza la aplicación, y la otra se utiliza para crear los usuarios, iniciando sesión y cerrando sesión al momento una vez el nuevo usuario ha sido creado.

Mientras que la primera instancia de `FirebaseAuth` `auth`, asociada al entrenador, se inicializa como se indica en el código 5, también puede verse en el mismo fragmento cómo se inicializa la segunda instancia `auth2` auxiliar:

```
private FirebaseAuth auth;
auth = FirebaseAuth.getInstance();

private FirebaseAuth auth2;
FirebaseOptions firebaseOptions = new FirebaseOptions.Builder()
    .setDatabaseUrl("https://siwmncoach.firebaseio.com/")
    .setApiKey("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXX")
    .setApplicationId("siwmncoach ").build();

FirebaseApp myApp = FirebaseApp.initializeApp(getApplicationContext(),
firebaseOptions, "SwimNCoach");
auth2 = FirebaseAuth.getInstance(myApp);
```

Código 6: Instanciación de las variables de la clase de autenticación de Firebase

En el código 6 puede observarse cómo se utiliza `auth2` para crear un nuevo alumno. Concretamente este método se ejecuta cuando se pulsa el botón de 'añadir' alumno. Se procede a utilizar `auth2` para crear el usuario utilizando el email y una contraseña generada de manera aleatoria. Se aprovecha la sesión iniciada para enviar un email al nuevo alumno para que establezca su propia contraseña y posteriormente se cierra la sesión y se guarda el alumno en la base de datos.

```
auth2.createUserWithEmailAndPassword(newStudentEmail,
generatePass()).addOnCompleteListener(getActivity(), new
OnCompleteListener<AuthResult>() {
```

```

@Override
public void onComplete(@NonNull Task<AuthResult> task) {

    if (!task.isSuccessful()) {
        String exception = task.getException().toString();
        Toast.makeText(getActivity(), R.string.email_fail,
Toast.LENGTH_SHORT).show();
    }
    else
    {
        String newStudentUID = auth2.getCurrentUser().getUid();
        Student newStudent = new Student(newStudentUID,
newStudentName, newStudentEmail);

        auth2.sendPasswordResetEmail(newStudentEmail);
        auth2.signOut();
        saveStudent(newStudent);
        Snackbar snackbar = Snackbar.make(v, R.string.student_added,
Snackbar.LENGTH_LONG);
        snackbar.show();

        //Go back to the latest fragment
        getFragmentManager().popBackStackImmediate();

    }

}

});

```

Código 7: Creación de nuevo alumno como usuario con autenticación

5.2.1.3 Registro de tiempos de los alumnos durante el transcurso de una sesión de entrenamiento en tiempo real

Por último cabría destacar el funcionamiento del fragmento de código encargado de que el usuario pueda registrar en tiempo real los tiempos de los alumnos a lo largo del transcurso de una sesión de entrenamiento. Se ha simplificado para poder mostrar el funcionamiento completo de manera intuitiva.

Cada alumno participante en la sesión tiene un cronómetro local que reacciona de manera independiente ante las pulsaciones del usuario sobre su vista. Pero todos ellos comparten una cuenta atrás global, lo que sirve para saber cuándo ha terminado el tiempo que se estableció para la sesión.

Como se muestra en el código 8, cuando el entrenador pulsa sobre la vista de un alumno:

- Si la cuenta atrás global no se ha iniciado, se inicia y se vuelve visible el botón FINALIZAR SESIÓN.
- Se guardan a nivel local los datos acumulados en el último tramo.
- Si la cuenta atrás global no ha terminado, el cronómetro del alumno se reinicia.
- Si la cuenta atrás ha terminado y ya no quedan más alumnos por llegar, se pide confirmación mediante un diálogo para guardar todos los resultados. Si el entrenador confirma la acción, los resultados de la sesión se guardan en la base de datos.

```
holder.itemView.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        if(!globalTimerInit){
```

```
            globaltimer.start();
```

```
            globalTimerInit = true;
```

```
            endSessionButton.setVisibility(View.VISIBLE);
```

```
            endSessionButton.setEnabled(true);
```

```
        }
```

```
        storeLocalResults();
```

```

    if(!globalTimerEnded){
        localTimer.cancel();
        localTimer.start();
    }

    else {
        nTimersFinished++;

        if(nTimersFinished == items.size()){
            showSaveDialog();
        }
    }
}
});

```

Código 8: Registro de tiempos de los usuarios durante el transcurso de una sesión de entrenamiento

El entrenador también puede pulsar directamente al botón de FINALIZAR SESIÓN, y se le abriría el diálogo para confirmar que desea guardar los resultados, como se especifica en el código 9. Ante una respuesta positiva por parte del usuario, los resultados se guardan en la base de datos.

```

    endSessionButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            showSaveDialog();
        }
    });
}

```

Código 9: Botón para finalizar sesión de entrenamiento

5.2.2 Aplicación web

Para el desarrollo de la aplicación web se ha utilizado el editor de texto Sublime. Además, se emplea el servicio de Firebase Hosting para alojar el código y la misma base de datos de Firebase que en la aplicación móvil de la que puede leer los datos referentes a las sesiones de entrenamiento de los alumnos. También se emplea Google Charts para mostrar las estadísticas y JQuery para modificar la estructura de la web de manera dinámica. Se utilizan los lenguajes de programación JavaScript y HTML.

En el caso de la aplicación web también se pretende validar la implementación en referencia al diseño que se establece en el capítulo en las ilustraciones 12-15. Por ello se presenta en las ilustraciones 32-38 las capturas de pantalla del resultado de la aplicación web tras la implementación.

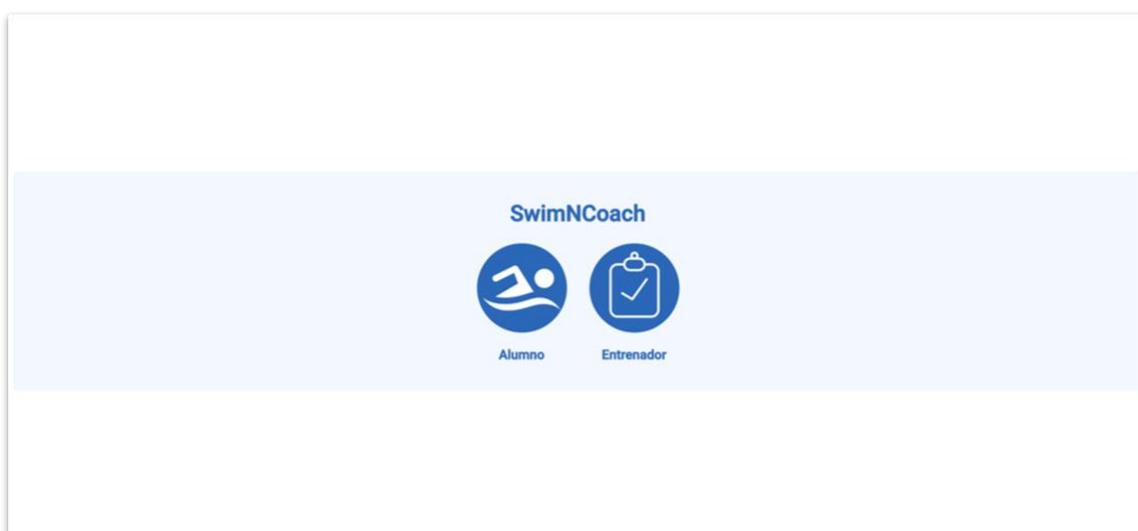


Ilustración 32: Captura de pantalla de la aplicación web - elegir rol

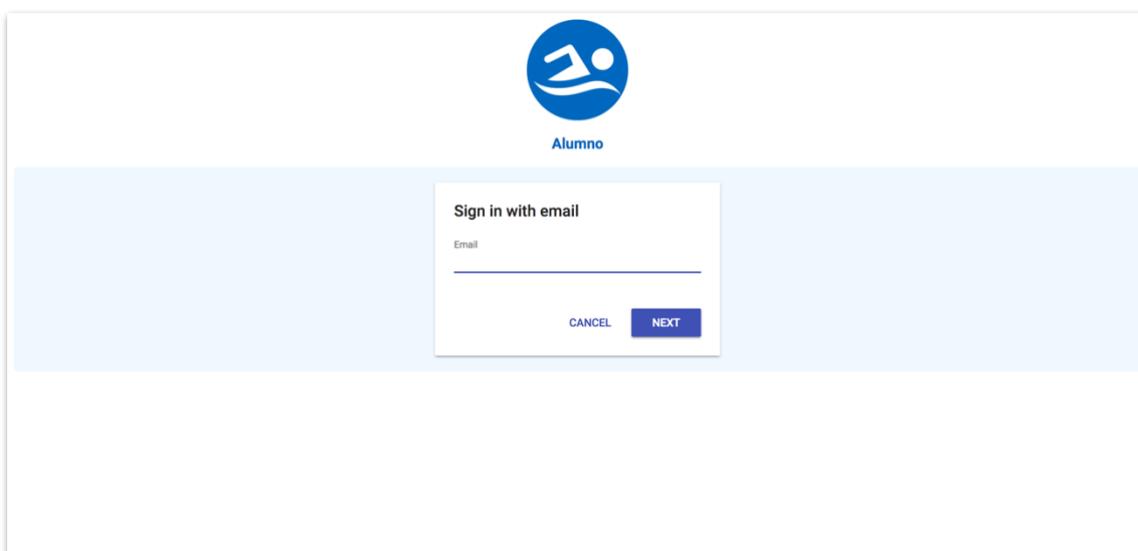


Ilustración 33: Captura de pantalla de la aplicación web - inicio de sesión del alumno

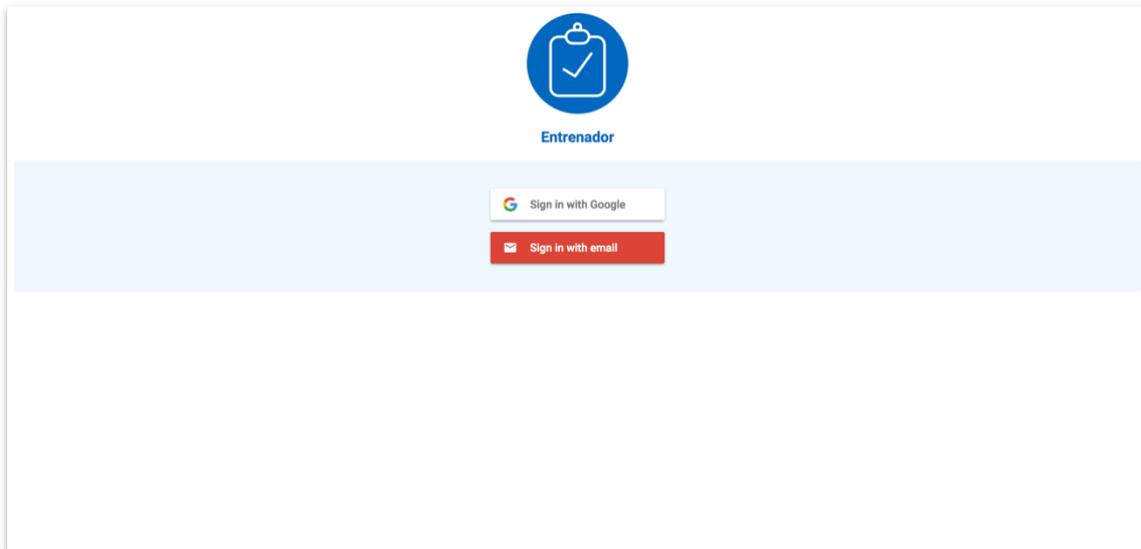


Ilustración 34: Captura de pantalla de la aplicación web - inicio de sesión del entrenador

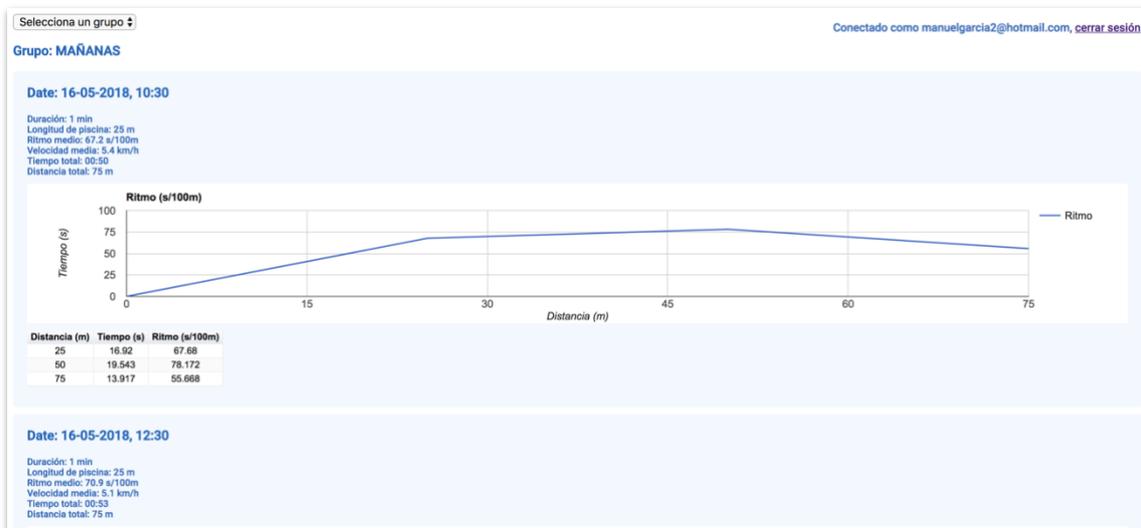


Ilustración 35: Captura de pantalla de la aplicación web - estadísticas del alumno 1



Ilustración 36: Captura de pantalla de la aplicación web - estadísticas del alumno 2



Ilustración 37: Captura de pantalla de la aplicación web - estadísticas del entrenador 1



Ilustración 38: Captura de pantalla de la aplicación web - estadísticas del entrenador 2

A continuación, se presentan las líneas de código más relevantes de la aplicación web. Concretamente, lo más interesante es la generación de los gráficos con las estadísticas utilizando la herramienta Google Charts, ya que la lectura de la base de datos se hace de manera muy similar a la interacción con ella que se ha mostrado en la aplicación web. Hay un *listener* escuchando sobre una referencia a la base de datos y cuando hay cambios (se añaden, modifican o eliminar datos) se activa. La primera vez que se activa lee todos los datos a los que se pueden acceder desde la referencia, por lo que se puede hacer una única lectura inicial de la base de datos y mostrar los resultados.

5.2.2.1 Dibujar gráfica del ritmo de una sesión con varios alumnos

Ya que se va a mostrar cómo utilizar Google Charts y JavaScript para mostrar una gráfica, se presenta aquella que tiene un poco más de complejidad, y es la que representa en un gráfico de líneas el ritmo de varios alumnos durante una sesión de entrenamiento, accesible para el entrenador. Esto se recoge en los códigos 10-12.

Es necesario crear una tabla de datos y añadir tantas columnas como estudiantes hay. Después rellenamos las filas con los valores que se computan cogiendo un resultado de cada alumno y añadiéndole al principio el número de metros recorridos en cada registro.

```
// Create the data table.
var data = new google.visualization.DataTable();

data.addColumn('number', 'x');
for(var i=0; i<results.length; i++){
```

```

        data.addColumn('number', results[i].studentName);
    }

    var metersAcumulated = poolSize;

    rows = computeRows();

    for(var i=0; i<rows.length;i++){
        var row2add = rows[i];
        row2add.unshift(metersAcumulated);
        data.addRow([row2add]);
        metersAcumulated += poolSize;
    }

```

Código 10: Crear table de datos y añadir filas y columnas para mostrar en el gráfico de ritmo

Por último, establecemos las opciones del gráfico donde podemos indicar, por ejemplo, el nombre de los ejes y dibujamos la gráfica en el contenedor deseado.

```

// Set chart options
var options = {
    title: "Ritmo (s/100m)",
    hAxis: {
        title: 'Distancia (m)'
    },
    vAxis: {
        title: 'Tiempo (s)'
    },
    interpolateNulls: true
};

```

Código 11: Especificar las opciones del gráfico de ritmo

```

var chart = new google.visualization.LineChart(
document.getElementById((chartID+counter)));
chart.draw(data, options);

```

Código 12: Dibujar gráfico de líneas del ritmo en el contenedor HTML especificado

5.3 Implantación

Para poder implantar el sistema es necesario que dependiendo del rol que tenga el usuario instale y utilice unas herramientas u otras.

Los entrenadores tendrán que instalar la aplicación Android en su dispositivo. Para ello es necesario la APK (*Application Package File*). Como de momento no está disponible en la tienda de aplicaciones de Android, será necesario activar en la configuración del dispositivo la opción de poder instalar aplicaciones de origen desconocido. Una se haya hecho este proceso la aplicación podrá instalarse y utilizarse en el dispositivo.

Los alumnos necesitarán un navegador para poder acceder a la web para ver los resultados de sus sesiones de entrenamiento. Además, la web está alojada en el servidor de Firebase por lo que se puede acceder directamente y está disponible en todo momento.

5.4 Conclusión

Para llevar a cabo la implementación del sistema se han utilizado una serie de herramientas que han facilitado la labor. La plataforma de desarrollo Firebase ha resultado muy útil ya que incluye varios servicios que se han adaptado perfectamente al ámbito del proyecto. Elegir correctamente las herramientas que se van a utilizar para desarrollar el sistema resulta de gran importancia ya que de otra manera podría haber habido complicaciones.

A lo largo de este capítulo se ha especificado con detalle la implementación e implantación del sistema. La primera se ha dividido en dos partes, considerando primero la aplicación móvil y en segundo lugar la aplicación web. Para ambas se ha podido validar el diseño y se ha mostrado capturas del sistema en funcionamiento, así como fragmentos de código que representan acciones relevantes.

En cuanto a la implantación se ha indicado lo que es necesario para que cada rol de usuario pueda acceder y utilizar el sistema.

Sin embargo, la planificación del proyecto no se ha ajustado por completo a la especificación del plan de trabajo inicial. Se ha logrado implementar la aplicación móvil y web pero la dedicación ha sido superior a la esperada. Por ello, de manera externa a la planificación se sometió la aplicación global a un conjunto de pruebas y tests específicos para validar los puntos clave del sistema.

6. Conclusiones

6.1 Trabajo realizado

El desarrollo del proyecto se ha abordado poco a poco desde una perspectiva inicial de análisis, para después pasar a especificar cuáles eran los requisitos que debía poseer el sistema y raíz de ello, se ha podido establecer un diseño y posteriormente una implementación del sistema.

Concretamente, el análisis, era necesario para conocer el entorno sobre el que se fundamentaba el desarrollo del proyecto con el fin de concretar qué características debería incluir la aplicación a desarrollar. Se realizó un recorrido por las distintas aplicaciones de natación que se encuentran disponibles para las plataformas móviles existentes. Algunas de ellas ofrecían funcionalidades interesantes pero la mayoría no eran completas y no estaban orientadas al entrenador. Con ello se consiguió dar forma a las características que sí que tenían que estar presentes en el sistema a desarrollar.

Conociendo estas características y mediante la utilización de un diagrama de casos de uso se concretaron los requisitos específicos que debía satisfacer el sistema y que había que implementar.

Posteriormente, se especificó el diseño del sistema. Para ello se dispuso de numerosos esquemas y diagramas que permitieron establecer una estructura por capas, profundizando en cada una de ellas.

Una vez formulado el diseño se ha llevado a cabo la implementación del sistema utilizando una serie de herramientas que han facilitado la labor. La plataforma de desarrollo Firebase ha resultado muy útil ya que incluye varios servicios que se han adaptado perfectamente al ámbito del proyecto. Se han especificado los detalles de la implementación y se ha podido validar el diseño establecido anteriormente. Como resultado, se ha cumplido con los objetivos de desarrollo establecidos en la planificación inicial obteniendo un producto completo.

6.2 Dificultades y soluciones

Se van a describir las dificultades más importantes encontradas durante la realización del proyecto junto con la solución aportada.

En el capítulo dos, cuando se analizaron los sistemas similares, se encontraron demasiados y la tarea de analizarlos todos llevaba bastante tiempo. Muchos de ellos incluyen más deportes además de la natación y de entre los que son exclusivamente dedicados a la natación la mayoría tiene una orientación dedicada al nadador, no al entrenador. Por ello, finalmente se optó por mostrar los que eran realmente relevantes para el análisis: aquellos que daban

prioridad a la natación y tenían objetivos comunes al sistema que se pretendía desarrollar.

Referente al capítulo cuatro de diseño, en un primer momento resultaba muy complicada la organización y nexos entre la base de datos, la autenticación y comprobación de usuarios y el alojamiento de la web. Además, dichos servicios debían ser accesibles desde la aplicación web y la aplicación móvil al mismo tiempo. Al final se optó por utilizar Firebase, la plataforma de desarrollo de Google que integra todos los servicios mencionados, tiene una buena documentación y es sencilla de utilizar.

En cuanto al diseño de la base de datos, al decidirse a utilizar una base de datos NoSQL, la manera de organizar los datos es muy diferente al modelo clásico de base de datos relacional. Por ello fue necesario el aprendizaje sobre este tipo de base de datos y la eficiencia de su estructura. Finalmente se pudo organizar los datos de manera óptima para que las llamadas de lectura a la base de datos resulten lo menos costosas posibles.

Con respecto a la implementación, se ha tenido que trabajar con tecnologías nuevas y lenguajes de programación de los cuáles no se tenían conocimientos muy avanzados, como por ejemplo JavaScript y la integración de Google Charts para la web o la plataforma Firebase. Ello conlleva un aprendizaje progresivo que en el fondo es muy positivo. Además, el tiempo empleado en este apartado ha excedido el de la planificación, por lo que finalmente el proyecto ha supuesto mayor dedicación a la esperada.

Pese a las dificultades encontradas, no han resultado ser un problema grave además se ha aprendido mucho con la elaboración del proyecto. El desarrollo y evolución constante está siempre presente en la informática.

6.3 Aportaciones

La principal aportación es el sistema que se ha desarrollado en sí, compuesto por una aplicación móvil y una aplicación web, cubriendo así el objetivo principal del proyecto. El sistema supone una herramienta que facilita las tareas que realizan los entrenadores de natación. Supone un avance en la estructura y análisis de sus entrenamientos, así como la gestión de sus alumnos y grupos. De igual manera es útil para los alumnos que pueden considerar sus resultados para obtener una visión de su desempeño y proponerse nuevos retos y objetivos.

El análisis del capítulo dos sobre los sistemas similares puede ser de interés para el lector. De igual manera la especificación de requisitos y el diseño del sistema pueden servir además como ayuda o referencia para el desarrollo de otros proyectos similares.

6.4 Trabajo futuro

El sistema puede ser ampliable con una serie de adiciones que extiendan sus capacidades y su difusión.

En primer lugar, la aplicación actualmente solo está disponible para Android y pese a ser el sistema operativo móvil más extendido en todo el mundo, también sería conveniente desarrollarla en iOS, ya que millones de usuarios utilizan igualmente este sistema.

La interfaz gráfica tanto de la web como de la aplicación móvil pueden mejorarse ya que suponen una parte importante de interacción con el usuario. Sobre todo dado que las estadísticas con gráficos son muy llamativas en la mayoría de aplicaciones relacionadas con el deporte. Además, sería conveniente incluir los gráficos de los resultados también en la aplicación móvil. Así el entrenador podría ver las estadísticas de las sesiones de una manera más intuitiva.

7. Bibliografía

Developer.android.com. (2018). Android Developers. [online] Disponible en: <https://developer.android.com> [Acceso 3 mayo 2018].

Firebase. (2017). Documentation | Firebase. [online] Disponible en: <https://firebase.google.com/docs/> [Acceso 10 mayo 2018].

Bibliografía: Google Developers. (2018). Charts | Google Developers. [online] Disponible en: <https://developers.google.com/chart/> [Acceso 17 mayo 2018].

Square.github.io. (2018). Picasso. [online] Disponible en: <http://square.github.io/picasso> [Acceso 10 mayo 2018].

GRIFFITHS, D. (2018). HEAD FIRST ANDROID DEVELOPMENT. [S.I.]: O'REILLY MEDIA.

Martin, R. (2004). UML para programadores en Java. Madrid: Prentice Hall.

Arias, M. (n.d.). Guía de HTML5, CSS3 y JavaScript.