



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación para reserva de actividades y horarios de un gimnasio mediante una app móvil

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Zurriaga Fito, Miguel Ángel

Tutor: Andrés Martínez, David de

2021 / 2022

Resumen

De todos es sabido que esta pandemia ha dejado a muchos locales y negocios cerrados, algunos de los perjudicados no paran de ver como la normativa cambia y cada vez les permiten abrir con cierto aforo, ciertas horas y luego los vuelven a cerrar. Como esto ya es un verdadero quebradero de cabeza hemos pensado ponérselo más fácil y aportar nuestro granito de arena. Para ello hemos tenido en mente los gimnasios. Después de que la gente pasara tiempo en sus casas sin hacer el mínimo ejercicio para mantenerse saludable, ha habido una avalancha de altas en los gimnasios para volver a esa forma en la que nos sentíamos bien con nosotros mismos. Esto hace que sea necesario un cierto control de acceso y planificación del aforo.

Para ello se creará una aplicación para *smartphones* ya que prácticamente todo el mundo lleva uno en el bolsillo. Con este desarrollo encontraremos una forma fácil de tener un buen control de acceso para los clientes de tu gimnasio y todo aquello que puedas ofrecer como clases, salas, horarios, etc... La solución que se ofrece es dicha aplicación utilizada mediante tecnología web desarrollada para Android que posteriormente podrá, con el mismo código, exportarse para otras plataformas, y una API para el acceso a los datos.

Palabras clave: App móvil; smartphone; Android; HTML; CSS; JavaScript; JS; aplicación híbrida; API; Apache Cordova



Abstract

Everyone knows that this pandemic has left many stores and businesses closed, some of those affected do not stop seeing how the regulations change and each time they allow them to open with a certain capacity, certain hours and then close them again. As this is already a real headache, we have thought to make it easier for you and contribute our grain of sand. For this, we thought about gyms. After people spent time at home without doing even the slightest bit of exercise to stay healthy, there has been an avalanche of joining gyms to get back to that way we felt good about ourselves. That requires some access control and capability planning.

For this, an application for *smartphones* will be created since practically everyone carries one in their pocket. With this development we will find an easy way to have a good access control for the clients of your gym and everything that you can offer such as classes, rooms, schedules, etc... The solution that is offered is said application used through web technology developed for Android that can later, with the same code, be exported to other platforms, and an API for data access. The solution that is offered is said application, used through web technology developed for Android that can later, with the same code, be exported to other platforms, and an API for data access.

Keywords: mobile app; smartphone; Android; HTML; CSS; JavaScript; JS; hybrid application; APIs; Apache Cordova

Tabla de contenidos

1.	Introducción	9
1.1	Motivación	10
1.2	Objetivos	10
1.3	Metodología	11
1.4	Estructura	12
2.	Estado del arte	13
2.1	Activa Club [7]	13
2.2	Centro deportivo Moncada [8]	15
2.3	Propuesta	17
3.	Especificación	19
3.1	Arquitectura	19
3.2	Casos de uso	20
3.3	Funcionalidades de los casos de uso	21
4.	Diseño de la solución	24
4.1	Arquitectura del sistema	24
4.2	Diseño de la capa de presentación	25
4.3	Diseño de la persistencia de datos	28
4.4	Tecnología utilizada	30
5.	Implementación	34
5.1	Cliente	34
5.2	Servidor	36
6.	Resultados y presentación	49
7.	Conclusiones	58
8.	Ampliaciones y futuros trabajos	59
9.	Bibliografía	61
10.	Anexo: ODS	63

Índice de figuras

Figura 1: Captura de pantalla app ActivaClub tabla reserva.....	15
Figura 2: Captura de la elección de las clases.....	16
Figura 3: Caso de uso de usuario no registrado.....	20
Figura 4: Diagrama de casos de uso.....	21
Figura 5: Cliente - servidor.....	24
Figura 6: Boceto inicio de sesión.....	25
Figura 7: Boceto de la ventana menú.....	26
Figura 8: Boceto de la ventana Mi cuenta.....	26
Figura 9: Boceto de la ventana Mis reservas.....	27
Figura 10: Boceto de la ventana Mis puntos.....	28
Figura 11: UML [10].....	29
Figura 12: Estructura de la parte cliente.....	34
Figura 13: Estructura de la parte servidor.....	36
Figura 14: Proyectos en GitHub [18].....	39
Figura 15: Heroku [19].....	39
Figura 16: Visual Studio Code [20].....	41
Figura 17: SDK Android.....	42
Figura 18: AVD Manager.....	43
Figura 19: Apache Cordova.....	43
Figura 20: Flujo de autenticación con OAuth 2 [21].....	44
Figura 21: Token obtenido.....	45
Figura 22: Comando para exportar la aplicación y firmar.....	46
Figura 23: Build.json.....	46
Figura 24: Instalación de aplicaciones desconocidas versión inferior a la 10.....	47
Figura 25: Instalación de aplicaciones desconocidas versión superior a la 10.....	47
Figura 26: Inicio de sesión del cliente.....	50
Figura 27: Ventana principal de la aplicación.....	51
Figura 28: Mi cuenta y mis datos.....	52
Figura 29: Mis reservas y la opción para reservar.....	53
Figura 30: Listado de clases que oferta el gimnasio.....	54
Figura 31: Horario filtrado por clases.....	55
Figura 32: Horario sin filtro de clases.....	56
Figura 33: Mis puntos y listado de recompensas.....	57



Índice de tablas

Tabla 1: Comparativa de aplicaciones	18
Tabla 2: Usuario no autenticado	21
Tabla 3: Ver datos de la cuenta	22
Tabla 4: Ver reservas del usuario	22
Tabla 5: Ver puntos acumulados	22
Tabla 6: Ver aforo en tiempo real	23
Tabla 7: Acceder al gimnasio	23

1. Introducción

Los avances de esta nuestra sociedad sobre todo en el ámbito de la tecnología [1] hacen que las empresas estén obligadas a adoptar nuevas medidas para ir de la mano según la tendencia actual.

En pleno siglo veintiuno es difícil encontrar a alguien que no tenga un ordenador en casa o un móvil en el bolsillo, pero no hablamos de un móvil como los de antes que solo sirven para realizar llamadas o recibir mensajes. Hablamos de los conocidos *smartphones* que nos acompañan haya donde vayamos y que hoy en día incluso nos entra ansiedad si hemos salido de casa sin él. A decir verdad, lo que más podemos encontrar son esos diminutos aparatitos que además de realizar todo aquello que podíamos hacer con los antiguos móviles, es como si dispusiéramos de un pequeño portátil que nos llevamos a todos lados [2] y esto las empresas lo saben.

En la actualidad, si algo es cierto, es que la tecnología nos ayuda en nuestro día a día y es prácticamente imposible no hacer uso de esta. Empezó todo con las páginas webs. Gracias a Google, las personas éramos capaces de buscar cualquier cosa y con ello llegó la globalización de la comunicación. Después, como los *smartphones* empezaron a tomar más presencia empezaron a desarrollar las webs de forma *responsive* para que se pudieran visualizar correctamente y aplicaciones a medida para los sistemas operativos de estos terminales que había en el mercado. Hoy en día, aunque tanto las páginas webs como las aplicaciones móviles coexisten, se prefiere que las personas tengan una experiencia de usuario relacionado directamente con el *smartphone* [3], así que las empresas apuestan por este tipo de negocio.

Por otro lado, estamos pasando por una pandemia. El Covid 19 ha tenido un impacto negativo en la sociedad y mucho más para los negocios. En este caso queremos poner el foco de atención en los gimnasios. Un negocio muy sufrido en esta época con sus salas interiores, acumulación de gente y fuertes restricciones que han ahogado a aquellos que veían como debían realizar cambios en sus instalaciones, en su forma de ejecutar las clases que ofrecían, en horarios que

debían modificar y en todas las restricciones que se encontraban cada muy poco tiempo.

Por todo ello, hemos pensado que debíamos aportar nuestro granito de arena y por esto hemos visto que la necesidad de los gimnasios pasaba por una aplicación móvil para la planificación de todo aquello que podían ofrecer y que el cliente pueda acceder y reservar con la seguridad de que podrá hacerlo sin tener que darse media vuelta una vez llega a la entrada del gimnasio y la persona situada en el mostrador se vea obligada a indicar que no es posible su ingreso en estos momentos.

1.1 Motivación

La realización de este proyecto está motivada por el nicho de mercado que se ha podido comprobar al realizarse ciertas restricciones de horario y ver que muchos de los gimnasios no tienen una buena planificación o control de todo aquello que ofertan. Prácticamente todos tienen una página web, pero no todos disponen de una aplicación para *smartphones* y como ya hemos visto anteriormente en la introducción, los clientes hacen más uso de sus dispositivos móviles que de sus ordenadores, además que es una buena herramienta para poder tener un control en el aforo ya que prácticamente todos los clientes irán al gimnasio con estos en sus bolsillos.

Es bastante común que cuando un gimnasio oferta clases con un horario fijo sí que tengan un plan de acotamiento para el aforo, pero dado que la sala de musculación y el aforo de entrada suelen ser muy elevados, estos no están regulados ni en sus páginas webs ni en sus aplicaciones, de aquellos gimnasios que dispongan de una aplicación dedicada.

1.2 Objetivos

Para este proyecto se ha puesto en marcha una serie de objetivos, dado las necesidades de los gimnasios.

Principalmente, crear una aplicación para *smartphones* que sea de muy fácil utilización e intuitiva para que en un simple vistazo los clientes puedan tener claro las posibles de reservas que pueden realizar, así como las que ya han realizado.

Así mismo, una API para poder conectar con su base de datos donde tengan ya la información de aquello que ofertan y poder mostrarlo a cada cliente que requiera de su consulta sin la necesidad de tener que llamar o realizar dicha consulta de forma externa. Hay que recordar que muchos de los gimnasios ya tienen su página web y pueden disponer de información que no quieren volver a hacer y poder seguir utilizándola. También es posible que ya tengan una aplicación antigua y estén buscando actualizarla. Estos son nuestros dos principales y potenciales clientes.

1.3 Metodología

Una vez tenemos claro los conceptos básicos de los que va a contar nuestra aplicación tenemos que adoptar una metodología que nos sirva para planificar los pasos de nuestro proyecto.

Al principio se valoró utilizar la metodología Scrum [4] pero una característica principal de esta es de realizar entregas parciales conocidas como *Sprint* para la verificación del cliente hasta la entrega final así como reuniones cada poco tiempo y dado que nosotros vamos a presentarle a los gimnasios un prototipo ya hecho para que el cliente nos indique cuáles son sus necesidades y funcionalidades que debe implementar la aplicación, se ha decidido en utilizar la metodología tradicional de modelo en cascada [5].

El modelo en cascada es un procedimiento lineal que se caracteriza en dividir los procesos que se van a desarrollar en las siguientes fases:

- La primera fase es la del **análisis** y consiste en planificar y analizar las necesidades, así como los requisitos para conocer la viabilidad del proyecto.
- Posteriormente pasamos al **diseño** donde formularemos una solución específica en base a las exigencias, tareas y estrategias definidas en la fase anterior.



- La fase de **implementación** incluye la programación del software y la búsqueda de posibles errores, así como las pruebas unitarias.
- A continuación, realizamos la **verificación** por parte del cliente y su explicación en el manejo de la aplicación.
- Finalmente, procedemos a la **entrega** con su respectivo mantenimiento y la posibilidad de mejorar el software.

1.4 Estructura

En primer lugar, en el primer capítulo, se ha hecho una pequeña introducción de qué pretendemos así de las tecnologías que vamos a utilizar.

Posteriormente, en el segundo capítulo, realizamos una pequeña comparativa de las posibles ofertas que ya tenemos en el mercado para poder sacar lo mejor de cada una y quizá sus posibles carencias para utilizarlas a nuestro favor.

A continuación, en el tercer y cuarto capítulo, realizaremos un análisis y diseño de nuestra aplicación haciendo referencia a las necesidades de los gimnasios.

Una vez ya tengamos claro nuestro proyecto, implementaremos una versión de pruebas, como podemos observar en el quinto capítulo, utilizando una aplicación real en Android como ejemplo de lo que podría ver un cliente, así como el despliegue de la versión API en un servidor real para simular la puesta en marcha en los servidores donde ya tenga la infraestructura desplegada el gimnasio y una muestra real de la aplicación en el sexto capítulo.

Finalmente, en el séptimo y octavo capítulo, obtendremos las conclusiones de todo lo mencionado en este documento, así como de los posibles problemas que hayamos encontrado en el proceso de realización del proyecto y los futuros trabajos que se podrían realizar en concordancia al proyecto o las ampliaciones de este.

2. Estado del arte

Primero hay que remarcar que este tipo de proyectos se hace de forma más específica con las necesidades de cada cliente, no existe una solución única, quizá ni si quiera existe una solución mejor o peor que otra, simplemente con distintas funcionalidades y que se adecue mejor a las necesidades de cada gimnasio. Es verdad que podemos encontrar algunas opciones de empresas que ofrecen unas aplicaciones amplias y generalizadas que podrían venirle bien a un gimnasio que se está iniciando [6] ya que incluyen funcionalidades como un sistema de facturación o de administración y que lo único que pueden personalizar, además de la parte visual como el logo, texto o colores, son los horarios de las clases.

Esto hace que el precio sea algo elevado porque el gimnasio puede tener otro sistema para realizar ciertas funciones y que cualquier cambio que no se encuentre en esa aplicación suele elevar el presupuesto.

Por todo esto, nos hemos centrado en los gimnasios que, al no ser nuevos, ya disponen del sistema y base de datos con los clientes y que lo quieren ampliar facilitando a estos una aplicación para su *smartphone* en el que puedan realizar una comunicación más agradable y sencilla.

A continuación, se han puesto dos ejemplos que directamente hemos podido utilizar, de forma que se ha podido comprobar que tanto por su sencillez a la hora de que el cliente elija justo lo que está buscando, como por lo que ofrece el gimnasio, es una buena opción de lo que queríamos y necesitábamos para nuestro proyecto.

2.1 Activa Club [7]

Esta es una de las aplicaciones que se han utilizado para la inspiración de este proyecto y por su alta capacidad en contenidos.

Además de poder incluir tus propios avances en dicha aplicación, se van acumulando unos puntos, tanto por acceso como por las clases a las que se asistan que luego se pueden canjear en forma de premios que el gimnasio activa.

Esto es una funcionalidad interesante si el gimnasio tiene pequeños ganchos que quiera incluir para mantener a sus clientes activos más motivados.

Una vez hemos navegado un poco por lo que nos ofrece esta aplicación, nos damos cuenta de que a pesar de que registra tus entradas en el gimnasio, no hay ningún lugar en el que indique si es posible realizarla o se ha superado el límite de aforo, cuya observación es necesaria realizarla físicamente en el gimnasio antes de pasar el turno de la entrada. Ese pequeño defecto es algo imprescindible en estos tiempos de medidas que varían tanto. No es de agrado para el cliente desplazarse hasta el centro para una vez en la entrada por restricciones de aforo ver que no será capaz de acceder a ninguna de las instalaciones hasta que se vacíe un poco el aforo y desaparezca la cola que hay en la entrada.

Esto podría no ser necesario si hubiese alguna forma de inscribirse en la sala de musculación, la sala principal del gimnasio, el cual no es para una clase dirigida, pero nos da una buena indicación de cuál es el aforo que hay, ya que las clases dirigidas tiene su propio aforo.

También tenemos que señalar, que han elegido una forma de reserva para las clases que, aunque intuitiva ya que se trata de un calendario en el cual se puede visualizar el día de hoy y por horas, se señalan las horas en las que se ofertan todas las clases y simplemente hay que deslizarse hasta encontrar la franja horaria y la clase deseada.

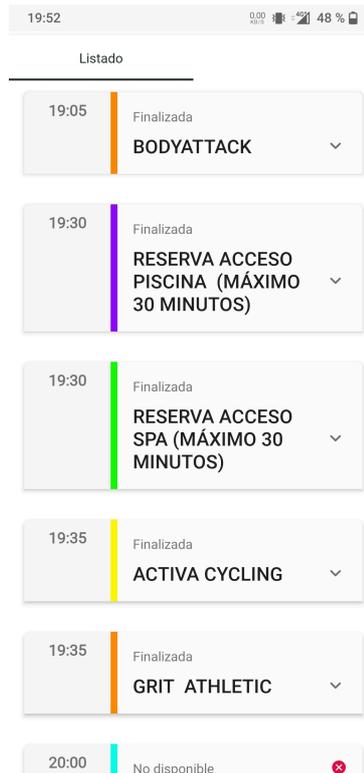


Figura 1: Captura de pantalla app ActivaClub tabla reserva.

El hecho de que la restricción sea de solamente poder realizar la reserva en ese día depende única y exclusivamente del gimnasio, así que eso no es el problema, pero como podemos observar, no se puede filtrar por la clase a la que deseas asistir obligando al usuario que vaya primero a la hora que le gustaría ir y luego teniendo que encontrar si existe en esa franja la clase deseada, así como posteriormente su disponibilidad.

Eso la verdad es que no nos ha convencido mucho, el desarrollo que tenemos en mente es precisamente el contrario, creemos que es mejor que el cliente elija a que clase quiere ir y posteriormente que le aparezca un calendario con todas las horas a las que se puede inscribir.

2.2 Centro deportivo Moncada [8]

Para ello hemos tomado como ejemplo también una aplicación de reserva simple, muy intuitiva y fácil para el cliente, indicando primero qué es a lo que nos queremos inscribir y posteriormente nos aparece todas las ofertas y las plazas que quedan, para que sin ningún tipo de duda podamos inscribirnos a estas.



Figura 2: Captura de la elección de las clases.

Como se puede comprobar en la Figura 2, en la aplicación del centro deportivo de Moncada, es más sencillo realizar la reserva de aquello a lo que queremos asistir.

La aplicación es tan sencilla que está enfocada únicamente a la reserva de actividades, para más información ya cuenta con una página web.

Uno de los problemas que encontramos es que no contempla ningún tipo de registro y para reservar una plaza hay que introducir el DNI, así mismo, no contempla ningún requisito límite de reservas y puedes usar el mismo DNI, que no comprueba que sea correcto, para asistir a todas las clases del mismo día.

Por último, hay que remarcar que han adoptado lo llamado como *web responsive* y no disponen de aplicación como tal, para ello hay que utilizar el navegador de nuestros móviles.

2.3 Propuesta

Después de hacer un recorrido por distintas aplicaciones que ofrecen los propios gimnasios y seleccionar dos de las cuales partiremos para realizar nuestro proyecto tanto por su sencillez, como por las posibles opciones que se pueden realizar en ellas, hemos decidido que nuestra aplicación debe tener los siguientes aspectos:

- **Inicio de sesión.** Como hemos podido comprobar con la aplicación de CDM, el hecho de que cualquiera pueda hacer una reserva sin la necesidad de ser miembro del gimnasio es un impedimento para el resto que, si es miembro, es verdad que luego no podrán acceder al gimnasio, pero puede llegar a colapsar las reservas y dejarlas inutilizadas a pesar de que nadie asista a las clases.
- **Interfaz amistosa.** La aplicación debe ser muy intuitiva en cualquiera de sus apartados. La sensación del cliente manejando cualquier tipo de aplicación es importante para mantenerlo interesado, además, al gimnasio van todo tipo de personas y de todas las edades.
- **Filtro de reserva por clases.** Aunque implementemos la reserva de clases con y sin filtro, vemos que tiene mucha utilidad y hace la aplicación más amigable si previamente se indica a qué clase se desea asistir y posteriormente en el horario solo aparecen las opciones de reserva que estamos buscando sin que se repitan las horas del mismo día en caso de que el gimnasio disponga de clases simultáneas o incluso de que se solapen en la pantalla.
- **Control de acceso.** Ciertamente el control de acceso lo pueden implementar de muchas formas distintas, una pulsera, la huella dactilar o pasar por recepción son las técnicas más extendidas. Nosotros planteamos la posibilidad de que cada cliente disponga de un código QR y que sirva del mismo modo que una pulsera. Lo más importante de este punto no es el método que llegue a ser utilizado, sino el de poder visualizar a tiempo real cuántos miembros se encuentran en el gimnasio y saber si voy a poder entrar cuando llegue o tendré que darme media vuelta y volver en otro momento.



Aplicación para reserva de actividades y horarios de un gimnasio mediante una app móvil

Para tener una visión más sencilla de lo que proponemos, se va a exponer en la tabla 1 la comparativa que hemos encontrado en estas dos aplicaciones y la que proponemos.

Tabla 1: Comparativa de aplicaciones

	ActivaClub	CDM	FitStyleGym
Inicio de sesión	✓		✓
Interfaz amistosa		✓	✓
Filtro por clases		✓	✓
Control de acceso	✓		✓
Aforo a tiempo real			✓
Dietas	✓		
Ejercicios personalizados	✓		
Restricción de reservas por cliente	✓		✓
Reservar con más antelación que un día		✓	✓
Aplicación específica para móvil	✓		✓

Como podemos observar, en la tabla mostramos ciertos aspectos que nos han parecido interesantes destacar de cada una de las aplicaciones, siendo nuestra aplicación **FitStyleGym** la más completa.

3. Especificación

En este apartado debemos indicar que vamos a realizar el proyecto mediante una arquitectura de tres capas [9].

3.1 Arquitectura

Es un modelo de programación que abstrae en capas las partes de desarrollo del proyecto lo que nos ofrece ventajas como una alta escalabilidad en caso de que posteriormente las necesidades de los gimnasios aumenten se podrá ampliar de una forma sencilla y es una forma fácil de mantener el proyecto dado que el código cuenta con las siguientes partes:

- La capa de **presentación** es la parte de la vista que el cliente podrá visualizar en su terminal y ejecutará acciones que enviará a la siguiente capa.
- La capa de **negocio** es la que gestiona toda la lógica de la aplicación y funciona como un controlador situado entre la capa de presentación y la de datos.
- La capa de **datos** es la encargada de gestionar la persistencia, acceder y manejar los datos de nuestra aplicación.

Dado que las tres capas están diferenciadas y son totalmente independientes resulta muy sencillo reutilizar código para otras aplicaciones. Este proyecto está orientado a realizar varias aplicaciones con distintas funcionalidades según las exigencias de cada uno de los gimnasios y realizar el proceso mediante la arquitectura de tres capas hace que sea un trabajo ideal para conseguirlo. Esto hace también un fácil manejo de la incorporación de miembros a este proyecto, dado su alta abstracción de cada una de las partes, es posible realizar un desarrollo de forma simultánea.

3.2 Casos de uso

Para ampliar un poco más las distintas funcionalidades que tendrá la aplicación vamos a definir un diagrama de casos que nos ayudará a comprobar los pasos que podrá realizar el usuario de una forma más sencilla y previa al diseño final.

Como hemos mencionado anteriormente, el proyecto contempla que el gimnasio ya dispone tanto de una base de datos como de un sistema para manejar la información de los clientes y el alta de los mismos y no se necesitará un registro por parte del cliente para poder acceder a la aplicación ya que será el propio gimnasio el que les permita el acceso a esta.

El gimnasio, si no dispone de un sistema para listar las instalaciones de las que dispone, clases, horarios y aforo nos facilitará toda la información para el correcto funcionamiento de la aplicación.

Por todo ello, no va a existir un rol de administrador y solo contemplamos los casos de uso donde el cliente.



Figura 3: Caso de uso de usuario no registrado.

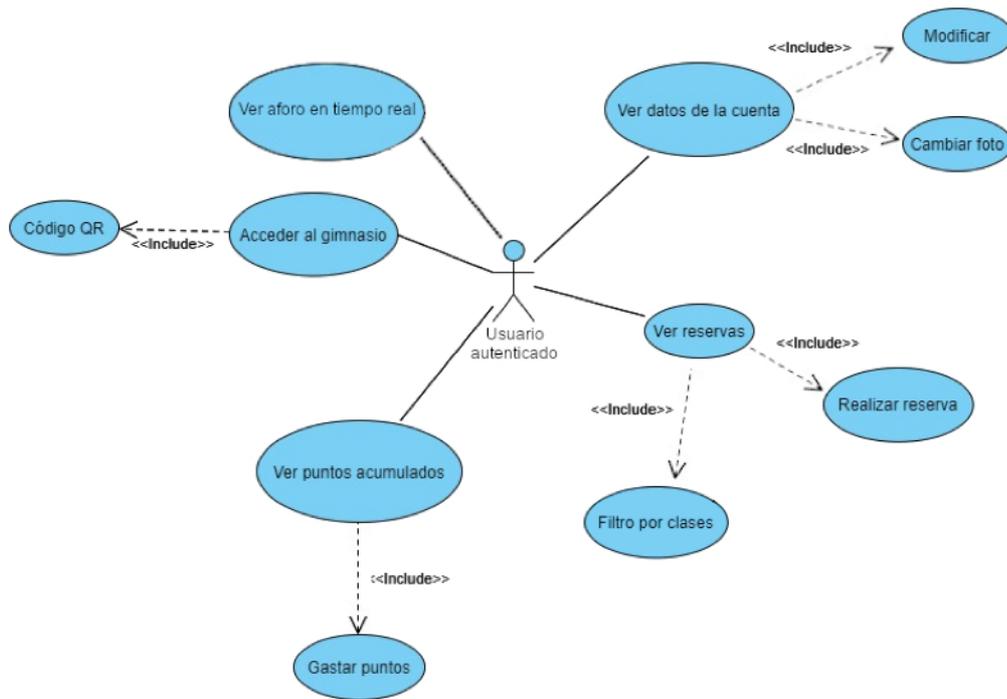


Figura 4: Diagrama de casos de uso.

3.3 Funcionalidades de los casos de uso

Primero encontramos a un actor que llamaremos usuario no autenticado que solo podrá iniciar sesión.

Tabla 2: Usuario no autenticado

Nombre	Iniciar sesión.
Actor	Usuario no autenticado.
Descripción	Utilizar los datos facilitados por el gimnasio para poder utilizar la aplicación introduciendo el usuario y la contraseña.
Precondición	Disponer del usuario y la contraseña y tener acceso a Internet.
Postcondición	El usuario puede utilizar la aplicación.
Excepciones	Introducir incorrectamente algún campo para el inicio de sesión.

Ahora que ya tenemos una cuenta y podemos utilizar la aplicación, tenemos acceso a los distintos casos de uso.

Tabla 3: Ver datos de la cuenta

Nombre	Ver datos de la cuenta.
Actor	Usuario autenticado.
Descripción	Visualizar los datos del usuario que corresponden a la cuenta <i>logueada</i> y con la posibilidad de cambiarlos, así como la foto del perfil.
Precondición	Tener acceso a Internet y haber iniciado sesión
Postcondición	Se actualizan los cambios
Excepciones	No puede haber campos vacíos.

Tabla 4: Ver reservas del usuario

Nombre	Ver reservas.
Actor	Usuario autenticado.
Descripción	Visualizar las reservas del usuario pudiendo realizar nuevas reservas utilizando un filtro por clases.
Precondición	Tener acceso a Internet y haber iniciado sesión
Postcondición	Se añade la nueva reserva a las reservas del cliente.
Excepciones	No se puede reservar si al momento del registro de esta no hay plazas libres.

Tabla 5: Ver puntos acumulados

Nombre	Ver puntos acumulados.
Actor	Usuario autenticado.
Descripción	Visualizar los puntos hasta la fecha actual que el usuario tiene acumulado con la posibilidad de utilizarlos.
Precondición	Tener acceso a Internet y haber iniciado sesión
Postcondición	Descuenta los puntos gastados.
Excepciones	No se pueden usar los puntos en los premios que sean superiores a los puntos acumulados.

Tabla 6: Ver aforo en tiempo real

Nombre	Ver aforo en tiempo real.
Actor	Usuario autenticado.
Descripción	Visualizar la cantidad de personas que hay dentro del gimnasio y el aforo máximo permitido.
Precondición	Tener acceso a Internet y haber iniciado sesión
Postcondición	
Excepciones	

Tabla 7: Acceder al gimnasio

Nombre	Acceder al gimnasio.
Actor	Usuario autenticado.
Descripción	Utilizar el código QR para poder entrar en el gimnasio.
Precondición	Tener acceso a Internet, haber iniciado sesión y que aún haya plazas para el acceso al gimnasio.
Postcondición	El cliente puede acceder al gimnasio.
Excepciones	Si no hay plazas no se permite el acceso.

4. Diseño de la solución

En este apartado vamos a explicar los conceptos de como se ha diseñado la aplicación para el posterior desarrollo de esta. Podemos decir que esta solución será la base de cualquiera de nuestras futuras aplicaciones para el resto de los gimnasios, independientemente de los datos que ya tenga el gimnasio o sus propias necesidades.

4.1 Arquitectura del sistema

Para empezar a explicar el proceso de cómo se ha diseñado la arquitectura de esta aplicación, es necesario que comentemos brevemente que se trata de una arquitectura cliente – servidor.

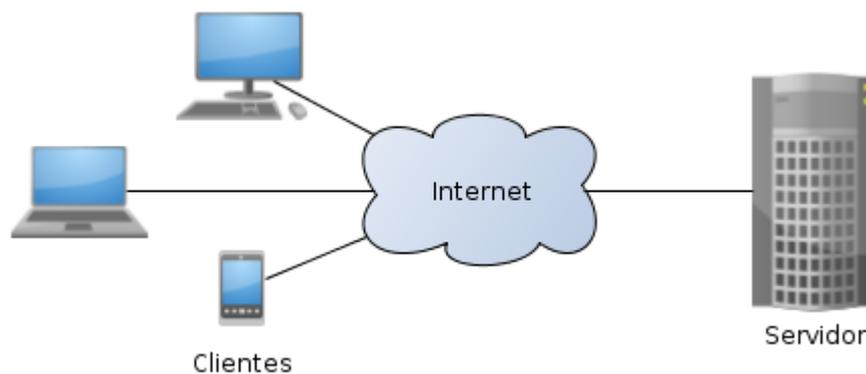


Figura 5: Cliente - servidor.

Todos aquellos a los que nos referiremos clientes que vayan a utilizar nuestra aplicación deberán tener un dispositivo móvil, Android para nuestro ejemplo, en el que mediante una aplicación y con conexión a Internet tendrá un contacto con nuestra API que estará en el lado del servidor.

En el lado del servidor se ejecutarán los servicios necesarios para el manejo de la información y la comunicación con la base de datos.

4.2 Diseño de la capa de presentación

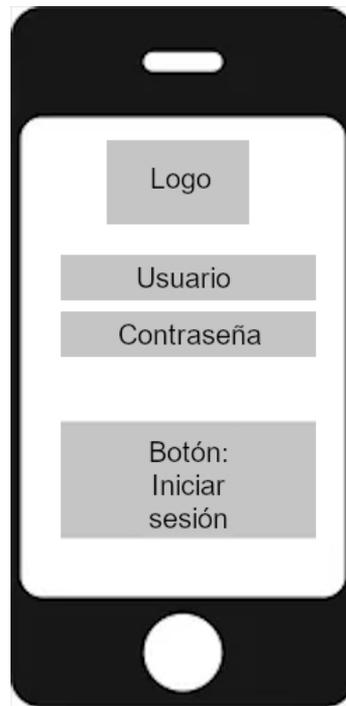


Figura 6: Boceto inicio de sesión

Primero de todo el actor que hemos denominado con el nombre de Usuario autenticado deberá autenticarse como muestra la figura 6. Este es el primer paso que se necesita para utilizar la aplicación. Sí que es cierto que una vez haya iniciado sesión no necesitará hacerlo mientras no esté inactivo más de 15 minutos.

Posteriormente entrará a una ventana a modo de menú de la aplicación el cual podrá encontrar las siguientes opciones que se pueden ver en la figura 7:



Figura 7: Boceto de la ventana menú

- **Mi cuenta.** En esta opción podrá disponer de toda aquella información que dependa de la cuenta del cliente y que esté disponible para su modificación como se puede ver en la figura 8. También incluye la opción de poder cambiar la foto de perfil que accionará la cámara del teléfono.



Figura 8: Boceto de la ventana Mi cuenta

- **Mis reservas.** La principal ventana de la aplicación, que muestra la figura 9, ya que es donde se podrán efectuar las reservas, así como visualizar las que ya tengamos realizadas y que aún no hayan caducado.
A la hora de buscar las clases que se imparten, tendrá acceso a un filtro para disponer solamente del horario de la clase elegida o sin el filtro donde aparecerá un horario con todas las clases.

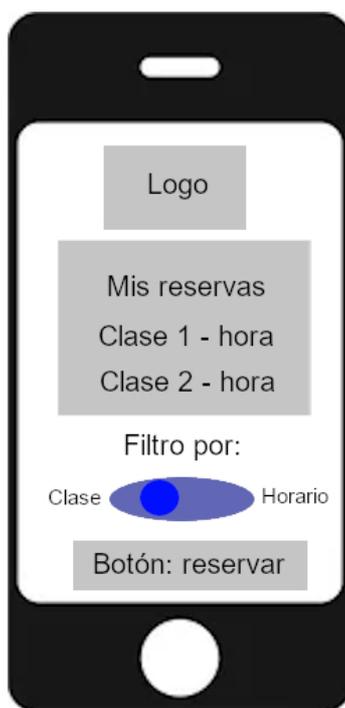


Figura 9: Boceto de la ventana Mis reservas

- **Mis puntos.** En esta última opción que se muestra en la figura 10, podemos ver en la ventana donde nos encontramos aparecerán los puntos acumulados hasta el momento que se podrán conseguir dependiendo de las políticas del gimnasio. También encontraremos un listado de los posibles premios y recompensas que ofrece el gimnasio y los puntos que cuesta cada una de ellas donde se podrá utilizar los puntos almacenados en ellas para canjear los premios.

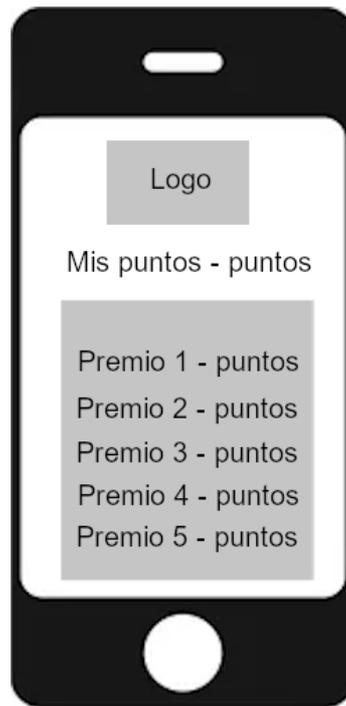


Figura 10: Boceto de la ventana Mis puntos

- Se muestra también un código QR, en la figura 7, que servirá como acceso al gimnasio además del número de personas que se encuentran dentro del mismo y el máximo de aforo del que dispone el gimnasio.

4.3 Diseño de la persistencia de datos

En este apartado mostramos el diseño de la UML que necesitamos para guardar nuestra información y que será necesario para el manejo de la aplicación.

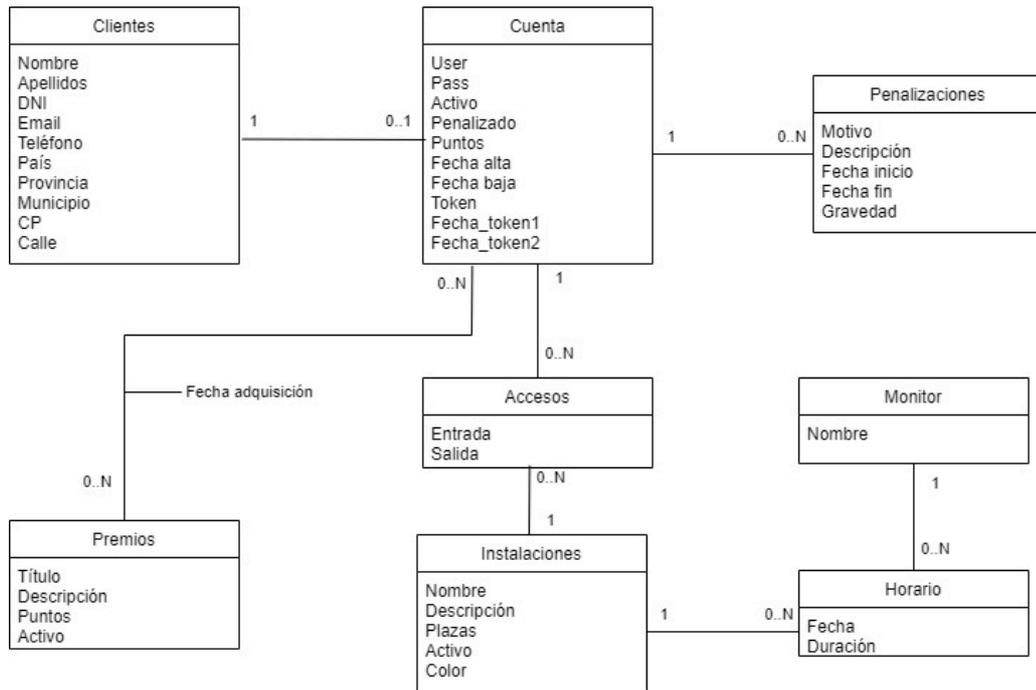


Figura 11: UML [10]

La estructura de la base de datos que se puede observar en la figura 11, es la general para todas las aplicaciones de este proyecto.

Por un lado, tenemos los datos del cliente, esta tabla es posible que sea sustituida por los datos que ya tenga el gimnasio si dispone de ello. Podemos encontrar esta información en la ventana **Mi cuenta** y en la relación con el caso de uso **Ver datos de la cuenta**.

Hemos incluido la penalización que puede traer las consecuencias oportunas que elija el gimnasio, como no poder asistir a clases si reserva y sin previo aviso no asiste.

Así mismo se ha añadido un posible campo de puntos por si el gimnasio quiere premiar al cliente por los accesos o los posibles logros que consiga en sus clases.

A forma de motivación para los clientes, hemos añadido una tabla de premios. Hemos podido observar que muchos de los gimnasios dan algún tipo de recompensas para que el cliente se sienta cómodo y motivado y renueve su matrícula, participe en las clases y acuda al gimnasio a menudo los cuales se encuentran en la ventana **Mis puntos** relacionados con el caso de uso **Ver puntos acumulados**.

El resto de las tablas son necesarias para el correcto funcionamiento en la parte cliente para que se pueda elegir donde y cuando se quiere realizar la reserva de las clases que encontrara en la ventana **Mis reservas** con su relación en el caso de uso **Ver reservas**, así como el posible control de acceso por parte del gimnasio y que dispone de un código QR para ello en la ventana principal de la aplicación relacionado con el caso de uso **Acceder al gimnasio**.

4.4 Tecnología utilizada

En este apartado vamos a separarlo en el apartado del cliente, el apartado del servidor y lo utilizado para realizar este proyecto.

- En la parte del cliente se utiliza como se ha mencionado anteriormente la tecnología web que con la ayuda de Apache Cordova hace de forma sencilla exportar la misma aplicación y compilarla para varios dispositivos móviles.
 - **HTML 5.** Aquí se realiza toda la estructura de la parte visual y donde se van a mostrar los datos.
 - **CSS 3.** Con la tecnología de las hojas de estilo en cascada podemos adaptar todo el contenido que previamente hemos incluido en html para una correcta visualización para el cliente.
 - **JavaScript.** Para incluir la funcionalidad es necesario hacer unos pequeños scripts, estos también son los que nos permiten comunicarnos con el servidor.
 - **Material Icons.** Para los iconos de la aplicación se ha utilizado la librería de Google que de forma gratuita con la utilización del servicio de Google Fonts, se puede incluir iconos a forma de texto el cual es mucho mejor en rendimiento que una imagen, además de poder aumentar y disminuir su tamaño y cambiar de color sin la necesidad de incluir varias versiones ni perder calidad de renderizado.
 - **Apache Cordova.** Por último, tenemos el conjunto de herramientas que nos proporciona compilar con muy pocos cambios para cualquier sistema operativo de los distintos dispositivos móviles que hay en el mercado.
- Por otro lado, tenemos la tecnología utilizada en la parte del servidor.

- **PHP** [11]. Este es uno de los lenguajes más populares que se desarrollan en la parte del servidor, muchas de las webs que podemos encontrar se han realizado con el CMS Wordpress, probablemente el más usado, desde sus inicios ha utilizado el lenguaje de programación PHP, esto y sus continuos avances hacen que sea uno de los principales junto a JavaScript para realizar páginas web.
- **MySQL** [12]. Para la persistencia de los datos se ha decidido utilizar MySQL, dado que se trata de una base de datos tradicional estructurada con muchos años de estabilidad y que funciona muy bien junto a PHP, se prefirió utilizar frente a la moderna NoSQL dado que esta es preferible cuando la cantidad de datos que se manejan es mucho mayor pero suele ser necesario contar con algún conjunto de herramientas externas dado que su gran potencial a la hora de escalar pero la falta de herramientas propias para su robustez, es de obligado uso utilizar alguna de estas para evitar futuros errores.
- **API REST** [13]. Como hemos mencionado anteriormente, se trata de una arquitectura cliente – servidor, y para la conexión de ambos utilizamos la común API Rest, se trata de una comunicación sin estado que los datos guardados no se enlazan con la solicitud del cliente, aunque esta característica tan común de estas APIs no es lo que nosotros necesitamos y por lo tanto lo hemos implementado mediante la utilización de unos *tokens* únicos y seguros que permitirán realizar las conexiones clientes sin la necesidad de transferir en cada transacción las credenciales del usuario.
- Por último, se han utilizado unas herramientas o servicios para realizar este proyecto que se explicará mejor en el siguiente apartado.
 - **Heroku**. Se ha decidido subir a este servidor este proyecto porque ya lo hemos utilizado anteriormente y tiene una parte gratuita perfecta para a demostración de proyectos o pequeños proyectos.
 - **GitHub**. Dado que hemos decidido confiar nuestra parte servidor con Heroku, este tiene una opción que sustituye el típico FTP que tanto estamos acostumbrados para transferir nuestros archivos al servidor por una conexión que utiliza con GitHub,



además se puede configurar una subida automática cuando detecte la realización de cualquier cambio en nuestro código. Por supuesto su control de versiones es algo espectacular que te evita muchos problemas a la hora de realizar cambios que parcialmente funcionan y en un futuro detectas cualquier vulnerabilidad o un fallo con la conexión a otro componente, asimismo, aunque no lo hemos necesitado para este proyecto, es una herramienta realmente fantástica si en el proyecto participan varios miembros en un mismo archivo.

Como se trata de una aplicación móvil hemos intentado que sea lo más ligera posible y utilizar la base de todas las tecnologías sin *frameworks* o herramientas externas que pudiesen hacer la aplicación más pesada o incluir herramientas que no necesitamos para esta aplicación, así como no consumir datos en la comunicación de forma innecesaria.

Por ello se ha utilizado la tecnología web desde su base que consiste en HTML, CSS y JS. Podríamos haber utilizado el tan extendido y utilizado cada vez más AngularJS o React pero, por los motivos que hemos comentado en el párrafo anterior, hemos decidido que para este proyecto que no es extremadamente grande prima la velocidad y la memoria a la comodidad de programación.

La balanza se ha decantado por Material Icons [14] en vez de su directo competidor [15] ya que pertenece a Google y sabemos que como empresa es imposible que nos deje sin soporte y está unido a Google Fonts, lo cual hace que la carga en la conexión sea inferior.

Para hacer la *build* del código web a la compilación para su exportación en Android se ha utilizado Apache Cordova [16] que nos ofrece todas las funcionalidades que necesitamos. También existe una Ionic [17], pero con la misma filosofía que llevamos hasta ahora hemos pensado que, dado que Ionic utiliza Apache Cordova para realizar la exportación y sus *plugins* hemos pensado que no vale la pena introducirlo en nuestra aplicación ya que tiene muchas herramientas para la visualización que aumentaría el tamaño de nuestra aplicación y su base es exactamente igual a la realizada con Apache Cordova.

En la parte servidor hemos utilizado PHP ya que nos sentimos más cómodos y familiarizados con el lenguaje, ofrece una increíble respuesta para las API Rest y

está muy extendido en las páginas webs y sus herramientas para el manejo de la base de datos MySQL realiza una increíble función dando un soporte extraordinariamente sencillo al programador y son herramientas que nos permite realizar el proyecto para una conexión con el prototipo real en Heroku de forma gratuita.



5. Implementación

5.1 Cliente

Para la realización de la parte cliente se ha utilizado tecnología web y valiéndonos de nuestra experiencia con esta junto a las funcionalidades de Apache Cordova obtenemos la siguiente estructura.

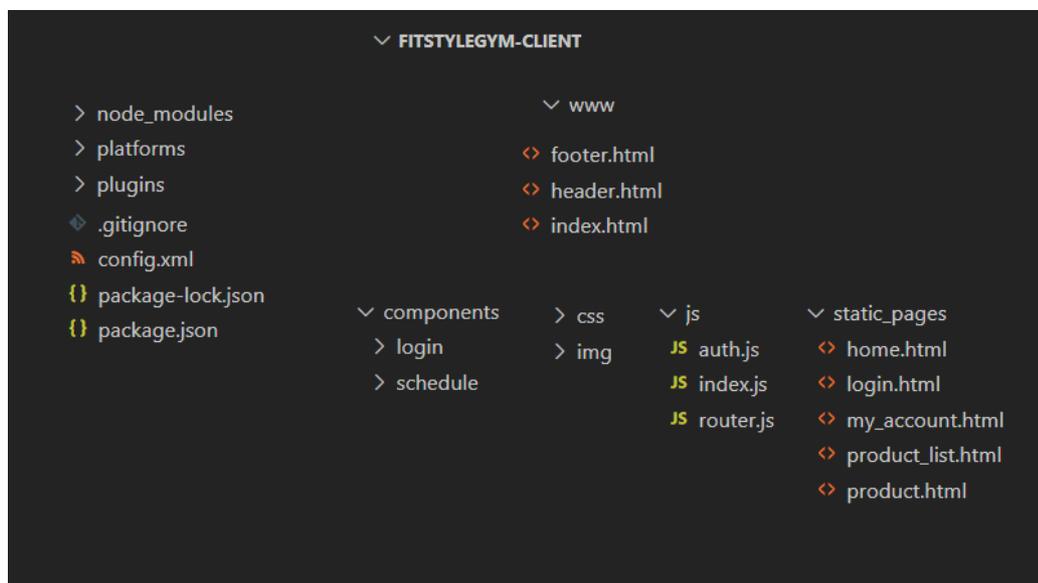


Figura 12: Estructura de la parte cliente.

En la figura 12 se muestra la estructura por carpetas y archivos que se ha utilizado en este proyecto para la parte del cliente. Para comenzar, debemos indicar que todo se encuentra contenido en la carpeta principal del proyecto de la parte cliente llamada FITSTYLEGYM-CLIENT el cual dentro podemos encontrar archivos de funcionalidad y creación del propio Apache Cordova que hace posible la compilación y exportación a los dispositivos que necesitemos, en nuestro caso Android, el cual se han situado a la izquierda para poder visualizarlos pero que no se va a explicar ya que son del propio sistema.

Por otro lado, en la parte derecha, encontramos la carpeta www, la cual hace referencia a todo nuestro código y los archivos web que necesitemos crear para la construcción de la aplicación.

Dentro de la carpeta `www` encontramos los tres archivos `html`, los cuales son las vistas y llamadas que se van a ejecutar en toda la aplicación. El `header` y el `footer` hacen referencia a enlaces y llamadas que todas las ventanas y procesos tienen, así que se ha decidido abstraer el código para que no sea repetitivo.

Seguidamente encontramos la carpeta `components`. Esta alberga el código tanto de la vista como unos `scripts` de unos componentes o módulos que posteriormente se pueden reutilizar y que son integrados en las vistas estáticas que se mencionarán posteriormente.

A continuación de `components`, podemos encontrar las carpetas `css` e `img` que contienen el código de las hojas de estilo para la correcta visualización del código `html` y las imágenes que aparecen en nuestra aplicación.

En la carpeta `js` tenemos lo que podríamos definir como el *core* o el núcleo de nuestra aplicación cliente.

- `Auth.js` contiene los mecanismos necesarios para la conexión con el servidor y nuestra cuenta, ya que en la aplicación se debe estar identificado para poder hacer las transacciones y operaciones.
- `Index.js` es el archivo que contiene toda la funcionalidad de la aplicación junto a los `scripts` particulares que contiene cada componente.
- `Router.js` es un sencillo planificador que hemos creado a modo de filtro. Este se encarga de planificar y ejecutar cuál de las vistas se debe de mostrar al cliente.

Para finalizar podemos observar una última carpeta llamada `static_pages`. Esta contiene las vistas de la aplicación a modo de ventana estática y que a diferencia de una página web convencional, el cliente ya tiene la vista y el código que se ejecutará en su *smartphone* y no necesita descargárselo del servidor, esto ahorrará mucha información que el cliente no necesita descargarse a través de la red y por lo tanto no consumirá recursos como los datos móviles.

Con todo esto obtenemos la parte de aplicación que va a tener en su *smartphone* el cliente. Todo esto se conectará con las funcionalidades del servidor que se explicarán en el siguiente punto.



5.2 Servidor

Ahora es el turno del servidor, aquí se encuentran las funcionalidades y consultas que todo cliente realizará y se comunicará con los datos que tengan los gimnasios.

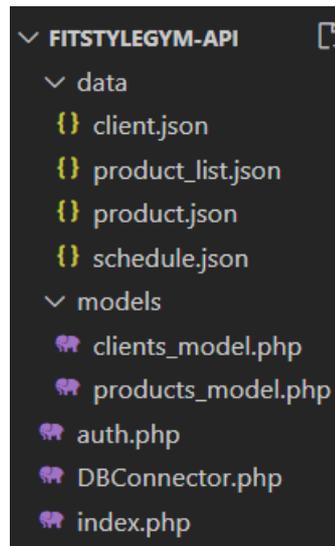


Figura 13: Estructura de la parte servidor.

Como podemos observar en la figura 13, la estructura de archivos y carpetas para la organización y funcionamiento del servidor y los datos es bastante simple.

Por un lado, encontramos la carpeta data, ahí se encuentran los archivos json que contienen la estructura que deberán tener todos los datos como objetos de tipo json. Esta carpeta hace de interfaz o referencia para que la funcionalidad de las demás operaciones no cambie, si los objetos mantienen siempre una misma integridad, hace más sencillo su escalabilidad ya que sabemos en todo momento qué tipo de datos estamos manejando.

A continuación, podemos encontrar la carpeta models, esta contiene los archivos que se comunican directamente con la base de datos. Al tener separada las consultas a la base de datos y la parte del controlador obtenemos una mejor organización e independencia en caso de escalar el proyecto o incluso que varias personas participen en él sin la necesidad de saber qué hace cada una de las

clases. Este tipo de organización está sacada del patrón de arquitectura MVC (modelo, vista, controlador) que se suele utilizar al crear páginas web en php.

Para finalizar, existen los siguientes tres archivos alojados en la raíz:

- Auth.php que se encarga de la autenticación del cliente. Como ya hemos mencionado anteriormente y explicaremos mejor en el siguiente apartado, se trata de una API Rest, esta tecnología se caracteriza entre otras por no tener estado, esto quiere decir, que a diferencia de las páginas webs convencionales, las peticiones son independientes unas de otras, no se puede guardar una sesión para saber qué cliente había iniciado sesión, así como un posible carrito de compra en el típico comercio electrónico. Para solventar este problema y de una forma segura, se ha realizado la creación de un *token* para mantener una especie de sesión. Este lo genera el servidor y lo almacena en la base de datos dejando dos fechas de caducidad y se lo envía al cliente para que pueda continuar con las siguientes transacciones. El cliente guardará en la aplicación ese *token* el cual indicará que ya tiene iniciada la sesión sin la necesidad de volver a autenticarse ni guardar contraseñas que pueden hacer más vulnerable la aplicación ante un ataque.

En cada comunicación con el servidor, el cliente mandará el *token* que tiene guardado para que el servidor compruebe si puede realizar la operación o necesita autenticarse de nuevo. Si el *token* que envía el cliente existe y ninguna de las dos fechas que mencionábamos antes se han alcanzado ya, el cliente podrá operar con normalidad. Cuando este haga una transacción con el servidor y la primera fecha sea anterior a la fecha actual, la cual solo dura cinco minutos, el servidor generará un nuevo *token* que mandará al cliente para renovar e introducir un mecanismo de seguridad que no funcionaría si es la contraseña la que va navegando de un lado a otro. En este proceso la persona que está utilizando la aplicación no continúa utilizándola con normalidad, además se actualiza de nuevo la segunda fecha. Esta segunda fecha que dura quince minutos sirve para una vez sea anterior a la fecha actual el cliente deberá volver a iniciar sesión para que, frente a una



suplantación, robo de datos o inclusión de otras aplicaciones no puedan estar en permanente comunicación.

- DBConnector.php solamente tiene un código genérico de conexión a la base de datos y su conexión, así como la ejecución de las sentencias SQL que vienen de los archivos que contiene la carpeta models, esto hace más sencillo la tarea de exportar el código a otras plataformas o una abstracción de código que si cambiara la versión de la base de datos, los propios datos de conexión o cualquier paradigma de programación que necesite ser reprogramado, se haga únicamente en este archivo y no en todos los que contienen sentencias de acceso a la base de datos.
- Index.php hace de controlador, aquí se llama al resto de archivos y es donde el cliente únicamente se puede conectar. Como cualquier API Rest tiene las siguientes funcionalidades que el cliente puede consultar utilizando una llamada https:
 - GET: <https://ejemplo.com/clientes/1>. Este obtendría los datos del cliente con la id 1.
 - POST: <https://ejemplo.com/clientes>. La llamada con post crearía un nuevo cliente.
 - PUT: <https://ejemplo.com/clientes/1>. Put actualiza los datos del cliente 1.
 - DELETE: <https://ejemplo.com/clientes/1>. Borra el registro cliente al que le corresponda la id 1.

En este apartado se va a especificar un poco más qué se ha realizado para el prototipo de este proyecto, ya que hasta ahora se ha realizado una explicación de la aplicación de modo general que tendrían todos los gimnasios. Ahora se pondrá un ejemplo como si un gimnasio específico nos hubiese pedido la aplicación y se le haya adaptado para este.

Lo primero que hemos realizado es crear dos repositorios en GitHub.

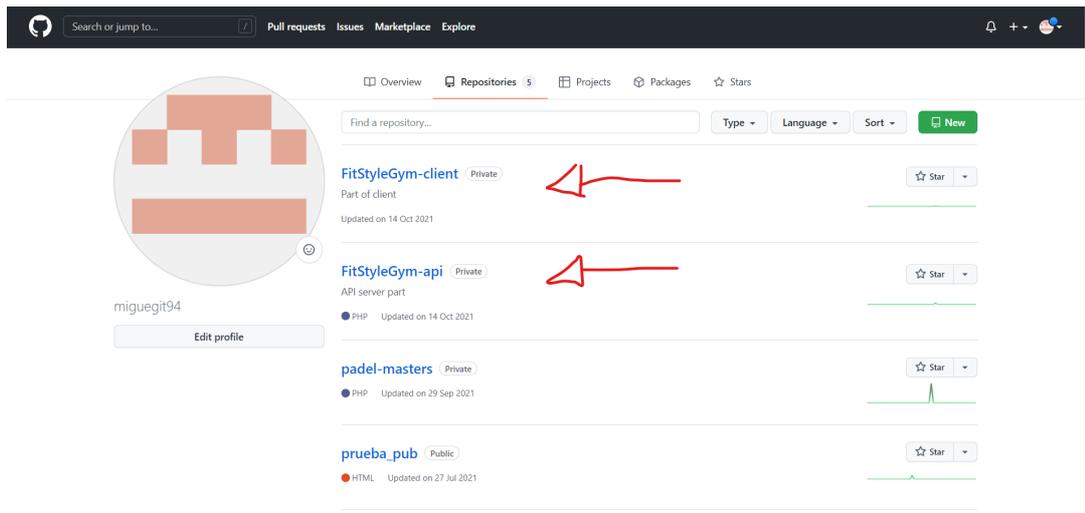


Figura 14: Proyectos en GitHub [18].

Los que podemos observar en la figura 14 remarcados con una flecha, un repositorio para todo el código relacionado con el cliente y otro distinto para el servidor que posteriormente conectaremos con Heroku.

En Heroku tenemos varios precios y planes, pero los conceptos básicos los podemos ver en la figura 15 donde se comentan los planes que hay generales, aunque hay más configuraciones que se pueden consultar en su propia página.

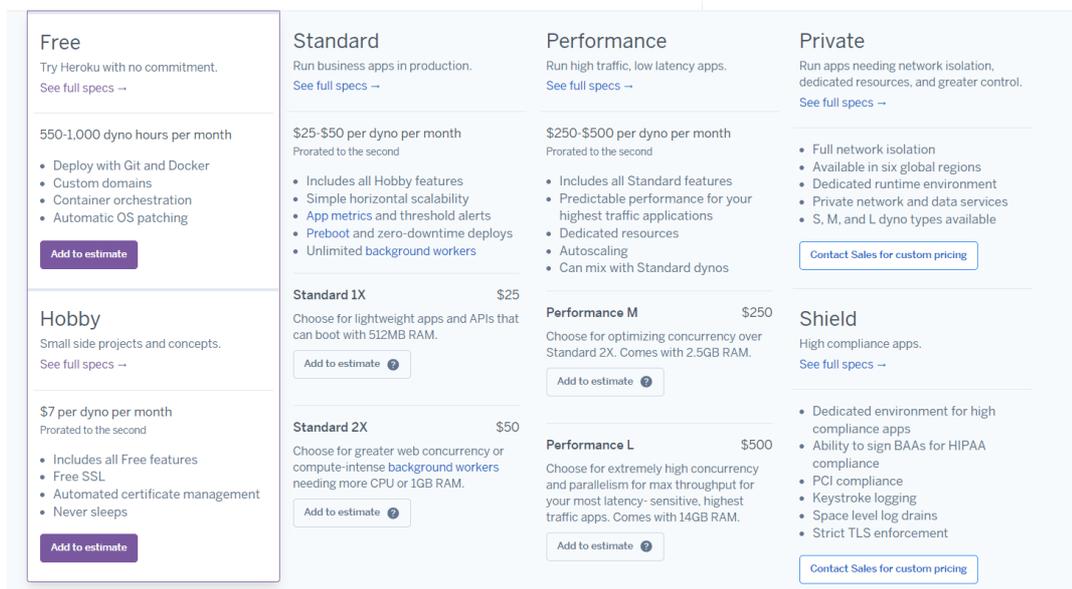


Figura 15: Heroku [19].

Heroku es una plataforma de servicio basada en un sistema de contenedores administrados, con servicios de datos integrados y un ecosistema poderoso, para implementar y ejecutar aplicaciones modernas. La experiencia del desarrollador de Heroku es un enfoque centrado en la aplicación para la entrega de software,



Aplicación para reserva de actividades y horarios de un gimnasio mediante una app móvil

integrado con las herramientas y los flujos de trabajo para desarrolladores más populares de la actualidad.

Para un proyecto prototipo que no va a tener más funcionalidad que la propia demostración de este, se ha escogido la opción *FREE*. Nos da la opción de conectar con un repositorio de GitHub que previamente hemos creado, también podremos obtener de forma gratuito una base de datos con limitaciones de conexiones y cantidad de información que se puede transmitir, aun con todo esto, es suficiente para la realización de este proyecto, y si la necesidad se vuelve mayor o es necesario pasarlo a producción, tiene unos planes asequibles y estables que por proyectos anteriores hemos podido comprobar que es una buena opción.

Una de las restricciones y que hacen posible que sea gratuito, es que, cuando lleva treinta minutos sin ser utilizado entra en una fase de *sleep* que mantiene toda la configuración y datos, pero deja las conexiones inactivas, las cuales son levantadas de nuevo con tan solo iniciar una comunicación con el servidor. Es algo perfecto para prototipos que permite ahorrar energía y cuando es necesario reactivar el servicio, solo es necesario hacer una petición y en unos segundos ya se puede utilizar, no es necesario entrar a la configuración y levantar el servicio manualmente.

Para poder escribir código de una forma sencilla, con un buen corrector de sintaxis y semántica de programación para todos los lenguajes que vamos a utilizar y ya que se pueden agregar módulos y extensiones de casi cualquier tipo ya que es un programa que ha obtenido mucha popularidad en estos tiempos, además de ser una herramienta gratuita, se ha utilizado Visual Studio Code.

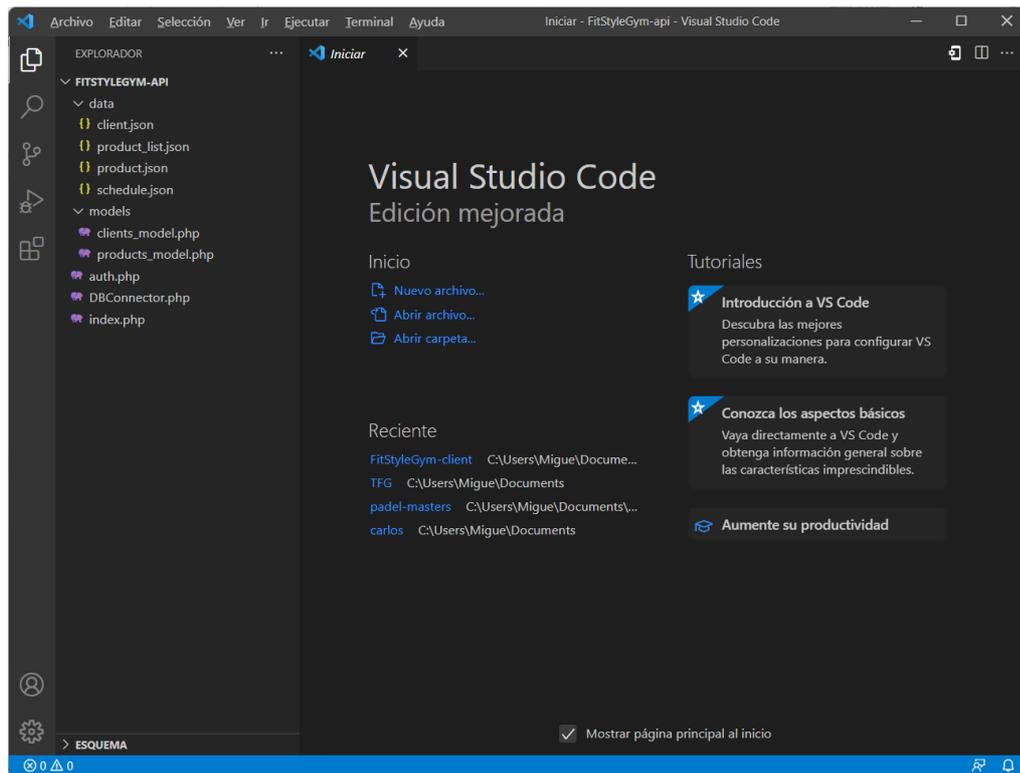


Figura 16: Visual Studio Code [20].

Para su mejor manejo de php, se ha utilizado la extensión PHP Intelephense, el cual ayuda con su manejo de errores sintácticos y semánticos, añade una paleta de colores para cada tipo de estructura del lenguaje, así como una configuración personalizada a gustos del programador.

Como este prototipo se ha realizado para Android, se ha instalado otra extensión llamada Android iOS Emulator, el cual vincula nuestros emuladores de Android al propio Visual Studio Code.

Además, Visual Studio Code, tiene implementada su propia consola, el cual se pueden ejecutar comandos e instalaciones desde ahí mismo. Una gran ventaja que permite que sin salir de la aplicación realizar distintas conexiones con otras herramientas o utilizar los propios comandos de Apache Cordova.

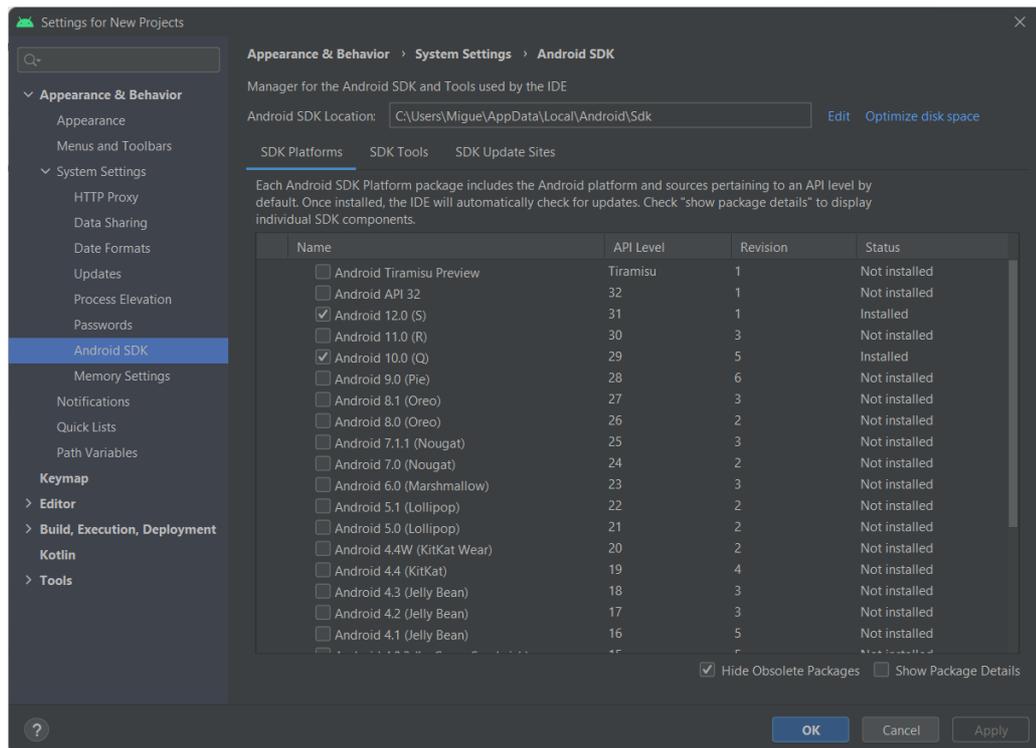


Figura 17: SDK Android.

Obviamente se necesitaba instalar Android Studio para poder descargar el SDK de Android e instalar un dispositivo que posteriormente podríamos emular junto a la extensión del mencionado anteriormente Android iOS Emulator, el cual nos permite realizar todas las pruebas necesarias de nuestro proyecto para su futura compilación en el estado final.

Se ha decidido instalar el SDK 12 para realizar la aplicación más actualizada posible y además hemos procedido a la instalación de unas versiones anteriores, nos ha parecido que la 10 era suficiente por si algún dispositivo no actualizado aún mantiene una versión antigua que tenga soporte por Google, para poder comprobar el buen funcionamiento de nuestra aplicación ya que Apache Cordova, nos indica en su documentación, que la versión que estamos utilizando es la 10.x.

También hemos decidido utilizar para realizar nuestras pruebas y poder emular la aplicación, la virtualización del Pixel 4.

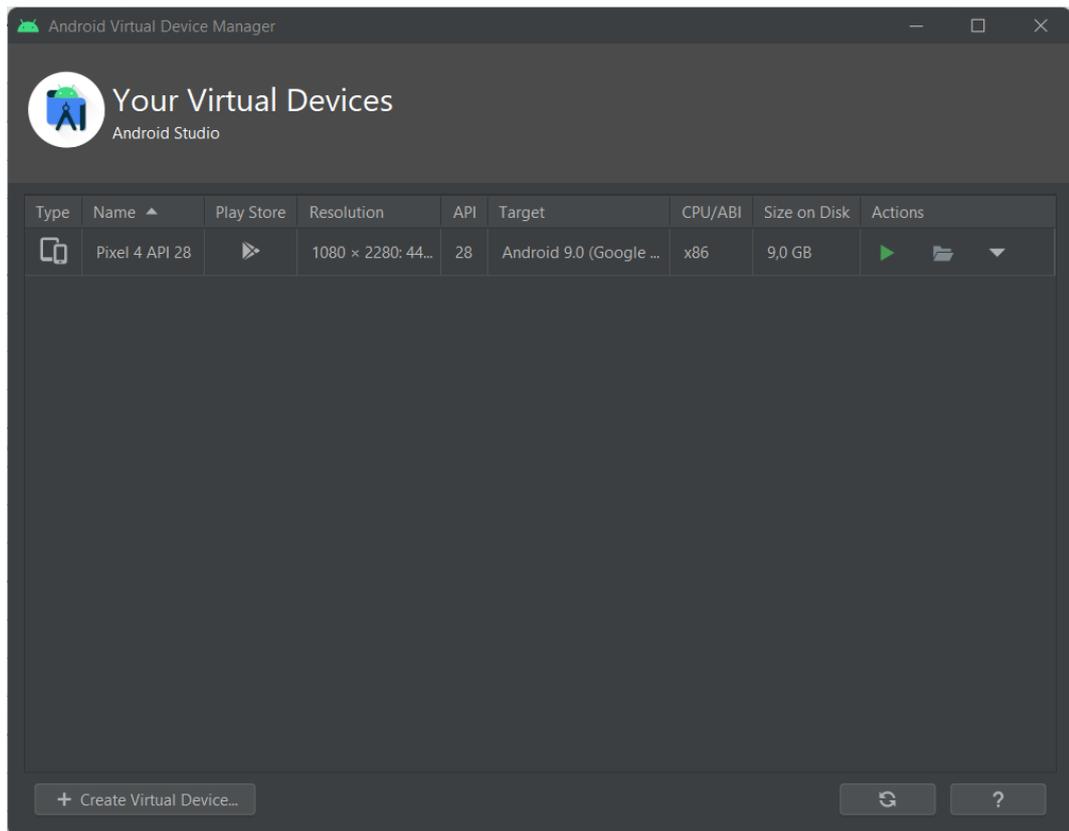


Figura 18: AVD Manager.

Por último, se ha descargado y configurado Apache Cordova.

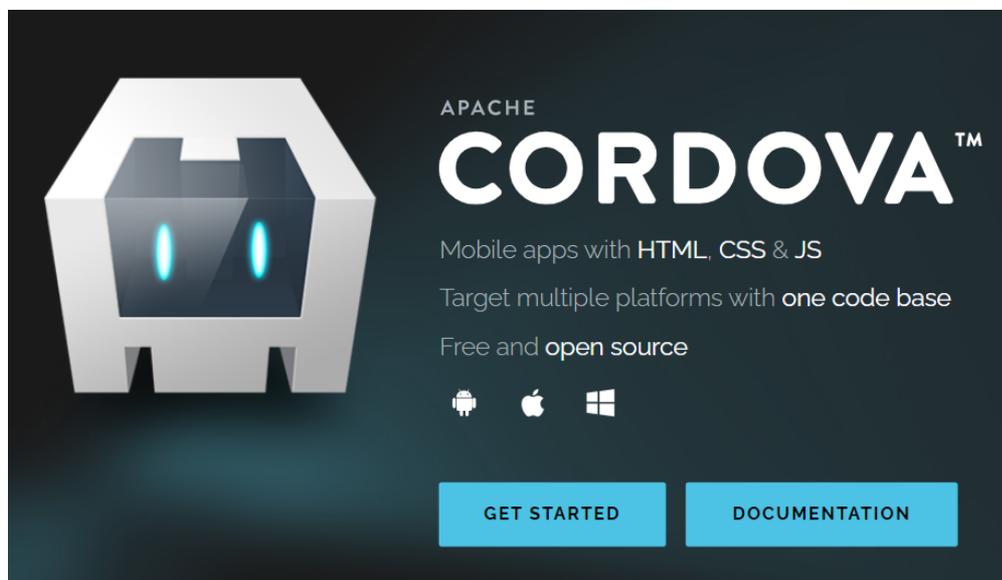


Figura 19: Apache Cordova.

Recordemos que era necesario para poder compilar nuestro código con tecnología web a Android y poder ejecutarlo en nuestro propio *smartphone*.

Una vez todo listo ya empezamos a escribir código para conseguir nuestro propósito, aunque previamente se realizó un esquema de qué necesitábamos y qué queríamos realizar y lo teníamos todo bastante claro antes de empezar a escribir, decidimos empezar por la parte cliente por si posteriormente teníamos que ampliar alguna funcionalidad y realizar la parte del servidor con el mínimo de alteraciones.

Para conectar el cliente con el servidor y sus futuras transacciones de forma segura sin utilizar sus credenciales, salvo cuando inicia sesión, se utiliza el estándar OAuth 2.0, utilizado en prácticamente todos los sistemas de API Rest que existen en la actualidad, que junto al protocolo HTTPS obtenemos una gran seguridad para la comunicación con nuestras aplicaciones.

Authorization Code Flow

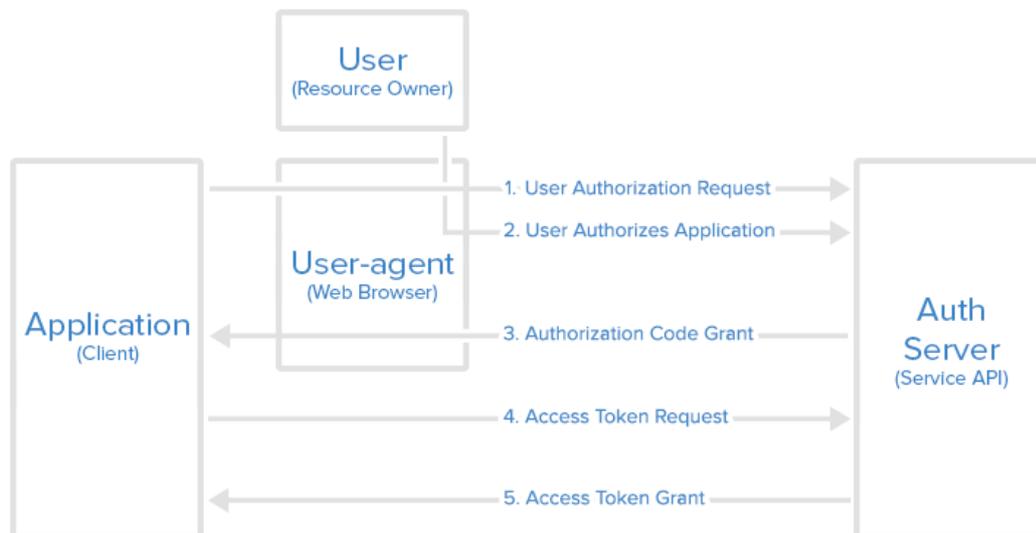


Figura 20: Flujo de autenticación con OAuth 2 [21].

Una vez realizamos el proceso de inicio de sesión y terminado las operaciones de la figura 20, si todo es correcto, se debería obtener una respuesta válida parecida a la que aparece en la figura 21.

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "2YotnFZFEjr1zCsicMwpAA",
  "token_type": "example",
  "expires_in": 3600,
  "refresh_token": "tGzv3JOkF0XG5Qx2TIKWIA",
  "example_parameter": "example_value"
}
```

Figura 21: Token obtenido.

Con estos cinco pasos que se muestran en la figura 20, obtenemos una gran seguridad en nuestro proyecto, que como hemos mencionado anteriormente servirá para crear una especie de sesión que este tipo de APIs no se puede obtener.

Además, le añadimos un mecanismo extra de seguridad con las dos fechas que hemos visto en la figura 11 y mencionado en el capítulo 5.2, la Fecha_token1 nos ayudará a realizar operaciones que mientras se ejecuten en una franja de cero a cinco minutos.

Utilizaremos el mismo *token* para las siguientes transacciones y si la Fecha_token1 ya ha caducado pero la fecha actual es inferior a Fecha_token2 realizará una renovación del *token* que mandará al cliente para las futuras transacciones. En caso de que esta fecha también caduque, borrará de la base de datos la persistencia de cualquier *token* relacionado con esa cuenta para que obligue desde la parte cliente la inserción nuevamente de las credenciales para su nuevo inicio de sesión y repetir el proceso.

Después de las pruebas y que este flujo de trabajo nos haya permitido realizar la aplicación y obtener el resultado que esperábamos, indicamos a Apache Cordova que ya queremos exportar la aplicación a un entorno real con el siguiente comando que se muestra en la figura 22.

Aplicación para reserva de actividades y horarios de un gimnasio mediante una app móvil

```
cordova run android --release -- --keystore=../my-release-key.keystore --storePassword=password  
--alias=alias_name --password=password --packageType=bundle.
```

Figura 22: Comando para exportar la aplicación y firmar.

Esto genera un archivo llamado build.json que tiene una estructura como la de la figura 23.

```
{  
  "android": {  
    "debug": {  
      "keystore": "../android.keystore",  
      "storePassword": "android",  
      "alias": "mykey1",  
      "password": "password",  
      "keystoreType": "",  
      "packageType": "apk"  
    },  
    "release": {  
      "keystore": "../android.keystore",  
      "storePassword": "",  
      "alias": "mykey2",  
      "password": "password",  
      "keystoreType": "",  
      "packageType": "bundle"  
    }  
  }  
}
```

Figura 23: Build.json.

Ahora ya tenemos nuestra aplicación exportada en su estado *release* lista para ser usada en un *smartphone* real con Android.

Algo que tenemos los usuarios de Android a diferencia de los de iOS, es que podemos instalar aplicaciones sin la necesidad de encontrarlas en nuestra tienda del teléfono. Por supuesto, este proceso se realizará para la prueba de funcionamiento de la aplicación, puesto que los futuros clientes estarán más cómodos y familiarizados si la aplicación se encuentra en la Play Store de Android o en una posible exportación a iOS en App Store.

Para las versiones anteriores a Android 10, para poder instalar aplicaciones no provenientes de la Play Store, era necesario realizar un paso previo como se muestra en la figura 24.

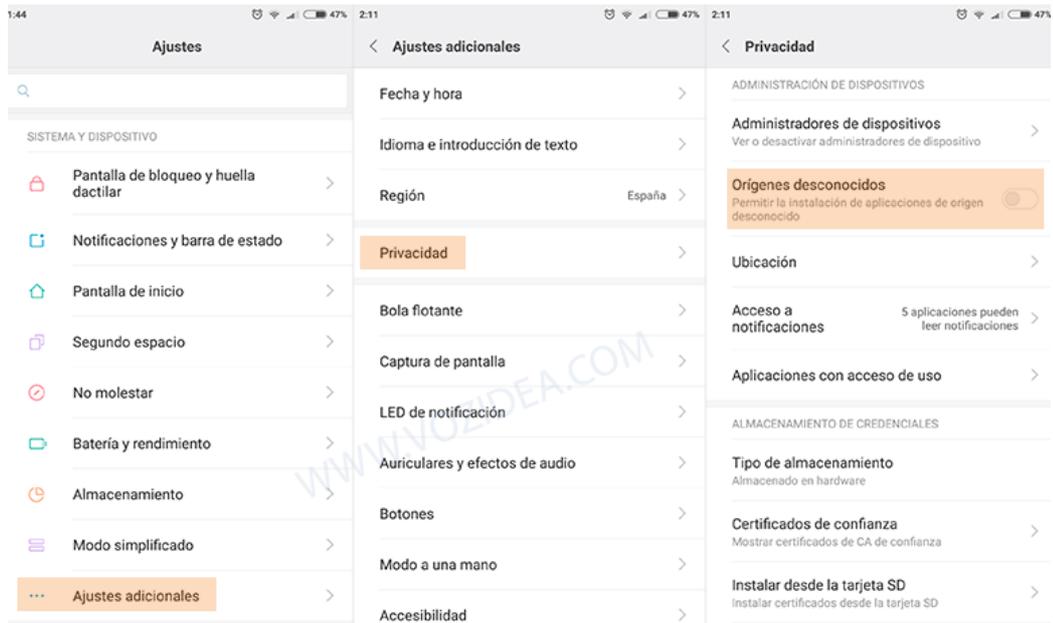


Figura 24: Instalación de aplicaciones desconocidas versión inferior a la 10.

Actualmente, se ha vuelto un poco más complicado, pero nada que de momento no se pueda hacer, simplemente hay que realizar los pasos de la figura 25.

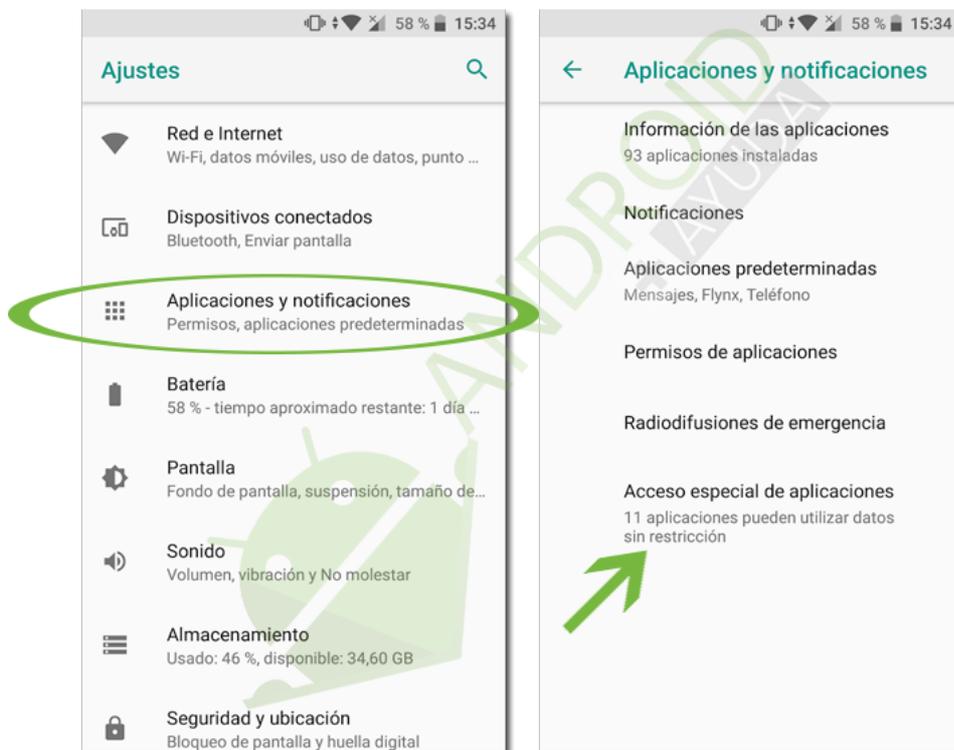


Figura 25: Instalación de aplicaciones desconocidas versión superior a la 10.

Aplicación para reserva de actividades y horarios de un gimnasio mediante una app móvil

Y posteriormente elegir cual es el origen de dicha instalación.

Después de cualquiera de ambos procesos ya estamos listos para probar de forma real nuestra aplicación. Este prototipo se explicará de forma más detallada en el siguiente apartado.

6. Resultados y presentación

En este apartado vamos a mostrar y explicar paso a paso como nos ha quedado nuestro prototipo, simulando como hemos mencionado anteriormente, una posible petición de un gimnasio como aplicación real.

La parte del servidor ya explicada, no se mencionará para que no quede redundante. Solo hay que aclarar que la finalidad de este proyecto estaba orientada a la parte cliente y a una API que realizaría las conexiones entre una posible base de datos ya existente por parte del gimnasio y nuestra personalización. Por supuesto, en caso de ser necesario, se podría montar un panel de gestión en caso de que el gimnasio no dispusiera de ningún tipo de sistema previo.

Para este prototipo hemos supuesto que los datos ya los tenían y que el gimnasio nos ha facilitado la información necesaria para completarla, la cual, vamos a introducirla directamente en la base de datos.

Ahora sí, podemos pasar a la parte cliente.

Lo primero que se va a encontrar el cliente al abrir la aplicación es la ventana de inicio de sesión como muestra la figura 26.



Figura 26: Inicio de sesión del cliente.

Después de proporcionarle al cliente de forma previa un usuario y una contraseña, este podrá iniciar sesión con ella, asimismo, dado que prácticamente todos los *smartphones* tienen un buen sistema de huella dactilar como principal medida de seguridad, se ha añadido también la posibilidad de que el propio terminal guarde las credenciales para con el mecanismo de autenticación biométrica pueda realizar el inicio de sesión.

Una vez dentro del sistema de la aplicación podemos encontrar lo que muestra la figura 27.

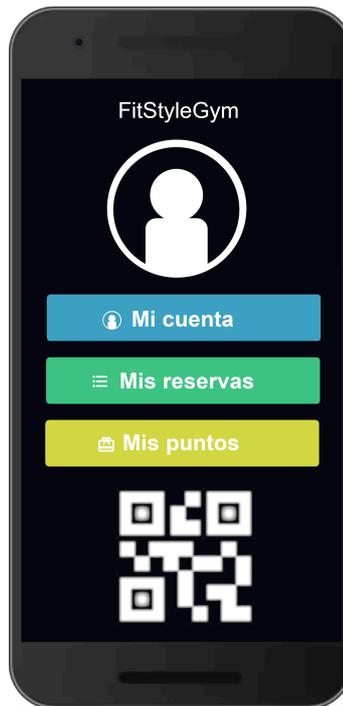


Figura 27: Ventana principal de la aplicación.

El nombre del gimnasio o el título que este le haya querido dar a la aplicación, se verá siempre en la parte de arriba para poder identificarlo de forma clara rápida.

Lo siguiente que encontramos es la foto del cliente asociado a la cuenta que previamente el gimnasio ha podido utilizar para el típico carné que antiguamente teníamos todos los socios del gimnasio para poder identificarnos al estilo del DNI.

A continuación, podemos encontrar tres botones que explicaremos a continuación, pero antes de cambiar de ventana podemos ver que también encontramos el código QR que asocia nuestra aplicación a la entrada del gimnasio.

Algunos gimnasios utilizan una pulsera para su identificación en un torno que solo se abre si la lee de forma correcta, otros utilizan un registro de la propia huella dactilar del cliente, cada gimnasio puede adoptar las pertinentes medidas de seguridad y de control. En esta ocasión se ha añadido un código QR autogenerado al dar de alta al cliente en la aplicación y que este puede mostrar en un torno o en la recepción si dispone de cualquier lector para este tipo de códigos.

Ahora podemos hablar del primer botón que hemos encontrado en la figura 27, la cual al presionarlo se abrirá la ventana que se muestra en la figura 28.



Figura 28: Mi cuenta y mis datos.

Al presionar en el botón de Mi cuenta, podemos encontrar los datos que previamente le hemos facilitado al gimnasio para realizar la matricula.

Podemos cambiar nuestra foto de perfil presionando Cambiar foto y hacernos un buen selfi que aparecerá en nuestro perfil.

Por otro lado, podemos encontrar el resto de información que disponemos en el previo registro de los datos que hemos facilitado al gimnasio. Podemos cambiar datos como el nombre, apellidos, correo electrónico, número de contacto, dónde vives, pero no es posible cambiar otro tipo de datos relacionados más con la cuenta o con el proceso de pago como puede ser el documento nacional de identidad o pasaporte, o el usuario que son dos cosas inmutables, así como la contraseña de acceso a la aplicación que deberá realizarlo indicándoselo al personal del gimnasio en la recepción.

Otra opción de la que disponemos es presionando el botón de Mis reservas.



Figura 29: Mis reservas y la opción para reservar.

Lo primero que encontramos en esta ventana como se puede observar en la figura 28 son las reservas que ya disponemos en un apartado que dispone de un *scroll* vertical por si no caben en nuestra pantalla. En caso de no tener ninguna reserva aparecerá sin mostrar ningún elemento de la lista y con una frase como “No hay registros de ninguna reserva para esta cuenta”.

Por otro lado, tenemos a nuestra disposición la posibilidad de hacer la reserva de las distintas clases que imparta el gimnasio sin la necesidad de tener que llamar previamente para que nos registren y nos den información de las clases disponibles.

Antes de presionar el botón de Añadir reserva, podemos elegir la futura visualización del horario de las clases.

Tenemos la opción de ordenar por clase, la cual muestra una ventana previa al horario como la que se muestra en la figura 30.

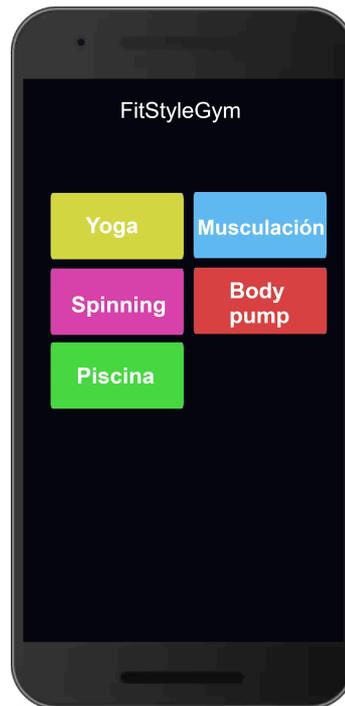


Figura 30: Listado de clases que oferta el gimnasio.

Una buena forma de ordenar el horario y como ya hemos podido comprobar en el capítulo dos de esta memoria, para el cliente es más intuitivo el filtro de clases en el horario para que directamente pueda elegir la hora a la que desea asistir sin la necesidad de perderse en el probable solape de las distintas clases que se realicen en distintas aulas a las mismas horas.

Simulando que hemos presionado en *Spinning* pasamos a la siguiente ventana que podemos ver en la figura 31.

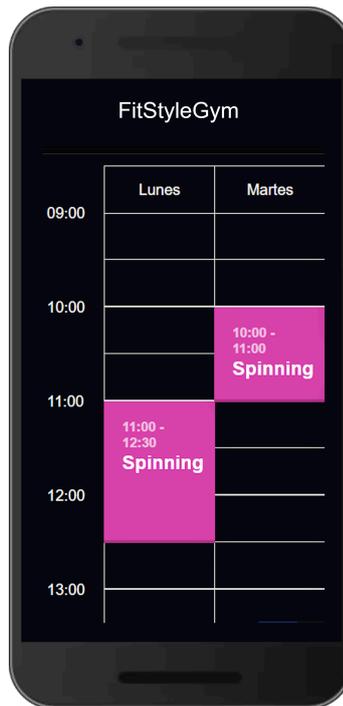


Figura 31: Horario filtrado por clases.

Como podemos ver en la figura 31, aquí ya hemos llegado al horario de las clases el cual muestra dónde se imparte aquella opción que hemos elegido previamente, su duración y si hay o no plazas para poder realizar nuestra reserva.

Al hacer clic en el horario donde exista una de las clases que se mostrarán en el mismo color que en el listado, ya que este se guarda en la base de datos, nos preguntará si queremos reservar y si aceptamos volveremos a la ventana de la figura 29 para verificar que se ha añadido la reserva.

Volviendo al funcionamiento de la ventana de reserva en la figura 29, podemos observar un *toggle* el cual hemos utilizado para ordenar por clase. En caso contrario, si utilizamos la opción de ordenar por horario, nos saltaremos la ventana de la figura 30 e iremos directamente a lo que muestra la figura 32.



Figura 32: Horario sin filtro de clases.

Como se puede observar en este caso, aparecen todas las clases que el gimnasio tiene disponible y que el cliente podrá reservar ordenadas por día y hora. De la misma forma que en el paso anterior, las clases se pueden distinguir de una forma fácil por sus colores.

A gusto del cliente podrá elegir la opción de reserva, ya que lo más cómodo que hemos podido comprobar es primero realizar la búsqueda de la clase y posteriormente su reserva. Esto de todas formas, como se explicó en la sección 2.2, muchas aplicaciones usan un horario similar al que se puede observar en la figura 32, aunque pensamos que no es muy práctico ya que si el gimnasio ofrece distintas clases a la misma hora, aparece solapado o se repiten las horas, hay gente que puede estar acostumbrado a este método de búsqueda y por eso hemos implementado las dos opciones, hay que pensar que si el cliente se siente más cómodo, mejor expectativas tendrá del gimnasio y querrá renovar sin pensárselo.

Por último, hemos añadido lo que nos ha parecido curioso que muchos gimnasios tienen.

Hay que pensar que los gimnasios viven de que los clientes se matriculen y estén bien a gusto y motivados para seguir matriculándose y yendo mes a mes, por lo tanto, muchos ofrecen una serie de regalos que se pueden conseguir

mediante unos puntos que van adquiriendo por distintos mecanismos, ya sea por acceder al gimnasio, por asistir a las clases o por alcanzar ciertos objetivos que se planteen.



Figura 33: Mis puntos y listado de recompensas.

Como podemos observar en la figura 33, hemos realizado un listado que previamente el gimnasio nos ha facilitado, indicando cuántos puntos son necesarios para adquirir los premios y recompensas. Asimismo, el cliente puede ver cuántos puntos ha almacenado hasta la fecha y puede gastárselos presionando en cualquier recompensa de la lista siempre que estén activos y que los puntos almacenados sean superiores al ítem seleccionado.

Posterior a este paso, se le indicará que debe dirigirse al mostrador para recibir la recompensa y se le descontarán los puntos de su cuenta.

Como hemos mencionado en contadas ocasiones y para terminar este punto, este es el prototipo que podemos usar para enseñar a los gimnasios y ofrecerles nuestra aplicación, pero cabe destacar que cada gimnasio puede tener distintas necesidades y funcionalidades ya programadas que no quieran perder. Por todo ello y junto a la API que nos permitirá realizar nuestras modificaciones con su sistema ya funcionando previamente, podemos realizar cualquier cambio que sea necesario.

7. Conclusiones

Ya estamos llegando al final de este proyecto y por ello podemos destacar que estamos satisfecho del resultado de este. El proyecto se orientó a no solo obtener una experiencia cerrada en cuanto a conceptos siendo escalable en su progreso y siendo una opción válida para un sector que cada vez está en auge. Por esto fue importante realizar una buena planificación y pensar bien paso a paso que, y como queríamos realizar los trabajos, así como una exhaustiva investigación de que tenían ya desarrollado los gimnasios.

A la hora de elegir la tecnología de las que se compone este proyecto, hemos valorado tanto aquellos conocimientos de los que ya disponíamos y podíamos aprovechar para explotar al máximo nuestras capacidades y por las tecnologías que están en expansión y nos permiten una escalabilidad a medida que el proyecto crezca.

No tenemos la menor duda de que lo elegido para este proyecto ha sido un completo acierto y cabe destacar de que realizarlo para *smartphones* ya que prácticamente todo el mundo lleva uno en el bolsillo, como ya se mencionó al principio de este proyecto, el prototipo se ha realizado para Android, pero gracias a la tecnología que hemos utilizado, con un par de cambios se podría volver a exportar para iOS o para cualquier sistema operativo que exista en el mercado de móviles. Esto no sería sido posible si el código hubiese sido programado en el lenguaje de cada uno de los distintos dispositivos que existen en el mercado, principalmente porque necesitaríamos unos conocimientos de los que no disponemos y probablemente más personal, añadiendo la problemática de que no sería tan escalable como ahora y que para cualquier cambio de versión, sería necesario programarlo en todas las plataformas que dispongan de nuestra aplicación, hablando como mínimo de los tan usados sistemas como pueden ser Android e iOS.

8. Ampliaciones y futuros trabajos

Por último, hay que añadir que después de la realización de este proyecto y adquirir más habilidades tanto de programación como de las tecnologías que íbamos utilizando, a pesar de que hemos logrado los objetivos propuestos desde un inicio, se han planteado mejoras a lo largo de la realización de este.

De primeras pensamos crear el sistema completamente desde cero sin incluir las posibles funcionalidades que ya tiene el gimnasio, esto encarecería el precio inicial de la propuesta, pero es verdad que podemos obtener una aplicación más extensa y planificada desde el inicio, así como auto gestionable por su parte y menos intrusiva para nosotros. Para este proyecto, los nuevos cambios que se deseen realizar deben tener una base ya hecha con anterioridad o pedirnoslo a nosotros para realizarlos. Pensamos que una buena opción sería dotarles de una intranet donde poder gestionarlo. Añadir un pequeño panel donde dar de alta a los clientes, las clases, los horarios y todas aquellas funcionalidades que se manejan desde la aplicación y que pueden ser susceptibles a cambios.

También sería de valorar una posible automatización para que el cliente mediante tarjeta pueda renovar la cuota mensual, trimestral o anual. Esto sería susceptible a ofertas que se le pueden aplicar a los ya clientes que dispongan de la aplicación y que el gimnasio pueda tener una administración que permita su puesta en marcha o modificación sin tener que pasar por nosotros.

Si tenemos operaciones con tarjeta también podríamos añadir junto a los puntos una opción para poder comprar aquellas cosas que se ofertan o no como regalos y que el gimnasio disponga de una tienda online.

Como la tecnología utilizada para la realización de este proyecto es orientada a páginas web, un trabajo futuro podría ser, a petición del gimnasio, crear la aplicación para que los clientes puedan hacer sus gestiones mediante la web y utilizando un ordenador, ya que desde casa es posible que, si la aplicación va adquiriendo más funcionalidades, los clientes quieran disponer de un mayor control pudiendo iniciar sesión desde un ordenador.

Como futuro trabajo se le puede proponer al gimnasio disponer de un sistema de estadísticas para los propios clientes mediante las rutinas previamente seleccionadas, así como planes generales o personalizados de pago, para aquellas proposiciones por los cuales cada uno decide ir al gimnasio bien sea adelgazar, muscular o prepararse para cualquier tipo de deporte. Esta podría contar desde dietas o ejercicios propuestos a un seguimiento de peso y progresión en los ejercicios.

Estas son alguna de las ampliaciones a este proyecto o futuras propuestas y trabajos, pero como está orientado este proyecto, es el gimnasio el que dispone de unas necesidades y nosotros nos adaptamos a cada uno de ellos.

9. Bibliografía

- [1] «zonamovilidad,» octubre 2021. [En línea]. Available: <https://www.zonamovilidad.es/avances-tecnologicos-y-su-impacto-en-la-sociedad.html>.
- [2] «Smartphones vs computers,» octubre 2021. [En línea]. Available: <https://www.perficient.com/insights/research-hub/mobile-vs-desktop-usage#:~:text=Mobile%20devices%20drove%2061%25%20of,increase%20from%2063.3%25%20in%202019..>
- [3] «Aplicaciones móviles vs páginas webs,» octubre 2021. [En línea]. Available: <https://www.ttandem.com/blog/el-futuro-de-los-sitios-web-y-las-aplicaciones-moviles/>.
- [4] D. v. S. E. R. J Sutherland, The Power of SCRUM, CreateSpace, 2011.
- [5] J. & G. F. Cervantes Gómez, Taxonomía de los modelos y metodologías de desarrollo de software más utilizados, 2020.
- [6] «Apps Gym,» octubre 2021. [En línea]. Available: <https://programasb2b.com/top/software-gimnasio>.
- [7] «ActivaClub horarios,» noviembre 2021. [En línea]. Available: <https://activaclub.es/horarios-de-actividades/>.
- [8] «CDM reservas,» noviembre 2021. [En línea]. Available: <http://reserva.centrodeportivomoncada.com/>.
- [9] R. J. V. D. V. & J. P. M. Granados, Programación en Capas, 2012.
- [10] «Generador de UML,» enero 2022. [En línea]. Available: <https://app.diagrams.net>.
- [11] «PHP,» enero 2022. [En línea]. Available: <https://www.php.net/manual/es/index.php>.
- [12] «MySQL,» enero 2022. [En línea]. Available: <https://dev.mysql.com/doc/apiphp/en>.
- [13] A. S. a. t. D. o. N.-b. S. Architectures, «API Rest,» enero 2022. [En línea]. Available: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
- [14] «Material Icons,» diciembre 2021. [En línea]. Available:



<https://fonts.google.com/icons>.

- [15] «Fontawesome,» diciembre 2021. [En línea]. Available: <https://fontawesome.com/icons>.
- [16] «2021,» diciembre. [En línea]. Available: <https://cordova.apache.org/>.
- [17] «Ionic,» diciembre 2021. [En línea]. Available: <https://ionicframework.com/>.
- [18] «GitHub,» enero 2022. [En línea]. Available: <https://github.com/>.
- [19] «Heroku,» enero 2022. [En línea]. Available: <https://www.heroku.com/pricing>.
- [20] «Visual Studio Code,» diciembre 2021. [En línea]. Available: <https://code.visualstudio.com/>.
- [21] «OAuth 2,» [En línea]. Available: <https://oauth.net/2/>.

10. Anexo: ODS

Los objetivos de desarrollo sostenible marcan unas pautas y tratan unos puntos necesarios para conseguir un futuro sostenible para todos. En la propia página web de las Naciones Unidas podemos encontrar hasta 17 objetivos para transformar el mundo y que se relacionan entre sí e incorporan desafíos globales a los que nos enfrentamos día a día, como la pobreza, la desigualdad, el clima, la degradación ambiental, la prosperidad, la paz y la justicia. Estos puntos en los cuales los países pertenecientes a las Naciones Unidas como en España, el país en el que nos encontramos, han adoptado una serie de medidas para implementar un plan de acción de la conocida **Agenda 2030**.

No vamos a poder contribuir en la vacunación salvo con información veraz ni tampoco terminar con el hambre en el mundo, pero ¿Qué podemos hacer para ayudar?

Todos podemos empezar por promover y proteger nuestra propia salud y la de nuestro entorno, tomando decisiones bien informadas, practicando relaciones sexuales seguras y vacunando a nuestros hijos.

Podemos unirnos y exigir a nuestros gobernantes que tienen el poder de realizar los cambios de todo un gran país a un compromiso por la salud de las personas. Es su deber proporcionarnos los conocimientos y los medios para poder lograr un gran avance entre todos, por ello, a través de las escuelas, clubes, equipos y organizaciones tenemos la necesidad de promover una mejor salud para todos y en especial aquellos más vulnerables.

Por todo ello creemos que podemos aportar nuestro granito de arena y con ello entramos de lleno en el tercer objetivo titulado <<salud y bienestar>>.

Se trata de garantizar una vida sana y promover el bienestar en todas las edades.

Actualmente, el mundo se enfrenta a una crisis sanitaria sin precedentes; la COVID 19 está propagando el sufrimiento humano, desestabilizando la economía

mundial y cambiando drásticamente las vidas de miles de millones de personas en todo el mundo.

Por esto pedimos que con precaución hagas lo posible para no contagiar ni ser contagiado, pero, no por ello debes quedarte aislado en casa y continuar con una vida sedentaria que solo lleva a tener más problemas en el futuro.

La obesidad infantil es algo que preocupa mucho, dado que su curva parece que va incrementando debido a la comida rápida y los ultras procesados, así como la vida que tiende a ser más cómoda y sedentaria. Desde los colegios se intentan tomar medidas para que todos los niños tengan unas horas a la semana de gimnasia para que empiecen a tener un hábito que debe ser trasladado a la memoria para el resto de la vida.

Nosotros creemos que justo es ahí donde podemos ayudar al mundo de forma directa incluyendo facilidades en la rutina de los ejercicios y acceso a unas buenas y saludables dietas. También de forma indirecta haciendo posible que terceros como son los gimnasios y disponen de las herramientas y el espacio donde la gente puede realizar ejercicios de fuerza, tan importante y que se está popularizando últimamente incluso más que el antiguo trabajo de cardio que se recomendaba para casi todo. Ofrecerle las herramientas necesarias para llegar a todo el mundo y que sea fácil de manejar para que cualquier persona y a cualquier edad puedan comprenderla y utilizarla para que no tengan ningún tipo de excusa para no realizar ejercicio.

Por todo ello se ha realizado este proyecto, motivado con poder ayudar a las personas en su salud y bienestar de forma adecuada, mantenerlos motivados para que se convierta en una rutina y no en algo pasajero como los propósitos que realizamos el 1 de enero y que después de una semana parece que no nos acordamos.

Esperemos que con nuestra aplicación podamos aportar al mundo ese impulso que necesitamos para mejorar y que la gente se conciencie que el ejercicio es muy importante para el ser humano.

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

