

## Treball de Fi de Grau

## GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques Universitat de Barcelona

# GORILLA COACH, UNA APLICACIÓ PER A ENTRENADORS PERSONALS (iOS + API)

## **Jabel Riera Soler**

Director: Santi Seguí Realitzat a: Departament de Matemàtica Aplicada i Anàlisi, UB

Barcelona, 27 de juny de 2016

## Taula de continguts

Agraïments	4
ESTRUCTURA DE LA MEMÒRIA	5
	-
0.INTRODUCCIÓ	6
RESUM	6
D'ON SORGEIX LA IDEA I ON VA ENCAMINADA	6
Breu estudi de Mercat	6
1.ANÀLISIS	8
GIMNASOS I ENTRENAMENTS: TOT UN MÓN!	8
Tecnologia	9
REQUERIMENTS	10
Figures	11
FROM ANDROID TO IOS	12
OBSERVACIONS TÈCNIQUES	12
2.PLANIFICACIÓ	14
3.DISSENY FUNCIONAL	16
ΑΡΙ	16
Routing	16
Αριιζαζιό iOS	19
STORYBOARDS	19
CONTROLADORS ESPECÍFICS	20
MVC	20
4.DISSENY GRÀFIC	22
Introducció	22
Menús	24
VISTES I FLUXES	26
1- ENTRENAMENTS	27
2- Exercicis	30
3- Màquines	32
4- CALENDARI	34
5- CLIENTS I MISSATGES	35
6- REGISTRE I LOGIN	38
7- VARIES	41
8- ENTRENAMENTS	44
5.IMPLEMENTACIÓ	48
ΑΡΙ	48
ASPECTES GENERALS	48
BUSINESS LOGIC	52
IOS	53
CICLES DE VIDA A IOS	54
IOS PROTOCOLS	55
CRIDES A LA API	56
6.CONTINUÏTAT DEL PROJECTE	58
7.CONCLUSIONS	60

## Agraïments

Al acabar la defensa del Treball que ara mateix esteu llegint davant del tribunal corresponent, acabarà si tot va bé una etapa molt important de la meva vida. Si miro enrere i intento mirar amb vista d'ocell els anys que han transcorregut a la facultat es veuen d'una hora lluny moltes persones i és a tots ells a qui vull dedicar aquesta secció.

En primer lloc vull agrair a la meva família tot el suport que m'han donat en tot moment i totes les seves empentes que en els moments més difícils m'han fet seguir lluitant, canviar de perspectiva o enfocar les coses d'una altra manera. En certa manera han compartit amb mi durant molts trimestres l'estrès de les temporades d'exàmens i m'han facilitat les coses tan com ha sigut possible.

També m'agradaria dedicar unes línies a la meva parella. Si algú ha patit i m'ha recolzat quan se'm ajuntava feina de l'empresa, de la universitat i del màster junts ha sigut ella. Ha convertit estones realment dures en moments preciosos de recordar i poder compartir aquest camí amb ella superant les dificultats colze amb colze ha sigut una experiència inoblidable.

No puc passar pas per alt tampoc la inestimable ajuda rebuda per part del gimnàs In 20 Minutes de Barcelona, especialment del Doctor Ricard Cánovas i l'entrenador Víctor Cánovas, que m'han ajudat molt a definir les necessitats reals així com les interfícies i característiques que serien més d'ajuda per a un entrenador personal.

Com no podia ser d'una altra manera estic també eternament agraït amb tot el claustre professorat de la UB. Sempre han tingut un buit per donar-me un cop de mà o explicar-me en hores fora de consulta la típica cosa que et perds mentre segueixes la trajectòria d'una mosca a classe (o altres coneixements que no entens per la seva dificultat, es clar).

I per acabar, m'agradaria també agrair l'ajuda i guia que he rebut per part del tutor del meu treball, Santi Seguí, que m'ha tret de més d'un penya-segat sense sortida durant els mesos que ha durat aquest treball i va acceptar ajudar-me fins i tot en una matèria que no és pròpia de cap departament de la UB com és la programació per a dispositius iOS.

Moltes gràcies a tots de tot cor

## Estructura de la memòria

Per a cobrir tots els apartats necessaris per a entendre aquest projecte he dividit la memòria en els següents apartats:

- 1. Introducció: Breu explicació del perquè he escollit aquest projecte, l'escenari en el que neix i els objectius que s'han fixat dins del marc de Treball de Final de Grau.
- 2. Anàlisis: Apartat dedicat a l'estudi més detallat dels requeriments concrets de l'aplicació, la tecnologia triada per a assolir-ho i un primer anàlisis inicial de l'aplicació.
- 3. Planificació: Planificació mitjançant diagrames de Gant del projecte a desenvolupar.
- 4. Disseny funcional: Cobreix tant el disseny de l'aplicació d'iOs com el de la API dissenyada per a comunicar la aplicació amb el servidor de dades.
- 5. Disseny gràfic: Patrons seguits per a dissenyar una interfície d'usuari intuïtiva, responsable i que segueixi els patrons d'usabilitat necessaris per a l'entorn on està enfocada.
- 6. Implementació: Estudi en detall dels principals protocols i els handicaps durant la fase de desenvolupament.
- 7. Continuïtat del projecte: Situació de la part desenvolupada dins el marc global de l'aplicació a comercialitzar.
- 8. Conclusions
- 9. Bibliografia

## 0. Introducció

## Resum

Gorilla Coach és una aplicació per a entrenadors de gimnàs que els permet cobrir totes les necessitats que tenen en la seva feina del dia a dia. Per una banda permet que organitzin amb total llibertat les rutines d'entrenament, màquines i exercicis amb que treballen i entrenen els seus clients. I per altra banda els permet gestionar els clients, un calendari d'entrenaments, totes les seves estadístiques, un canal de comunicació privat i dur a terme sessions d'entrenament amb dues modalitats diferents.

## D'on sorgeix la idea i on va encaminada

Gorilla Coach va sorgir de la petició d'un gimnàs que freqüento des de fa ja més de 7 anys. Des del principi cronometren els clients amb cronòmetres convencionals i anoten els resultats en fulles personalitzades per a cada client on caben les dades de només 4 sessions. Un cop finalitzades les 4 sessions anoten els resultats de la última com si fossin els primers del següent full.

Evidentment per a tenir estadístiques reals del progrés dels clients el sistema que fan servir actualment té moltes mancances i ja feia temps que volien una APP per a poder registrar tots aquests progressos en un lloc centralitzat, amb persistència de dades i un format en que aquestes fossin analitzables fàcilment i llegibles amb gràfiques. Com que al gimnàs tots van amb iPhone la decisió de la plataforma sobre la que fer la APP inicial va ser senzilla, tot i que tinc planejat fer la versió Android un cop finalitzada la iOs ja que la APP serà comercialitzada externament al gimnàs en qüestió.

És per això que durant l'etapa de planificació i anàlisis de les necessitats no només he tingut en compte els requeriments i opinió que em van transmetre els propietaris i entrenadors del gimnàs sinó també la d'entrenadors externs que treballen pel seu compte amb els que he treballat per a saber quines coses serien un *must-have* dins de l'aplicació.

## Breu estudi de Mercat

Abans d'entrar en l'estudi de mercat de les aplicacions per a coachs en si hem d'entendre que hi ha 2 tipus de coachs o entrenadors actualment. Per una banda hi ha els entrenadors que treballen en gimnasos amb una nòmina fixa i per altra banda hi ha els coachs que entrenen pel seu compte.

Aquest segon grup mai s'ha endinsat en el món de les APPs mòbils ja que el desenvolupament d'un sistema pràctic per a la feina que fan és una solució tan personalitzada i complexa que econòmicament no els sortia a compte.

Per altra banda algunes firmes de gimnasos si que han intentat endinsar-s'hi invertint una gran quantitat de diners en aplicacions que després quedaven ràpidament desactualitzades i amb un mal funcionament degut a la falta d'inversió per el baix R.O.I. (Retorn de la inversió) que els donaven.

L'estudi de comercialització de la APP no pertany a aquest document però el model que es seguirà per a la comercialització serà SAAS amb uns preus directament lligats al nombre de clients que tingui el Coach/gimnàs. D'aquesta manera el ROI és immediat i sempre positiu tant per a petits entrenadors com per a firmes grans de gimnàs amb milers de clients.

## 1. Anàlisis

## Gimnasos i entrenaments: Tot un món!

Així doncs, com sol passar quan t'enfrontes a un projecte d'un àrea totalment diferent a teva, descobreixes a mesura que avança la fase de disseny que és tot un món amb un munt de particularitats que en més o menys mesura afectaran el desenvolupament del projecte. O com a mínim podem dir que haurien d'afectar-lo ja que si orientem un projecte a un àrea s'espera que sigui coherent amb tot el que això suposi.

I em vaig trobar de repent amb el meu entrenador personal des de ja fa casi 8 anys parlant de fer una aplicació per a gestionar els entrenaments del seu gimnàs. La idea no era dolenta, parlant de preus van veure que era inaccessible fins i tot per un gimnàs d'alt standing com el seu, però precisament això va ser el que em va fer obrir els ulls a la oportunitat que tenia davant.

Els devia molt a nivell personal per tot el suport que m'havien donat els darrers anys, no només a nivell d'estat físic sinó també personal i vaig decidir tirar-ho endavant. D'entrada comptava amb professionals del sector altament reconeguts, amb mètodes d'entrenament propis i llibres al mercat que em podrien assessorar en cualsevol moment sobre qualsevol dubte que tingués i orientar-me al respecte. I per l'altra, si aconseguia planejar bé el projecte podria cobrir una part del mercat d'aplicacions mòbils que a dia d'avui encara segueix mancant degut a la impractibilitat per raons de pressupost que tenen els gimnasos de tenir la seva pròpia APP.



La aplicació no estaria disponible només pels seus clients, per tant els entrenaments havien de ser gestionables, cada entrenador podria tenir les seves pròpies rutines, els seus propis exercicis i fins i tot les seves pròpies màquines. D'aquesta manera aconseguiríem tenir impermeabilitat en les dades dels coach per a que confiin en Gorilla Coach. Cap entrenador faria servir una aplicació mitjançant la resta d'entrenadors pogués plagiar-li les seves tècniques.

Els clients ja no serien només els del gimnàs, per tant seria necessari implementar un sistema de control d'usuaris, sessions, i permisos a nivell general de l'aplicació. A més a més, molts entrenadors no tenen una base de dades dels seus clients com el gimnàs en qüestió, per tant seria de gran ajuda i un plus per a vendre el servei que des de la pròpia aplicació els entrenadors poguessin donar d'alta clients, gestionar els pròxims entrenaments i fins i tot comunicar-se amb els clients.

I per si tots aquests canvis fossin pocs, el mètode de cronometratge inicial proposat pel gimnàs va haver de dividir-se en dos per cobrir el fet que no tots els entrenadors tenen les mans lliures mentre entrenen els seus clients.

De la nit al dia la aplicació havia canviat totalment, i encara no havia començat a planificar res (per sort). Arribats a aquest punt em vaig dir a mi mateix que si es comercialitzava l'aplicació el millor seria tenir també una APP d'android i d'aquí va sorgir la idea de desenvolupar la aplicació com a APP + API per a futura escalabilitat del sistema Gorilla Coach i poder realitzar una APP Android i una Web 100% compatibles i en un temps molt més curt al tenir gran part de la business lògic ja desenvolupada.

Així doncs, implementant una API que fos qui es comuniqués amb la Base de Dades se simplificava tot molt: passava a ser un projecte factible.



Estructura general APP + API

### Tecnologia

Estava bastant clar que la part de iOS la desenvoluparia en Swift. És un llenguatge totalment nou per a mi, però donada la seva semblança a Java que li han donat a Apple i les poques ganes que tenia de tornar-me a ficar en el món de objective-c modificat al gust d'Apple era la millor opció per a l'aplicació.

En la decisió dels servidors i la implementació de la API la decisió tampoc va tenir molta complexitat. Tinc contractats varis servidors en diferents datacenters d'Espanya i Europa i estic acostumat a treballar amb PHP per als serveis que ofereixo als meus clients en la meva feina professional. Així doncs la API es desenvoluparia amb PHP sobre un servidor CentOS 6 i fent servir MySQL com a suport per a la persistència de dades. No tenia experiència en el desenvolupament de APIs en PHP, però gràcies a l'assignatura de Software distribuït si que havia pogut tractar amb APIs restful de primera ma. En el cas de la meva app hi

hauria sessions pel mig per tant no entraria dins del que s'entén com a restful però els principis de Routing bàsics propis d'aquestes aplicacions ja els tenia assolits i el handicap vindria només per a veure com s'implementaven en PHP.

Així doncs ja teníem els escollits a dur a terme la missió: Una APP per a dispositius iOS programada tant íntegrament com fos possible en Swift i una API programada en PHP, tot sobre un servidor CentOS i fent servir com a motor de bases de dades MySQL:



### Requeriments

Podem veure doncs que la APP té un abast sempre ampliable i caldrà definir molt bé els límits establerts per a aquest treball. Com a base serà indispensable dissenyar l'estructura de la base de dades i desenvolupar la API, juntament amb el desenvolupament de la aplicació d'iOS amb les següents funcionalitats:

- Ha de permetre a un coach:
  - Entrenaments, Exercicis i Màquines:
    - Crear, visualitzar, modificar i eliminar entrenaments
    - Crear, visualitzar, modificar i eliminar exercicis
    - Crear, visualitzar, modificar i eliminar màquines
    - Sincronitzar totes les dades amb la base de dades online
  - Esdeveniments
    - Crear nous esdeveniments
    - Visualitzar esdeveniments existents
    - Entrenar clients a partir dels esdeveniments
  - o Clients
    - Donar d'alta nous clients
    - Visualitzar clients
    - Donar clients de baixa
    - Veure estadístiques dels clients
    - Comunicar-se amb els clients per xat/missatges
  - o Entrenar:
    - Entrenar en temps real als clients guardant totes les dades de l'entrenament
    - Permetre registrar entrenaments en mode "free hands" un cop finalitzat l'entrenament

El client per part seva tindria una APP per a veure totes les seves estadístiques i comunicar-se / organitzarse amb el seu entrenador però aquesta app no la cobrirà aquest treball i es farà a posteriori.

### **Figures**

En aquest apartat descriuré breument les necessitats de cada "objecte" amb que treballarem així com les dependències existents entre ells i les accions que s'han de poder realitzar amb ells més detallades.

#### Entrenaments

Els entrenaments hauran de tenir un nom propi i un identificador únic que els vinculi a l'entrenador que els ha creat. Hauran de guardar un conjunt d'exercicis propis també creats per mateix entrenador i s'hauran de poder visualitzar, editar, re-anomenar o eliminar en qualsevol moment. S'hauran de sincronitzar amb la base de dades mitjançant la API i haurà de ser possible veure'n la llista completa i visualitzar els detalls de qualsevol d'ells.

#### Exercicis

Són necessaris per a crear entrenaments i a l'hora depenen directament de les màquines. Els exercicis, al igual que els entrenaments van vinculats a l'entrenador que els crea i consten d'ún nom, una descripció opcional i una referència a la màquina amb que es realitzen. També s'han de sincronitzar a través de la API i s'han de poder llistar, visualitzar i eliminar en qualsevol moment.

#### Màquines

De les tres figures que conformen un entrenament aquesta és la més bàsica i només té dependència del coach que la crea, però tan exercicis com entrenaments depenen d'ella. Consten d'un nom i una imatge (opcional) per a facilitar la labor de l'entrenador. De la mateixa manera que els entrenaments i exercicis, s'ha de poder llistar, visualitzar i eliminar.

#### Clients

Els clients poden ser donats d'alta per els coach però hauran de confirmar les seves dades a través de l'aplicació i establir la seva pròpia contrasenya. Per a registrar-se necessitaran el codi de confirmació que es generarà quan el coach els doni d'alta. Han de poder ser llistats i eliminats en qualsevol moment i se'ls pot vincular a esdeveniments (sessions d'entrenament)

#### Esdeveniments

Els esdeveniments vinculen un coach amb un client (seu) en una relació única amb una data i hora concrets. També es poden guardar notes per a l'entrenament o el gimnàs en que es durà a terme. S'han de poder visualitzar des del calendari i han de servir per a poder crear els accessos directes de propers entrenaments per a un ús fluit i intuïtiu de l'aplicació.

#### Entrenament (Històric)

L'històric d'entrenaments ha de guardar els resultats de cada sessió d'entrenament per a poder mostrar després estadístiques de cada client. Un històric d'entrenament té una llista d'exercicis realitzats que a la seva vegada contindrà un seguit de series cadascuna amb els resultats de les repeticions realitzades.

#### Missatges

Els missatges són intercanvis de text curt entre un entrenador i un client. No poden ser borrats ni editats, només creats i mostrats i guarden la informació de si el receptor els ha llegit o no (No com a acusació de recepció sinó per a mostrar si un usuari té missatges entrants sense llegir).

## From Android to... iOS

Si alguna cosa he après molt més del que m'esperava durant el Treball de Final de Grau aquesta ha sigut sense cap mena de dubte Swift i iOS. Al decidir fer la API coneixia el llenguatge per una banda i la lògica d'operar de les APIs per una altra ja que com he comentat anteriorment ho havíem tractat a Software Distribuit. Per tant la feina en el cas de la API va ser únicament juntar 2 coses que ja coneixia. En el cas de la aplicació per a iOS programada en Swift, en canvi, tot era nou. No coneixia ni Swift ni els cicles de vida de les aplicacions, vistes, etc

Com que ja sabia que hauria de ser en iOS abans de començar vaig fer un curs online<sup>1</sup> per a poder fer les estimacions de temps una mica més acurades i no patir tant durant el desenvolupament de la APP. Com era d'esperar el curs cobria els punts més bàsics i ensenyava a utilitzar les components més estàndards del sistema operatiu i llenguatge i a l'hora de desenvolupar la APP hi vaig haver de posar molta imaginació i molts consells de comunitats online que han suposat una ajuda increïble sense la qual segurament no hagués pogut acabar ni la meitat de coses, o com a mínim, no amb la qualitat que considero que té al final la APP tan a nivell visual com intern.



Al final he de dir que la experiència ha sigut agradable i sobretot molt satisfactòria a nivell personal. Hi ha hagut moments de desesperació i d'incertesa però mica en mica quan el producte anava agafant forma i jo anava agafant soltesa amb Swift i això m'ha permès polir la APP fins al resultat final presentat.

### **Observacions tècniques**

Si una cosa m'ha sobtat de Swift ha sigut el canvi radical que ha pres Apple al canviar de Objective-C a Swift els últims 2 anys. I no ho dic per haver canviat de llenguatge sinó perquè Objective-C i Swift són dos llenguatges totalment diferents. Sense anar més lluny, tot i que es considera un llenguatge fortament tipat en quant a variables i la seva dependència directe de Objective-C (iOS segueix estant programat en Objective-C internament) en la declaració de variables no cal especificar el tipus de dades.

Amb Swift, des d'Apple han volgut posar a l'abast del desenvolupador un llenguatge més senzill que actua com a intermediari entre Cocoa/Cocoa Touch i Objective-C i el resultat, a jutjar per la meva experiència

<sup>&</sup>lt;sup>1</sup> Curs online de www.udemy.com

personal és més que satisfactori. La majoria de tasques són molt més senzilles i menys tedioses que si s'haguessin de fer en Objective-C.

A més a més, hi ha la possibilitat de fer servir codi en Objective-C dins la aplicació programada en Swift. En el meu cas no ha sigut necessari en cap secció tot i que si que la llibreria que vaig necessitar per implementar el menú lateral està programada íntegrament en Objective-C en lloc de Swift



Com es pot veure en la imatge van dissenyar el llenguatge orientat a augmentar la productivitat dels programadors sense que es veiés afectat el rendiment de l'aplicació. De fet, en varis benchmarks realitzats Swift era més ràpid que Objective-C tot i està basat sobre aquest últim.

## 2. Planificació

Título	Esfuerzo	Día T
▼ 1) Anàlisis de viabilitat	5s 2d ▼ 6,5h	
• 1.1) Anàlisis inicial de requisits	3s	
▼ 1.2) Reunions	2d 1,5h	▼ <b>Ě</b>
<ul> <li>1.2.1) Reunió In 20 Minutes (Dr. Ricardo Cánovas)</li> </ul>	4h	E
<ul> <li>1.2.2) Reunió amb diferents coachs de l'H Barceló</li> </ul>	otel 4h	E
<ul> <li>1.2.3) Reunió In 20 Minutes (Dr. Ricardo Cánovas)</li> </ul>	5,5h	E
<ul> <li>1.2.4) Reunió amb diferents coachs de l'H Barceló</li> </ul>	otel 4h	E
• 1.3) Anàlisis de mercat i competència	1s	
<ul> <li>1.4) Recopilació de les necessitats recolectado</li> </ul>	des 1d	i i i i i i i i i i i i i i i i i i i
<ul> <li>1.5) Redacció de casos d'ús</li> </ul>	3d 5h	
<ul> <li>1.6) Estudi final de viabilitat i restricció de l'al del projecte</li> </ul>	bast 1d	
<ul> <li>Planificació i disseny</li> </ul>	5s 4d 1h	▼ ¥
2.1) Mock-ups de l'aplicació	3s	
• 2.2) Reunió In 20 Minutes (Dr. Ricardo Cánov	as) 3h	
• 2.3) Reunió In 20 Minutes (Dr. Ricardo Cánov	as) 3h	
• 2.4) Reunió In 20 Minutes (Dr. Ricardo Cánov	as) 3h	Ĭ
• 2.5) Disseny de la BBDD	4d	
2.6) Disseny de la API	1s 4d	
▼ 3) Desenvolupament	15s 3d 3h	• J
• 3.1) Configuració de l'entorn de treball i servi	dor 4h	i i i i i i i i i i i i i i i i i i i
• 3.2) Desenvolupament de la BBDD	7h	Ĕ
<ul> <li>3.3) Programació i configuració base API</li> </ul>	4d	Ĕ.
▼ 3.4) Programació API	4s	• J
3.4.1) Control d'errors	1d	E Contraction of the second se
▼ 3.4.2) Base usuaris i sessions	1s	▼
3.4.2.1) Funció Register	1d	8
• 3.4.2.2) Funció Login	1d	E C
<ul> <li>3.4.2.3) Funció ResetPassword</li> </ul>	1d	E C
• 3.4.2.4) Funció EditProfile	1d	E C
3.4.2.5) Funció CheckSession	1d	8
▼ 3.4.3) Funcions 'New'	1s 1d	▼ □
<ul> <li>3.4.3.1) Funció New client</li> </ul>	1d	E C
3.4.3.2) Funció New training	1d	E C
3.4.3.3) Funció New exercise	1d	3
3.4.3.4) Funció New machine	1d	C
<ul> <li>3.4.3.5) Funció New event</li> </ul>	1d	3
<ul> <li>3.4.3.6) Funció New Training History</li> </ul>	1d	C.
▼ 3.4.4) Funcions 'Get'	1s 3d	▼ □ □ □ □ □
3.4.4.1) Funcio 'updated' (Sync)	1d	C III
• 3.4.4.2) Funció get calenar	1d	C
3.4.4.3) Funció get clients	1d	8
3.4.4.4) Funció get exercises	1d	C
3.4.4.5) Funció get trainings	1d	E .
3.4.4.6) Funció get machines	1d	2
• 3.4.4.7) Funció get stats	1d	E
<ul> <li>3.4.4.8) Funció get prefs</li> </ul>	1d	



El recompte final d'hores aproximades dedicades a anàlisi, desenvolupament, test i documentació al final queda així:

Anàlisi	75h
Desenvolupament	470h
Test	40h
Documentació	50h

Com era d'esperar la fase a la que s'ha dedicat més temps és la de desenvolupament, que inclou també unes 80h d'aprenentatge especialment d'iOS i Swift. En les etapes d'anàlisi s'han comptat també les hores de reunions amb l'equip del gimnàs **In 20 Minutes** i altres entrenadors personals lliures.

## **3. Disseny Funcional**

### API

A grans trets la API consta d'una base de dades i la business logic pera controlar-la i servir-la. Internament hi ha 3 protagonistes principals dins la API:

- **Routing**  $\Rightarrow$  *index.php*
- **Business Logic**  $\Rightarrow$  *functions.php*
- **Base de dades**  $\Rightarrow$  gorillac-coach.sql

### Routing

Per al disseny del routing de la API he fet servir un microframework PHP que permet definir funcions d'enrutament personalitzades de manera ràpida, segura i controlada. El microframework en qüestió és *"Flight - An extensible micro-framework for PHP"* que m'ha facilitat també la creació de funcions personalitzades per al retorn de valors de la API.



Així doncs, l'arxiu *index.php* és el que s'encarrega de rebre les peticions, decidir si són o no correctes, i en cas que siguin correctes estructurar les dades rebudes i delegar-les a l'arxiu *functions.php*. També és aquest arxiu el que s'encarrega d'informar a l'usuari si es dóna algun error durant el procés intern de *functions.php* o de confirmar les dades registrades o valors de retorn en cas que tot sigui correcte.

Les crides a API poden contenir peticions GET, POST o GET | POST i additivament en algunes funcions s'ha requerit de la recepció d'un arxiu JSON Object, especialment per a les funcionalitats de nous registres (usuaris, entrenaments, exercicis, etc). En tots els casos és l'*index.php* el que estructura les dades rebudes ja siguin per URL o per JSON Object abans d'enviar-les a *functions.php*.

#### Base de Dades

La base de dades es podria dividir conceptualment en 2 blocs principals i un auxiliar. Els dos blocs principals serien les taules referents a l'emmagatzematge de dades d'entrenaments i les taules referents als usuaris. Les taules auxiliars són 3 i serveixen per a diferents funcions auxiliars. A continuació en detall per blocs començant per les taules referents als entrenaments:



#### A continuació les taules referents als usuaris:



I finalment les tres taules auxiliars que serveixen per al control d'errors, la recuperació de contrasenyes i fer broadcasting de notícies entre els usuaris registrats en cas que sigui necessari:

erro	rs			2
$\gg$	id_error	int(11)	P	$\stackrel{d}{=}$
	code	int(11)	P	$\stackrel{\rm de}{=}$
	esca	varchar(200)	P	${}^{\rm d} {}^{\rm b}$
	eses	varchar(200)	P	$\stackrel{\rm de}{=}$
	enen	varchar(200)	P	$\stackrel{{}_{\scriptscriptstyle \oplus}}{=}$
	Add field			

Þ	id_pass_token	int(11)	6	$\frac{1}{2}$
	id_user	int(11)	<pre>s</pre>	$\stackrel{\rm d}{=}$
	token	varchar(250)	P	$\stackrel{\rm d}{=}$
	expiry	datetime	P	$\stackrel{{}_{\scriptscriptstyle \oplus}}{=}$

nev	vs			7
Þ	id	int(11)	6P	$\oplus$
	message	varchar(100)	P	÷
	publish_date	timestamp	P	$\stackrel{\Delta}{=}$
	for_whom	int(11)	P	$\frac{\Delta}{\nabla}$
+	Add field			

## Aplicació iOS

L'estructura d'un projecte iOs en certa manera, evidentment, pot ser comparada amb la d'un projecte Android, però internament hi ha certs canvis molt importants a tenir en compte:

#### Storyboards

A iOs les vistes no es declaren individualment sinó que s'organitzen en Storyboards si programem amb XCode i Swift. Equivalen als antics arxius .XIB però a diferència d'aquests, un Storyboard representa tot un conjunt de vistes que mantenen alguna relació entre elles. Generalment s'organitzen per seccions de l'aplicació o en cas de incloure funcions complexes amb diferents vistes aquestes també podrien organitzar-se en un Storyboard a part.

<ul> <li>StoryBoard</li> <li>LaunchScreen.storyboard</li> <li>Trainings.storyboard</li> <li>Main.storyboard</li> <li>OfflineTraining.storyboard</li> </ul>	Gorilla Coach utilitza 10 Storyboards diferents (Imatge esquerra) i com podeu veure estan agrupats per diferents seccions de l'aplicació.
<ul> <li>Exercises.storyboard</li> <li>Machines.storyboard</li> <li>RealTimeTraining.storyboard</li> <li>Login.storyboard</li> <li>SmartDevices.storyboard</li> <li>Settings.storyboard</li> </ul>	Els Storyboards es poden referenciar per codi i es pot navegar entre les vistes que contenen sense cap mena de problema ni restricció. Per a dur a terme un canvi de vista entre diferents Storyboards a iOS farem el següent:

let myStoryBoard : UIStoryboard = UIStoryboard(name: "RealTimeTraining", bundle: nil)

let destinationSB = myStoryBoard.instantiateViewControllerWithIdentifier
("RealTimeTrainingViewController") as! RealTimeTrainingViewController
self.presentViewController(destinationSB, animated: true, completion: nil)

### **Controladors específics**

Un altre aspecte que canvia molt en iOs si es ve d'Android és com es controlen i s'opera amb els diferents objectes que contenen les vistes. Si que és cert que pots declarar el 100% de la interfaç per codi i en aquest cas la diferència no seria tant gran, però si dissenyes les vistes amb Storyboard com recomanen des de la web d'Apple Developer hi ha tot un seguit de passos a seguir abans que pots treballar amb un objecte de la vista.

Per ser més concrets, diguem que a Android crees una taula dinàmica. L'únic que hauries de fer des dela classe que controli aquella vista és referenciar un objecte del tipus tableview a partir de la ID que tu mateix has assignat a l'XML i ja podries treballar-hi. Amb iOS, en canvi, la cosa cambia molt. Per fer servir una taula a iOS s'han de seguir els següents passos:

- 1. Afegir la taula (UITableView) a la vista de l'storyboard que volguem.
- 2. Afegir una classe concreta que representi un exemple de cel·la de la taula\*
- 3. Vincular la cel·la a la classe recent creada
- 4. Vincular la taula a la vista amb una @IBOutlet ( @IBOutlet weak var taula: UITableView!)
- 5. Indicar a la taula qui serà el proveïdor de les dades (Normalment "self". Es pot fer gràficament també)
- 6. Haurem de declarar a la classe controladora que extén de UITableViewDataSource i UITableViewDelegate

\*En aquesta classe han de constar tots els elements de la cel·la que volem poder referenciar per codi (botons, etiquetes, subvistes, etc) però si hi ha algun element estàtic que no canvia depenent de l'estat de la cel·la es pot no incloure.

Un cop fet tot això ja podrem treballar amb les dades de la taula d'una manera similar a com es fa a Android tot adaptant als cicles de vida de les vistes a iOS.

D'una manera similar, moltes altres components natives de iOS hauran també de seguir un procés semblant. Algunes d'aquestes components necessàries per a Gorilla Coach són:

- UIView
- Tab Bar Controller
- Navigation Bar
- Page View Controller
- Switch
- Date Picker
- Picker View
- Etc

### MVC

De la mateixa manera que a la API hem decidit que l'estructura seria MC ja que no necessita vista a la aplicació en iOS seguirem un patró MVC.

El **model** serà una base de dades interna *SQLite* que ens facilitarà tenir la persistència de dades d'una manera fàcil i ràpidament accessible. També farem servir la memòria que el sistema posa al nostre abast com a *System Preferences* per emmagatzemar

El **controlador** estarà format per una única classe (*controller.swift*) que serà com pertoca la que es comunicarà tant amb totes les classes que constitueixen la vista de la aplicació com amb la base de dades interna i, afegit degut a l'ús de la API, serà també qui s'encarregarà de comunicar-se amb la API, codificar i descodificar els JSON de resposta i els JSON Objects necessaris per a les crides, etc.

I per últim, la **vista** estarà formada per els Storyboards + les classes de les vistes que contenen.

En total el projecte té 75+ classes Swift. Les classes pertanyents a una mateixa vista estan agrupats dins de directoris virtuals i totes les classes controladores de vistes tenen una nomenclatura indentificativa clara per a una comprensió fàcil de la seva finalitat.

## 4. Disseny Gràfic

## Introducció

Abans d'entrar en detall a veure el disseny de cada vista cal especificar que en quant al disseny de l'aplicació s'ha tingut en compte des d'un primer moment i s'ha dissenyat colze amb colze amb l'equip d'entrenadors del gimnàs **In 20 Minutes** de Barcelona. És per a això, que a més dels típics *mockups* o *sketches* que hagi fet a paper els he agut de digitalitzar per anar-los presentant prototips en les diferents reunions que figuren al diagrama de Gantt fins a trobar un disseny funcional.



Realitzar aquestes reunions per dissenyar el disseny de l'aplicació em va permetre realitzar dos correccions importants en la interfície gràfica incloent finalment dos seccions que jo no havia tingut en compte:

						ne R	temore N	
				Gorilla Coach.sketch Edite		e name		
Ο.		n (i) - Q	+ 🥥 🔟 (5	0 0 0				
Insert		proup Create Symbol 30%				Exercice name	Add u	
Pages	+	-1.000 -800 -600 -400 -200 0	200 400 600 1	800 1.000 1.200 1.400 1.600	1.800 2.000 2.200	Machine name	- W	
Main Storyboard	8		Escollir exercicis Offline Free Training					
Login-Register			Offine Training			Exercice name	Add 2	
Offline Training	8					The machine name	The second se	
Trainings List			Machine name Add					
IOS GUI	8		A Exercice same			Exercice name	Add	
Offline Training			Machine name Remove			The second se		
Next Exercise			Dercke name Att			Cancel. Save trainin	15	
<ul> <li>Offline Training</li> </ul>	0		Machine name		P 74			
icons/back			Exercice name Remove					
All. Here you ca	1e 8	Offine Training			Insert Exercise	Trainings List	Next Exercise	
Rectangle 4	13	Offline Training	Machine name Add		om	+++ Vodatory E5 6:18 AM	64% IND Annual Vodatore ES	18 AM 64%
Bar / Status	BA g		73			Training Poution	+ Both Por	ture Training
T button/red/t	ig .	without having to constantly	Machine name		- Serie 1	trainings routines	Douy Pos	cure training
An text	8	chronometer your clients.	L Dercice name		Repetitions			
— bg	9		Con Martine come			Jose's Training	*** Prensa	(Pierna)
w button/red/t	9	6 6 5	Cancel Save training			Midcone hame	Last tim	e: Monday
An text	08					a second second		
— bg			Trainings List	Next Exercise	N N	Cardio Training 1	···· A Direct	
Topber	8		Trainings Routines +	Body Posture Training		Internine name	Last tim	er New
Aa Offine Tr	ai		A Jose's Training	A second X		100	141	
— bg	8		Machine name	Last time: Monday		Jabel's Training		
GORILLA		Free Training	Cardio Training 1			The machine name	A Sa Pector	al i Triceps
bg .	8		Machine name	Listtime:Nov 🗸			Last tim	e: Monday
<ul> <li>Escollir exercicis Of</li> </ul>	lin e <sup>2</sup> -	Choose Training from List	Juber's Training		Save exercise	Shoulder & Legs Training		
Bar / Status	Ba 8			Last time: Monday		a statistic statistic	10 <sup>(10)</sup>	
topber	2		Shoulder & Legs Training					
- battom			A. Barla Barlan Talalan	Dersal		n Iraining		
All Cancel	9		Machine name	100				
Ad Save trai	ing			Abdominals ×				
button	2.000			Lincome Monday				
— bg								
person 1	99			Fed Topicy Preschet				
T person 2	-1-			Contracting Contraction				
- elements	<sup>di</sup> 8							
Q Filter	1 13							

La primera i més important d'aquestes dues és el mode *offline* o *hands free*. Inicialment l'aplicació només anava a tenir el mode *real time* per capturar els temps i dades en general dels subjectes entrenats. En la segona reunió es va veure que això no era pràctic ja que en certes persones necessiten tota la atenció i les dos mans per a entrenar-la i és a posteriori quan anoten tots els resultats. Així doncs vaig dissenyar una interfície per a que poguessin registrar entrenaments seleccionant totes les dades un cop aquest hagi acabat.

La segona cosa que jo no havia tingut en compte i que em van demanar que inclogués va ser un accés directa per a la creació de tot tipus de coses. Bàsicament volien poder crear qualsevol entrenament, exercici, màquina, client o esdeveniment amb un sol click en compte d'haver d'anar a la secció corresponent i crear a "New Client", per exemple. A més a més volien que estigués disponible des de tota la interfície principal.

La petició tenia lògica així que vaig investigar-ho i finalment vaig trobar una manera d'incloure-ho sense que molestés. Inicialment m'havia plantejat posar-ho al menú lateral. D'aquesta manera no estava visible i no molestava si no es necessitava. Però la sol·lució que em va venir al cap després era molt més optimitzada i no tenia lògica executar-la des del menú. Vaig decidir incloure els accessos directes en un Custom UIAlertView.





## Menús

Es pot accedir a les diferents vistes que conformen l'aplicació a partir de dos menús: Menú lateral d'entrenador i Menú (barra) inferior de clients.

El **menú lateral** es desplega fent swipe del cantó esquerra al cantó dret o fent clic a l'icone de menú superior. Per a tancar-lo s'ha de fer el procediment invers, o bé swipe de dreta a esquerra o click de nou al botó de menú. Aquest menú conté accés a totes les vistes que fan referència a la gestió dels programes de coaching: **rutines d'entrenament**, **exercicis**, **màquines**. També és des d'aquesta secció des d'on es pot **tancar la sessió** i accedir a la **configuració** del perfil propi per a canviar les dades personals com el *nom*, *cognom*, *contrasenya* o *fotografia*.



El **menú inferior** conté, per la seva banda, els accessos a totes les vistes de gestió d'usuaris com són el **calendari**, els **missatges** o els **clients** en si. El menú inferior és accessible des de totes les vistes que el conformen per poder navegar entre elles amb més facilitat. Funciona com les *tabs* o pestanyes, amplament implementades no només en plataformes mòbils sinó també a nivell web. La vista actual en la que s'està marca l'icone inferior amb el to blau per defecte de gorilla coach.



## Vistes i Fluxes

En aquesta secció comentaré més detalladament què conté cada vista, com s'ha estructurat i d'on ve i on va l'usuari quan passa per elles. L'índex total de vistes de la Gorilla Coach:

- Entrenaments
  - Llista d'entrenaments
  - o Configurar entrenament
  - Vista / Edició entrenament
- Exercicis
  - o Llista d'exercicis
  - o Nou exercici
- Màquines
  - o Llistat de màquines
  - o Nova màquina
- Calendari
  - o Calendari
  - Nou esdeveniment (Sessió d'entrenament)
- Clients
  - o Llistat de clients
  - o Nou client
  - o Estadístiques

#### • Registre

- Vista benvinguda
- o Login
- o Pantalla Registre
- o Tria d'usuari
- $\circ$  Registre (Part 1 )
- $\circ$   $\,$  Registre (Part 2 )
- o Imatge de Perfil
- $\circ$  Settings

#### Varies

- o Main (Vista principal)
- o Menú Lateral
- o Container Sync Api
- $\circ \quad \text{Container Sync Offline} \\$
- o Contenidor Menú inferior

### 1- Entrenaments

#### 1.1 - Llista d'entrenaments

#### Descripció

En aquesta vista es poden veure els entrenaments que ha creat un entrenador (Els seus propis). Cada entrenament té un títol, una imatge d'una de les màquines que es fan servir i 'l'últim cop que s'ha fet servir aquella rutina d'entrenament.

#### Es pot venir de...

La vista de la llista d'entrenaments és una de les més usades de l'aplicació. S'hi pot accedir des del menú lateral per a la seva administració però es reutilitza cada cop que s'entrena algun client ja sigui amb mode *offline* o *realtime* sempre i quan s'esculli la modalitat de realitzar un entrenament a partir d'una rutina ja guardada.

#### Es pot anar a...

Aquesta vista pot redirigir depenen de d'on es vingui a molts llocs diferents. Si es ve del menú lateral i per tant s'estan administrant les rutines d'entrenament llavors pot redirigir a les vistes de crear un nou entrenament o visualitzar/editar un entrenament. Si en canvi s'està entrenant algú, portarà a les vistes de pre-entrenament dels Storyboards de training offline i realtime respectivament. També pot tornar en tots els casos a la vista anterior.

#### Addicionalment...

A més a més, es podrà re-anomenar un entrenament o eliminar-lo de la llista fent swipe sobre l'element desitjat si es ve del menú lateral. La funció de reanomenar es realitza dins d'un Custom AlertView programat específicament per a aquesta tasca. Si en algun moment es borren tots els trainings i no hi ha dades a mostrar la UITableView s'ha programat per a mostrar un missatge informatiu i facilitar a l'usuari un camí pel que seguir sense que pugui arribar a pensar que és algun error i s'ha quedat en blanc.

•००० vodato	ne ES 🗢 13:04	6 🖲 23 % 🌅
<	<b>Trainings Routines</b>	+
16781	Rutina1	
-	Label	
1	Rutina2	
H	Label	
11 1	Rutina1.1	
	Label	
4	Rutina2.1	
(a)	Label	
<b>T</b>	TrainingTest	
1 miles	Label	
1 90	TrainingTest	
E.	Label	
L A	TrainingTest	
	Label	
85 0	TrainingTest	



•০০০ vodafone ES 😤	13:16	€ @ 31% <b>⊑</b> →+
<	Trainings Routines	+
It seems ti	Oops! nat there are no train Add raining	nings yet.

#### 1.2 - Configurar entrenament

#### Descripció

En aquesta vista es poden configurar nous entrenaments o editar els exercicis que pertanyen a un entrenament ja creat. La vista presenta una UITableView amb tots els exercicis disponibles i un botó d'afegir/eliminar depenent de si ja està afegit a la rutina que estem creant/editant o no. En cas de fer canvis i clicar a "Cancel" o el botó d'anar enrere, el sistema recordarà si els volem guardar abans de sortir. Si estem editant un entrenament aquest s'actualitzaria (en cas de clicar "Save Training") i en cas que fos una rutina d'entrenament nova es demanaria el nom. Es pot veure tot aquest procés a les captures de la part inferior.

#### Es pot venir de...

Es pot arribar a aquesta vista o bé creant un entrenament nou o bé editant-ne un d'existent, per tant, des de les vistes *Llista d'entrenaments* i *Vista / Edició entrenament.* No s'hi pot arribar des de la Llista d'entrenaments si no s'està administrant entrenaments. És a dir, si s'està a la vista d'entrenaments només per a triar un entrenament a realitzar a un client l'única opció possible serà escollir entrenament o tornar enrere, en cap cas editar l'entrenament allà mateix.

#### Es pot anar a...

Des d'aquesta vista es pot retornar a la vista *Llista d'entrenaments* quan es crea un entrenament nou o a la vista *Vista / Edició entrenament* si s'està editant un entrenament.

#### Addicionalment...

En cas de no existir cap exercici es mostraria un missatge molt semblant al que apareix a la llista buida d'entrenaments, guiant al coach per a que en pugui afegir de nous. Si no es fa cap canvi (en cas d'estar editant un entrenament) o no s'afegeix cap exercici (en cas d'estar-ne creant un de nou) si es clica a "Cancel" o es torna enrere no es preguntaria si es volen.







#### 1.3 Vista / Edició entrenament

#### Descripció

Com bé indica el propi nom, aquesta vista permet visualitzar un entrenament. És a dir, mostra el nom, juntament amb la llista d'exercicis i les seves corresponents imatges. Permet editar-lo per afegir-ne o treure'n (Crida vista *Configuració entrenament*).

#### Es pot venir de...

A aquesta vista si pot arribar de dues maneres diferents: La primera és clicant a un entrenament de la vista *Llista d'entrenaments* i la segona és al tornar de la pantalla d'edició de la rutina d'entrenament (Vista *Configuració entrenament*).

#### Es pot anar a...

Un cop més, i seguint els fluxes ja comentats en les dos vistes anteriors, des d'aquesta vista es pot accedir a la vista de **Configuració entrenament** o be retornar a la vista **Llista d'entrenaments** un cop s'hagi acabat d'editar la rutina d'entrenament actual.

#### Addicionalment...

Els botons de eliminar entrenament (Les x dins de cada exercici) i el botó d'afegir exercicis apareixen quan es clica el botó superior dret de la barra de navegació "Edit". Un cop en mode d'editor es poden realitzar les operacions comentades i un cop finalitzat es pot clicar l' "OK" pera mostrar la vista normal de nou. Quan es guarden canvis a l'entrenament es mostra un avís notificant si s'ha guardat correctament en color verd i de color vermell en cas que sorgeixi algun problema.

•ooo vodatone	ES 🗢 16:12	<b>€</b> @ 82 % <b>■</b> +
<	Training	Edit
100	Exercise1	
T	Label	
1. 20	Exercise2	
Ę,	Label	

••ooo vodafone	ES 🗢 16:12	<b>6</b> 🛞 82 % 💶 🕨
<	Training	ОК
	Exercise1 Label	×
H	Exercise2 Label	×
	Add Exercises	

Volver a Ajustes	16:37	L @ 94% 🛑
<	Training	OK
x	Training updated successfully	
	Exercise1 Label	
	Exercise2 Label	×
	Exercise3 Label	×
	Add Exercises	

### 2- Exercicis

#### 2.1 - Llista d'exercicis

#### Descripció

En aquesta vista es poden veure els exercicis que ha creat un entrenador (Els seus propis). En aquest apartat només es mostra el títol de cada exercici. El perquè d'aquesta decisió es comenta a l'apartat **El que falta per venir**. En aquesta vista l'entrenador pot administrar els exercicis a partir dels quals crearà les seves rutines d'entrenament i permet crear, eliminar o re-anomenar exercicis. Per a la visualització d'exercicis d'un entrenament ja existeix la vista *Veure / Editar entrenament*.

#### Es pot venir de...

Es pot arribar a aquesta Vista únicament des del menú lateral, fent polsant sobre "Exercicises" ja que porta a una vista d'administració pròpia únicament del coach.

#### Es pot anar a...

Aquesta vista pot retornar cap a la pantalla d'ininci *Main* des de la que s'hi ha accedit o be dirigir cap a la vista *New Exercise*.

#### Addicionalment...

Com a totes les vistes que inclouen alguna llista d'elements dins de l'aplicació, la llista d'exercicis mostra un missatge en cas que no n'hi hagi cap per a que l'usuari no es confongui pensant que és un error de l'aplicació.

🚥 vodatone ES 🗢	9:20	د 🛞 49 % 🌄 :
	Exercises	+
Exercice 1		
Exercice 2		-
Exercice 3		
Exercice 4		
Exercice 5		

andstone ES 😤	9:21	<b>6</b> /0 ≪ ■
<i>☆</i>	Exercises	+
Exercice 1		
Exercice 2		
	D	elete Rename
Exercice 4		
Exercice 5		***



#### 2.2 - Nou exercici

#### Descripció

Aquesta és la vista que el coach utilitza per a crear nous exercicis. Mostra els camps necessaris a emplenar amb text i un selector per escollir la màquina (ja existent) relacionada amb l'exercici que s'està donant d'alta.

#### Es pot venir de...

Només es pot arribar a aquesta vista des del botó + de la *Llista exercicis* ja que és també una vista purament d'administració del material del coach.

#### Es pot anar a...

Des d'aquesta vista només es pot retornar a la vista anterior de la que es ve (*Llista exercicis*). En cas d'haver creat correctament un exercici al tornar enrere ja estarà disponible a la llista.

#### Addicionalment

Aquesta vista té també control de canvis dels camps utilitzats per a donar d'alta els exercicis, de manera que si s'introdueixen un *Nom d'exercici* o una *Descripció* i es fa l'intent de tornar enrere la aplicació et recordarà que hi ha canvis sense guardar. Els camps del formulari, a més són navegables amb dues fletxes sobre el teclat i en aquesta mateixa barra es mostra el placeholder del camp que s'està editant en aquell precís moment (Es pot veure a la imatge inferior central).

🚥 vodafone ES 🗢	9:26	د 🖲 54 % 💶 🕫
<	New Exercise	
New Exerci	se Title	
Exercise Description	on (Optional)	
Enter some descriptio	n	
Machine		
	Màquina 1 Màquina 2	
	Màquina 3	
	Màquina 4 Màquina 5	
Cancel	Sa	ve Exercise





### 3- Màquines

#### 3.1 - Llistat de màquines

#### Descripció

En aquesta vista es poden veure les màquines que ha creat un entrenador (Les seves pròpies). En aquest apartat només es mostra el títol i imatge de cada màquina. En aquesta vista l'entrenador pot administrar les màquines podran donar d'alta després els exercicis que formen les rutines d'entrenament. Permet crear, eliminar o re-anomenar màquines. És la única vista que permet la visualització individual de la llista de màquines ja que només són una informació de suport sobre la feina que fa l'entrenador.

#### Es pot venir de...

Es pot arribar a aquesta Vista únicament des del menú lateral, fent polsant sobre "Machines" ja que porta a una vista d'administració pròpia únicament del coach.

#### Es pot anar a...

Aquesta vista pot retornar cap a la pantalla d'ininci *Main* des de la que s'hi ha accedit o be dirigir cap a la vista *New Machine*.

#### Addicionalment

Com a totes les vistes que inclouen alguna llista d'elements dins de l'aplicació, la llista d'exercicis mostra un missatge en cas que no n'hi hagi cap per a que l'usuari no es confongui pensant que és un error de l'aplicació. La reanomenació de màquines es dur a termes com en les altres vistes que hem vist mitjançant una Custom AlertView.







#### 3.2 - Nova màquina

#### Descripció

Aquesta és la vista que el coach té disponible per a crear noves màquines. En el cas de les màquines, al ser elements secundaris dins de l'aplicació (Podria haver funcionat tot sense màquines) només tenen Nom i imatge per a que siguin visualment identificables.

#### Es pot venir de...

Només es pot arribar a aquesta vista des del botó + de la *Llista màquines* ja que és també una vista purament d'administració del material del coach.

#### Es pot anar a...

Des d'aquesta vista només es pot retornar a la vista anterior de la que es ve ( *Llista màquines* ). En cas d'haver creat correctament un exercici al tornar enrere ja estarà disponible a la llista.

#### Addicionalment

Aquesta vista té també control de canvis dels camps utilitzats per a donar d'alta els exercicis, de manera que si s'introdueixen un *Nom d'exercici* o una *Descripció* i es fa l'intent de tornar enrere la aplicació et recordarà que hi ha canvis sense guardar. Tot i que només té un camp el formulari, es veu com a les vistes anteriors el placeholder a la barra superior afegida al teclat. La imatge se selecciona accedint a les fotografies del dispositiu.



Per motius de privacitat he substituït les imatges de la meva galeria per unes de caràcter general.

### 4- Calendari

#### 4.1 Calendari

#### Descripció

En aquesta vista es pot navegar per un calendari complet per a visualitzar els esdeveniments d'un coach (Els seus propis). Es pot fer swipe a sobre del calendari per a navegar per els mesos i anys. Per a tals accions també hi ha disponibles 2 fletxes just a cada lateral del títol del mes que s'està visualitzant, sota la Navigation Bar. Per a visualitzar els events d'un dia només s'ha de seleccionar el dia en qüestió. Quan un dia té esdeveniments (sessions d'entrenament) es marca amb un punt sota el número del dia a la vista del mes.

#### Es pot venir de...

Es pot arribar a aquesta vista des de qualsevol de les vistes principals de l'aplicació (*Main, Clients, Missatges*) mitjançant la barra de navegació inferior.



#### Es pot anar a...

Aquesta vista pot dirigir a la de creació d'un **Nou esdeveniment** si es clica a "New Event" o a qualsevol de la resta de vistes principals navegant-hi amb la barra de navegació inferior: **Main, Clients, Missatges.** Com en totes les vistes principals es pot anar també a les vistes de **Configuració entrenament, Nova màquina, Nou client, Nou exercici** i (repetit) **Nou esdeveniment** fent clic a la última de les opcions començant per l'esquerra del menú inferior i seleccionant la opció desitjada.

#### 4.2 Nou esdeveniment

•••• vodat	fone ES 1	7	10:05		C @	80 %
<		N	lew Eve	nt		
<		jur	nio, 20	16		>
DOM	LUN	MAR	MIÈ	JUE	VIE	sAa
19	20	21	22	23	24	25
Label						
Client						
		(	Client	3		
Time						
				14		
		10	(	05		
Gym		11				
Cy.						
		SA	VE EVE	NT		

#### Descripció

En aquesta vista podem crear noves sessions d'entrenament. Per a crear un nou esdeveniment es necessita saber la data, hora, client i opcionalment el gimnàs en que tindrà lloc la sessió d'entrenament. Tots els elements es poden escollir a partir d'un calendari en vista de setmana superior i 2 pickers. L'únic cas en que s'ha d'escriure és si es vol indicar el gimnàs.

#### Es pot venir de...

Només es pot arribar a aquesta vista des del botó *New event*" del *Calendari* o mitjançant l'accés directe que tenen totes les vistes principals.

#### Es pot anar a...

Des d'aquesta vista només es pot retornar a la vista anterior de la que es ve (*Calendari*). En cas d'haver creat correctament un exercici al tornar enrere ja estarà disponible a la llista i es mostra un avís informant-ne.

### 5- Clients i Missatges

#### 5.1 - Llistat de clients

#### Descripció

Aquesta vista representa la llista de clients que un coach té assignats. Es visualitza la llista mostrant una *fotografia* del client i el seu *nom* juntament amb la data de l'última vegada que se'ls ha entrenat.

#### Es pot venir de...

Es pot arribar a aquesta vista des de qualsevol de les vistes principals de l'aplicació ( *Main, Calendar, Missatges*) mitjançant la barra de navegació inferior.

#### Es pot anar a...

Aquesta vista pot dirigir a la de creació d'un **Nou client** si es clica a "New Client" o a qualsevol de la resta de vistes principals navegant-hi amb la barra de navegació inferior: **Main, Calendar, Missatges.** Com en totes les vistes principals es pot anar també a les vistes de **Configuració entrenament, Nova màquina, Nou exercici, Nou esdeveniment** i (repetit) **Nou client** fent clic a la última de les opcions començant per l'esquerra del menú inferior i seleccionant la opció desitjada.

#### Addicionalment

Com a totes les vistes que inclouen alguna llista d'elements dins de l'aplicació, la llista d'exercicis mostra un missatge en cas que no n'hi hagi cap per a que l'usuari no es confongui pensant que és un error de l'aplicació..



#### 5.2 - Nou client

#### Descripció

En aquesta vista es poden donar d'alta nous clients. Quan es dóna d'alta un nou client es facilita un codi de registre. Aquest és el codi que el client haurà d'introduir en el primer pas de registre com a client per a que se'l pugui vincular al coach corresponent. Els camps de *nom* i *cognom* són opcionals per al registre ja que el client els haurà d'introduir igualment durant el procés que dugui a terme per registrar-se. La diferència és que en cas d'emplenar-se el client ja veurà el seu nom i cognom en els placeholder dels respectius camps de text de les vistes de registre.

#### Es pot venir de...

Només es pot arribar a aquesta vista des del botó *New client*" de la *Llistat de clients* o mitjançant l'accés directe que tenen totes les vistes principals.

#### Es pot anar a...

Des d'aquesta vista només es pot retornar a la vista anterior de la que es ve (*Llistat de clients*). En cas d'haver creat correctament un exercici al tornar enrere ja estarà disponible a la llista i es mostra un avís informant-ne.

#### Addicionalment

Aquesta vista té també control de canvis dels camps utilitzats per a donar d'alta els clients, de manera que si s'introdueixen un *Nom* o Cognomi es fa l'intent de tornar enrere la aplicació et recordarà que hi ha canvis sense guardar. Els camps del formulari, a més són navegables amb dues fletxes sobre el teclat i en aquesta mateixa barra es mostra el placeholder del camp que s'està editant en aquell precís moment (Es pot veure a la imatge inferior central).

<	New	client			
Registration	Code:				
4	41	2	3	8	
This is the reg	sistration code so he/she can	you have t complete r	o give to egistrat	your nev	v
Name					
Last name					

•000	vodafon	eES 3	G	12	:52		L	@ 91	% 💼
<				New	client				
Reg	gistrat	ion C	ode:						
	Л	Λ		1	~	<b>)</b>	S	Q	
	4	4		1	2	- '	0	0	
Th	is is the	e regist	ratio	n code	you ha	ive to	give to	your n	iew
		0112 00	1102 01	ine corri	compre		sisciario		
Max									
Na	ne Velient								
Nev <	ne Vdient			New	client's	name			ОК
Nev Nev	ne Volient > el	's nam		New	client's	name		no	ок
Nev <	velient > el W	e	r	New	client's a y	name	i	no O	ок
Nar Nev C	el W	e d	r	New t	client's a y h	name u j	i k	no 0	ок р ñ
Nar Nev < Q a	el volient el w s	e d	r f	New t g	client's a y h	u j	i k	no O I n	ок p ñ
Nar Nev C	el S	e d : x	r	New t g	client's a y h v l	name u j b	i k n r	no O I n	ок р ñ



#### 5.3 - Missatges

#### Descripció

En aquest conjunt format per dues vistes el coach pot comunicar-se amb els seus clients i viceversa. Està dissenyat per ser un canal alternatiu als convencionals (Whatsapp, SMS, Telegram, Trucacdes) i que d'aquesta manera el coach pugui tenir tots els temes relacionats amb la feina en una mateix lloc centralitzat.

#### Es pot venir de...

Es pot arribar a aquesta vista des de qualsevol de les vistes principals de l'aplicació ( *Main, Calendar, Clients*) mitjançant la barra de navegació inferior.

#### Es pot anar a...

Aquesta vista pot dirigir a la vista de missatges d'un client si es clica a la cel·la de la taula corresponent o a qualsevol de la resta de vistes principals navegant-hi amb la barra de navegació inferior: *Main, Calendar, Missatges.* Com en totes les vistes principals es pot anar també a les vistes de *Configuració entrenament, Nova màquina, Nou exercici, Nou esdeveniment* i (repetit) *Nou client* fent clic a la última de les opcions començant per l'esquerra del menú inferior i seleccionant la opció desitjada.

#### Addicionalment

Com a totes les vistes que inclouen alguna llista d'elements dins de l'aplicació, la llista d'exercicis mostra un missatge en cas que no n'hi hagi cap per a que l'usuari no es confongui pensant que és un error de l'aplicació..





		16:33				% 💻
	(					6
El joan o	com ho	porta?	)			
			Ho			
luevo me	ensaje				E	nviar
El W E	R	ТҮ	U	1	0	Ρ
E W E S D	R F	T Y G H	U J	і к	O L	P Ñ
El W E S D Z	R F X (	T Y G H	U J B	I K	O L V	P Ñ
	El joan o	El joan com ho	El Joan com ho porta?	El joan com ho porta?	El joan com ho porta? Ho porta	El joan com ho porta? Ho porta be

### 6- Registre i Login

El registre té 9 vistes diferents amb només 2 possibles fluxes interns per tant aquesta secció l'estructuraré una mica diferent mostrant directament els 2 fluxes possibles. A continuació llisto les 8 vistes que conformen aquesta secció:

- 1. Vista benvinguda
- 2. Vista tria login/registre
- 3. Login
- 4. Pantalla Registre
- 5. Tria d'usuari
- 6. Registre (Part 1)
- 7. Registre (Part 2)
- 8. Imatge de Perfil
- 9. Settings

#### 6.1 - Registre

El registre complet es dur a terme amb un total de 9 vistes des que s'obra l'aplicació fins que l'usuari està registrat i amb la sessió iniciada. Aquestes 3 primeres vistes donen la benvinguda a l'usuari i li mostren la opció d'iniciar sessió en cas que ja tinguin un compte registrat o registrar-se si és el primer cop que utilitzen el servei. Si l'usuari decideix registrar-se la aplicació li demanarà que esculli entre registrar-se com a entrenador o com a usuari del servei que ja disposa d'un entrenador.



Un cop escollit el tipus d'usuari se li demanaran les dades de registre al llarg de 2 vistes, cadascuna amb 3 camps per a poder recollir el *correu electrònic, nom d'usuari, contrasenya* per una banda (dades usuari) i *nom, cognom i nif (opcional)* per una altra banda (dades persona). En cas que l'usuari s'estigui registrant com a client abans d'aquests dos passos se li sol·licitarà que introdueixi el codi que el seu coach personal li hagi facilitat per tal de poder-los vincular. Un cop verificat el codi introduït el procés és idèntic per als dos usuaris (visualment parlant)



Arribats a aquest punt ja es tenen totes les dades necessàries i únicament falta mostrar un resum de la informació per a si es vol corregir alguna dada i donar la opció de pujar al servidor una imatge de perfil per al seu usuari.







#### 6.2 - Login

Si ja es té un compte registrat al servidor per a utilitzar el servei, el procés es simplifica moltíssim. La pantalla de benvinguda es mostra igualment si és el primer cop que s'obre la APP després de que aquesta sigui instal·lada, però a la següent vista només s'ha de seleccionar *"Sign In"* que portarà fins a la tercera vista on només s'ha d'introduir el *nom d'usuari* o *correu electrònic* i la *contrasenya* escollida durant el procés de registre.



### 7- Varies

En aquesta secció tractaré 5 vistes que tot i que apareixen ben sovint a tots els fluxes comentats anteriorment no tenen una secció pròpia o mereixien ser explicats per separat. De les 5 vistes 2 són molt bàsiques: *Main* i *Menú Lateral*. Les altres 3 són contenidors i per tant no tenen la mida de pantalla d'un mòbil sinó només una àrea de la pantalla reservada per a poder-hi generar contingut dinàmicament i incloure'l des d'altres vistes. Començarem comentant la vista de *Main* i *Menú Lateral*:

#### 7.1 - Main (Vista principal)

Com bé indica el nom és la vista principal de l'aplicació. Un cop registrat i amb la sessió iniciada aquesta és la primera vista que apareixerà sempre en pantalla quan Gorilla Coach s'inicii. És el centre de tots els fluxes de vistes possibles de l'aplicació. Des d'ella es pot accedir als dos menús que porten a les diferents vistes d'administració i gestió d'entrenaments, exercicis i màquines i tota la secció d'usuaris, esdeveniments, missatges, etc. També permet accedir a la configuració del compte d'usuari, tancar sessió i la resta de vistes ja comentades anteriorment.

La seva organització és relativament senzilla i molt neta per a facilitar en tot moment que el coach pugui fer el que vulgui amb 2 o 3 clics màxim i minimitzar així el temps que mira a la pantalla enlloc del client al que està atenent.

Trobarem doncs una taula amb tots els següents esdeveniments que tenim programats, un botó per a poder iniciar un nou entrenament no programat en qualsevol moment i els dos menús per a la resta de gestions. Les vistes dels menús les hem comentat anteriorment per tant em centraré ara a explicar el procés d'inici d'entrenament (sessió). En el següent procés he indicat dues línies (Blava i Groga) per a que es vegi la diferenciació visual que es percep segons si es fa l'entrenament en temps real o en mode offline. Aquest canvi brusc i notable de color és per a que sigui fàcil a simple vista per a l'entrenador saber si està realment a un o altre mode a partir de la identificació dels colors. La vista de mode offline, a més, té un quadres informatiu del seu funcionament.



També hi ha la opció, com he comentat, de començar un entrenament lliure fent clic al botó *"Start Free Training"* que ens duria a la selecció del client al que volem entrenar mitjançant la vista de *Llista clients* i un cop escollit ens deixaria triar també el tipus d'entrenament entre entrenament a partir d'una rutina ja creada o entrenament lliure a partir dels exercicis que tenim donats d'alta:





#### 7.2 - Menú Lateral

El menú lateral ja ha sigut bastant comentat. És el menú que té el coach disponible per a administrar tots els recursos que després pot fer servir amb l'usuari. Des d'ell pot accedir a les vistes de *Llistat entrenaments, Llistat exercicis i Llistat màquines* juntament amb els accessos directes al tancament de sessió actual i la vista de *Configuració de perfil*.

El menú lateral es pot desplegar fent clic al botó superior esquerra o bé fent swipe d'esquerra a dreta. Per a amagar-lo s'ha de dur a terme el procés contrari.

A part del menú en si, estèticament a la capçalera del menú hi ha una foto del perfil que hagi iniciat sessió en aquest moment per a una identificació ràpida de l'usuari associat (Especialment útil per als gimnasos on es faci servir la APP en dispositius de l'empresa – no personals).

En el futur es preveu també que pugui implementar l'entrada d'entrenaments a través de smart devices (Apple Watch per a iOS i rellotges Android per a la APP (no creada encara) per a Android) i per això s'ha inclòs ja – encara que deshabilitat – el menú de *"Smart Devices"*.



#### 7.3 - Container Sync Api i Sync Offline

El container utilitzat per a notificar si les accions han anat com s'esperava o no l'hem vist ja en diverses vistes anteriors. Funcionen d'una manera similar als fragments d'Android en el sentit que inicialment declares un contenidor amb els seus atributs visuals (mides i posició a la pantalla únicament) i més endavant, quan necessites mostrar l'avís és quan crides al contenidor i hi col·loques la vista dins (amb tot el disseny, colors, labels, etc.). El disseny i implementació dels dos UIContainers utilitzats a l'aplicació és exactament idèntic. La única diferència és que els criden diferents classes.



#### 7.4 - Contenidor Menú inferior

Aquest menú també ha aparegut ja en nombroses vistes comentades amb anterioritat. Concretament a totes les vistes principals (*Calendari, Missatges, Clients i Main (Vista principal)*. Com s'ha comentat en aquestes vistes, és el menú que permet navegar entre elles.

La implementació és molt similar a la utilitzada per als avisos (*Container Sync Api* i *Container Sync Offline*). En aquest cas, però, la vista associada al contenidor no està formada per una UI creada a ma amb UILabel(s), UIButton(s) i UIView(s) sinó que només conté una UITabBar, és a dir, una barra de pestanyes. S'ha implementat addicionalment que l'element seleccionat es vegi del color blau de l'aplicació.



### 8- Entrenaments

He decidit deixar aquesta secció pel final ja que crec que era necessari entendre primer la resta d'estructuració de l'aplicació tant pel que fa al funcionament intern com per les vistes i la seva organització. Arribats a aquest punt ja sabem com l'entrenador organitza les seves rutines d'entrenament, exercicis i màquines a través del menú lateral i com administra les sessions programades amb els clients amb el calendari. També hem vist com pot mantenir converses amb ells i com pot afegir nous clients tot des de les vistes corresponents.

Hem vist que des de la vista *Main* l'entrenador pot decidir entrenar alguna de les sessions programades que apareixen a la llista o bé si ho desitja començar un entrenament nou fent click a *"Start Free Training"* i acte seguit seleccionant el client de la vista *Llistat clients.* En aquesta secció veurem com es dur a terme la captura de dades tant si s'utilitza el mètode en temps real (*Real Time Training* – El que tenia el fons blau ) com si s'escull el mode offline (*Hands Free Training* – El que tenia el fons groc).

El funcionament d'un entrenament és relativament senzill, però la seva representació únicament amb captures de pantalla resulta difícil de plasmar de manera que abans de passar a veure les vistes mostraré uns petits diagrames amb el funcionament.

Conceptes bàsics:

- Entrenament Offline: L'entrenador realitza l'entrenament sense l'aplicació i <u>posteriorment</u> registra tots els resultats a Gorilla Coach.
- Entrenament Real Time: L'entrenador realitza l'entrenament capturant els temps i la resta de dades en <u>aquell mateix moment</u> amb el cronòmetre i eines que ofereix Gorilla Coach.

A part de poder capturar els temps amb aquests dos modes es poden subdividir els entrenaments en dos grups més cadascun:

- Entrenaments a partir de rutines ja creades: Es tria una rutina, aquesta carrega la llista d'exercicis que conté i permet anar-los seleccionant per a realitzar l'entrenament. Detalls:
  - Un cop s'ha realitzat un exercici queda marcat de color blau.
  - L'exercici seleccionat (següent a realitzar) està marcat de color groc.
  - No hi ha un ordre específic. L'entrenador decideix quin ordre seguir.
  - L'entrenador pot finalitzar l'entrenament en qualsevol moment, no és obligatori acabar la llista d'exercicis.
- Entrenaments lliures: L'entrenador decideix sobre la marxa quin exercici de la llista total d'exercicis vol realitzar a continuació. Detalls:
  - Un cop s'ha realitzat un exercici aquest mostra el botó "remove" per si es vol rectificar.
  - És l'entrenador qui decideix quan s'acaba l'entrenament.

L'administració dels exercicis a realitzar, així com el fet d'escollir l'ordre o exercicis es dur a terme de la mateixa manera en els entrenaments Offline i els entrenaments en Real Time. L'única vista que canvia és estrictament la de captura de dades (i la pantalla inicial de cada tipus d'entrenament que canvia de color). Diagrama de recol·lecció de dades a partir d'una rutina d'entrenament existent:



Diagrama de recol·lecció de dades a partir d'exercicis de la llista (Free Training):



#### 8.1 - Entrenaments en temps real

Com acabem de veure el que caracteritza l'entrenament en temps real és que les dades es capturen al moment. Això vol dir que l'entrenador haurà d'anar marcant cada sèrie o repetició que fa el seu client. També es podrien registrar només els temps totals de cada sèrie però els resultats guardats llavors serien molt més pobres.



Exemple d'entrenament en temps real a partir d'una rutina d'entrenament existent. L'exercici amb el fons de color blau ja està realitzat. L'exercici marcat amb fons groc és el seleccionat per a dur a terme a continuació.



Exemple d'entrenament en temps real lliure. Els exercicis amb el botó "ADD" no s'han realitzat. L'exercici amb el botó groc "REMOVE" ja ha sigut realitzat. Es podria eliminar de l'entrenament.

#### 8.2 – Entrenaments "Hands Free"

L'entrenament Hands Free o Offline és el més adecuat per a aquells entrenadors o sessions d'entrenament en la que has d'estar al 100% per al client i només pots introduir les dades un cop acabat l'entrenament. A continuació mostro com es capturen les dades:



Exemple d'entrenament en temps real a partir d'una rutina d'entrenament existent. L'exercici amb el fons de color blau ja està realitzat. L'exercici marcat amb fons groc és el seleccionat per a dur a terme a continuació.



Exemple d'entrenament en temps real lliure. Els exercicis amb el botó "ADD" no s'han realitzat. L'exercici amb el botó groc "REMOVE" ja ha sigut realitzat. Es podria eliminar de l'entrenament i també es podria posteriorment.

## 5. Implementació

En aquesta secció de la memòria detallaré alguns aspectes generals de la implementació tant de la API com d'iOS. També faré menció d'aquells que han sigut especialment complicats o que m'han portat més temps d'investigació o desenvolupament. Entre ells cobriré alguns temes que són diferents d'Android per a mostrar com funciona en el sistema operatiu mòbil d'Apple.

### API

### **Aspectes Generals**

Com he mencionat anteriorment el micro-framework *Flight* ja incorpora algunes funcions bàsiques de la creació d'APIs. Ara bé, que aquestes serveixin o no ja ve determinat per les necessitats de cada projecte i en el meu cas vaig haver de fer alguns ajustaments i modificacions per tal que al final tot funcionés com jo necessitava.

#### Recuperar dades enviades per POST

Gran part de les crides a funcions que guarden dades a la base de dades les he programat per a que enviïn via POST. D'aquesta manera l'estructura de URLs es manté neta i a l'hora les dades estan ben organitzades. Per a que això funcioni correctament vaig programar inicialment algunes funciones generals. En aquest apartat veurem la funció que es dedica a capturar les dades enviades per POST i com posteriorment es llegeixen, passen el control necessari i s'envien a *functions.php* per a que aquest realitzi les tasques pertinents en cas que les dades siguin correctes.



53 \$params = API\_GetREQUESTParams();

Crida a la funció API\_GetRESQUESTParams

Per a qualsevol petició entrant a la API es mira si hi ha paràmetres per POST ja que hi ha funcions que utilitzen tant variables per URL (La sessió generalment) com paràmetres per POST (La llista de dades). Un cop recol·lectades les dades veiem com es llegeixen i tracten a continuació:



Recol·lecció de les dades

I dins d'un exemple complet com pot ser el *registre* quedaria així:



Com es pot veure, entre les línies 153 i 167 es recol·lecten les dades per a poder dur a terme la funció sol·licitada. Entre les línies 171 i 177 s'analitza quin tipus de registre serà i es crida a la funció de *functions.php* corresponent. I per últim es mostra el resultat comprovant entre les línies 180 i 183 que no hi hagi cap error.

Per veure un exemple pràctic, si fem la següent crida contra la API,

POST 🗸	https://www.gorillacoa	:h.com/api/v1/signup/?XI	DEBUG_SESSION_START=	15874	Params	Send	~	Save	
	Headers (1) Body •								
🔵 form-data	x-www-form-urlencodec	🖲 raw 🛛 🔵 binary	JSON (application/json)	) 🗸					
1 2 - { 3 "us 4 "pp 5 "mm 6 "pp 7 "us 8 "nc 9 "ss 10 "dr 11 "lc 12 } 13	sername":"Gorilla-Coac sssword":"12345", iil":"Gorilla-Coach-Te erms":1, serType":"coach", sme":"Coach1", urname":"Cognom1 Cogno ni": "00000001X", anguage":"esca"	n-Test", st@GorillaCoach.com m2",							

el resultat que aquesta ens tornarà serà el següent:



Aquest resultat inclou el moment de l'execució, un codi de resultat on **0** = **OK** i qualsevol altre valor diferent a 0 = **KO**. Adicionalment trobem la figura coach completa per a que es pugui guardar a la BD local desde la APP i el codi de la sessió, que es torna ja iniciada per no requerir que l'usuari després de registrarse hagi de fer un login.

Un altra aspecte en el que ens podem fixar respecte la implementació de la API és en la etiqueta **value**. Els retorns de la API tal i com ho he programt poden retornar els valors o bé dins l'etiqueta **value** o bé dins l'etiqueta **list** depenent de les dades que s'hagin de retornar. Donem-hi una ullada més en profunditat. La funció que s'encarrega de retornar tots els valors necessaris en el format correcte és la següent:



És una funció amb una estructura molt senzilla però que m'ha salvat moltes línies de codi i hores de programació. Com es pot veure sempre té en compte el **temps** i **codi de resultat** i, addicionalment, si hi ha un *error*, *valor o llista de valors* ho té en compte incloent dins de la variable **result** els camps JSON *list*, *value* o *error*. Finalment, el resultat es retorna.

Hem vist com queda l'estructura d'un value, però, com seria en cas de ser una llista? Anem a veure un exemple amb la funció GET que retorna els entrenaments. Les rutines d'entrenament per si soles no valen res, necessiten la llista d'exercicis que té el coach i les vinculacions entre rutines d'entrenament i exercicis. A més a més, els exercicis depenen de les màquines a les que es duen a terme, per tant la funció encarregada de retornar els entrenaments retornarà totes aquestes dades per poder operar amb elles un cop arribin a la APP.





### **Business Logic**

L'arxiu *functions.php* és l'únic amb accés a la base de dades i l'encarregat de dur a terme les peticions que entrin a través de l'*index.php*. Està estructurat en diferents blocs segons si són operacions de control de sessions, altes de nous registres o getters de dades ja guardades.

La lògica implementada <u>no es limita</u> al que seria equivalent en llenguatge SQL a INSERTS, UPDATES, SELECTS i DELETES sinó que controla moltes més coses com per exemple:

- Controlar que es tenen permisos per accedir a les dades sol·licitades.
- Enviar correu electrònic i **configurar claus d'accés per token temporals.**
- Comprovar estat d'actualització de dades entre una app-client i el servidor.
- En cas que alguna cosa falli, recuperar l'error en l'idioma corresponent de la base de dades i retornar-lo.
- Estructurar les dades recuperades de la base de dades per a poder-les retornar fàcilment un cop arribin a l'*index.php*.
- **Recórrer les dades rebudes** i fer amb elles les operacions pertinents (generalment INSERTS)
- Controlar les sessions dels usuaris

Per a una implementació òptima també he creat algunes funcions d'ús global. D'aquesta manera no cal tenir una funció interna per a recuperar un camp concret de preferències de l'usuari, o la ID del coach d'un entrenament concret, o l'usuari al que pertany una sessió rebuda. Algunes d'aquestes funcions són:

function countRowsById(\$tableName, \$fieldName, \$key, \$value, \$keyType) { ... }
function updateFieldByID(\$tableName, \$fieldName, \$fieldValue, \$key, \$value, \$keyTypes) ) { ... }
function deleteRowsByID (\$tableName, \$key, \$value, \$keyType) ) { ... }
function getOneFieldByID (\$tableName, \$fieldName, \$key, \$value, \$keyType) ) { ... }

Les funciones per a obrir i tancar les connexions a la base de dades, així com la configuració (nom del a BD, usuari, password, etc) es troben també en aquest mateix arxiu.

#### **Control d'errors**

El control d'errors l'he dut a terme amb tot un seguit d'identificadors i una taula extra a la base de dades. Els identificadors com és d'esperar fan referència a cadascun dels possibles errors que es poden donar dins de l'aplicació. A la taula "errors" de la base de dades el que hi trobem és el missatge que s'ha de mostrar en els diferents idiomes en que està disponible la aplicació. Ara mateix són 3 els idiomes disponibles (Català [es-ca], Castellà [es-es] i Anglès [en-en] ) però és ampliable simplement afegint una columna amb el codi de llenguatge internacional que es desitgi afegir i introduint les traduccions corresponents a cadascun dels missatges d'error.



El funcionament és molt simple, quan *functions.php* retorna el valor a *index.php* perquè el retorni a l'usuari si no és el format que espera i conté un codi d'error (Exemple  $\Rightarrow$  "error": 2) l'*index.php* crida a la funció *getError()* tot passant el codi d'error i la sessió de l'usuari per a poder recuperar l'error en l'idioma que l'usuari tingui guardat a les seves preferències. Per defecte es retorna en anglès si no s'ha pogut guardar l'idioma de l'usuari durant la fase de registre.

id_error	code	esca	eses	enen
1	1	Contrasenya incorrecta	Contraseña incorrecta	Wrong Password
2	2	Usuari incorrecte	Usuario incorrecto	User not found
3	3	Aquest nom d'usuari ja està en ús	Este nombre de usuario ya existe	Username already in use
4	4	Sius plau, empleneu tots els camps	Por favor, rellene todos los campos	Please fill all the fields
5	5	Hi ha hagut un error durant el registre	Ha ocurrido un error durante el registro	An error ocurred while setting up your acocount
6	6	El codi de verificació no és vàlid	El código de verificación no es válido	This verification ccode is not valid.
7	6	El codi de verificació no és vàlid	El código de verificación no es válido	This verification code is not valid.

Exemples de registres de la taula "errors"

### iOS

D'iOs he après tantes coses que realment no sé molt be per on començar ni què cobrir exactament en aquesta memòria. Per poder fer aquest treball vaig haver de fer un curs exprés de desenvolupament d'aplicacions iOS amb Swift a principis d'any. He de dir que hi ha moltíssimes coses que com és evident dins del món de la informàtica, són molt semblants a Android. Però a l'hora totes aquestes coses tenien detalls que eren diferents a la seva manera.

Per posar un exemple, clar que les vistes tenen el seu propi cicle de vida igual que a Android, però els diferents estats possibles així com la gestió interna de tot plegat és el que canvia. O el pas de variables entre classes i vistes, a Android ràpidament pensaríem amb bundles i en canvi a iOS n'hi ha prou amb declarar la variable fora de l'scope de la classe i automàticament la pots fer servir a la classe que la necessitis. A la resta de classes que no es fa servir la variable no hi serà present ni ocuparà memòria. O els típics Callbacks d'Android, amb iOS també es duen a terme de manera diferent.

Aquest seguit de coses seran les que intentaré cobrir en aquesta secció per tal de fer una aproximació al món d'Apple una mica més pràctica i deixant de banda els aspectes que són molt semblants a Android. També comentaré alguns temes de la lògica que conté la aplicació i sobretot de l'estructura de dades seguida.

### Cicles de vida a iOS

A continuació he creat un diagrama de les diferencies entre els cicles de vida de Android i iOS juntament amb les seves equivalències. Com es pot veure a continuació la majoria d'etapes del cicle d'Android es poden definir també a iOS i no és tant en l'estructura sinó en la manera interna que té el sistema de gestionar-ho on es noten les diferències de veritat.

A Android era molt comú trobar-te amb aplicacions mal programades que no tenien en compte el cicle de vida de les activities i això comportava que la bateria caigués en picat fins i tot quan l'aplicació en sí no era visible. Android sempre ha donat més llibertat a l'usuari i un control molt més exhaustiu dels recursos de què disposa incloent-hi l'accés i gestió del sistema operatiu i aquesta és una de les grans diferències que m'he trobat en quant als cicles de vida. A iOS, a menys que ho especifiquis explícitament l'aplicació no pot fer res un cop prems el botó *"Home"* del dispositiu. I fins i tot programant-ho específicament, les activitats en segon pla estan molt restringides de manera que és impossible que un procés es quedi buguejat en background consumint bateria.



#### **iOS** Protocols

La gestió asíncrona de dades entre la BD i la vista visible en el meu cas era un tema important. Quan l'aplicació inicia, es dirigeix a l'usuari a la vista *Main* però és molt possible que les dades encara no hagin arribat del servidor. Recordem que quan l'aplicació inicia es fa una request a la API per rebre totes les novetats i aquest procés pot trigar 1-2 segons a ser completat ja que implica:

- 1. Crida a API per a que ens digui quines seccions cal actualitzar
- 2. Crida per a cada secció que cal actualitzar
- 3. Guardar les dades a la BD Local de la APP

I aquí faltaria l'últim pas, que és:

4. Mostrar les dades a pantalla

Aquest últim pas a Android se soldria fer amb un Callback que actualitzaria les dades de la taula i la recarregaria. A iOS la segona part és exactament igual però els callbacks es defineixen d'una manera molt més general (i elegant/neta) i auto-controlen gran part dels problemes de vistes que solen ocórrer a Android. Per exemple, per a la actualització de dades entre API-APP jo he declarat aquest protocol:

import Foundation
protocol HomeProtocol{
func syncFinish()
}

Aquestes 3 línies són les úniques necessàries per a implementar un Protocol a iOS amb Swift. Amb Objective-C el tema es complica bastant però precisament per això Apple va decidir llançar Swift i per la mateixa raó jo vaig escollir Swift com a llenguatge pel projecte.

Un cop declarat el protocol, que com es pot veure no està vinculat a res i és 100% independent de la resta de classes només s'hauran de dur a terme 2 canvis més a la classe on es vulgui aplicar per tal que sigui operatiu. En primer lloc a la declaració de la classe s'haurà d'indicar que exten de *HomeProtocol* i que per tant implementarà el mètode *syncFinish()*:

```
25
26 class StartViewController: UIViewController,UITableViewDataSource, UITableViewDelegate,
DZNEmptyDataSetSource, DZNEmptyDataSetDelegate, HomeProtocol {
27
```

I en segon lloc, com és d'esperar s'haurà d'implementar la funció syncFinish() de la classe HomeProtocol:

```
314
315
func syncFinish(){
316
print("----Sync Finished")
317
getData()
318
319
}
```

En aquest cas el codi és molt curt ja que fa us de la funció *getData()* declarada a la mateixa classe *StartViewController* és qui té la funció de recuperar les dades de la base de dades locals, carregar-les en

variables per a poder ser tractades en temps real i refrescar la taula on es mostren les sessions d'entrenament programades. I per a cridar-la l'únic que s'ha de fer és cridar a la funció *syncFinish()* com es mostra a continuació:



### Crides a la API

Una part molt important de l'aplicació és la comunicació que aquesta té amb la API. Inicialment totes les crides les feia manualment però hi va haver problemes amb els temes de codificació de certs caràcters i altres aspectes mentre implementava les crides a les funcions POST de la API. Mirant alternatives vaig descobrir una llibreria que facilita bastant les comunicacions via HTTP i HTTPS i vaig decidir migrar-ho. La llibreria en qüestió és Alamofire, i tot i que no estalvies gaires línies de codi, et permet configurar qualsevol paràmetre com si s'estigués programant de 0 amb la diferència que et reporta els errors d'una manera molt més comprensible, i aquest aspecte, possiblement va ser el de més ajuda de tots els que inclou.

Totes les crides a la API es fan des del controlador (*Controller.swift*). A més totes elles segueixen una estructura molt semblant:

### BASE URL + ROUTING API (+/INLINE\_VARS)

És per això, que per tenir tot el codi net i controlat, he creat una funció encarregada de fer les connexions (Tan si són de tipus GET com si són de tipus POST). La funció és la següent:



En aquesta funció es pot apreciar la paraula *callback* a les línies 122 i 148. No ens hem de confondre, doncs no és un *callback* tal com s'entén o se sol utilitzar a Android. Quan es crida a la funció *request* per a fer una petició a la API aquesta s'executa dins de la llibreria que hem mencionat - *Alamofire* - que és multifil per a no interrompre el fil d'execució de la interfície. El que fa la crida *callback* que es veu a les línies comentades és tornar el control a la classe que s'estava executant per tal que es puguin guardar les dades a la base de dades prosseguint l'execució <u>únicament dins del codi que quedava a l'espera</u>. Seguint l'exemple anterior de la funció *updateClients():* 

2	<pre>self.request("GET", request:"clients/\(session)",params:["":""], json: jso     if let responseString = response as? JSON {</pre>	n) { (response) in
2	<pre>//print("response: \(responseString)")</pre>	
2	//Getting a string from a JSON Dictionary	
2	<pre>let result = responseString["result"].intValue</pre>	
2	<pre>//print("resultat es: \(result)")</pre>	
2		
2	switch result {	

Si es para atenció al final de la línia 291 es pot veure com se l'hi indica que s'espera una resposta per a executar el codi entre els brackets. Posteriorment es comprova que la resposta sigui un JSON que és el format esperat de la API i ja es passa a analitzar si el resultat és un 0 (Tot correcte) o un valor diferent (Algun error).

## 6. Continuïtat del projecte

Gorilla Coach està pensat per ser una solució integral per als entrenadors personals i per a arribar a aquest punt encara queda cobrir més d'un front. Amb aquest treball final de grau he pogut cobrir aproximadament un 80% de les funcionalitats necessàries per a iOS. Faltarien algunes coses com:

- Gestió de gimnasos
- Clients compartits
- App per a Apple Watch
- Notificacions Push
- Altres millores
  - o Imatges de rutines d'entrenament dinàmiques segons la part del cos treballada
  - o Possibilitat d'afegir sessions de control (Pes, cintura, músculs, etc)
  - o Explotació de les dades recol·lectades per a unes estadístiques úniques en el sector
  - o Millorar sistema de sincronització
  - Permetre posar en pausa entrenaments i continuar-los més tard o des d'un altre dispositiu

A més a més s'ha de comptar que no tots els entrenadors fan servir iOS i s'hauria de cobrir també la demanada en dispositius Android. Juntament amb Android s'haurien de tenir en compte els dispositius Apple Watch.

I per últim faltaria també una web per als directius de gimnasos que no fan servir la APP per entrenar i tota aquella gent que prefereix gestionar-ho per web, juntament amb el fet que la web és una plataforma que sol usar-se desde dispositius amb pantalles més grans i on és més fàcil informar als usuaris, gimnasos i entrenadors personals de totes les facilitats que ofereix el servei sense que aquests hagin d'instal·lar res en el seu dispositiu.

Tan per la web com per a Android hi ha la gran avantatge que la API ja està implementada, i que tret dels pocs canvis que requereixi implementar les millores restants d'iOS no s'hauria de tocar per a res, cosa que facilitaria molt la feina a realitzar.

Aquesta primera etapa ha sigut molt agradable i sobretot molt satisfactòria de veure la qualitat que des del meu punt de vista he pogut plasmar sobre el paper i sobre la pantalla i serà molt valuós tenir aquesta bona sensació per a continuar el desenvolupament.

## 7. Conclusions

Ha sigut una gran experiència submergir-me de nou en el món d'iOS després de tants anys sense tocar-lo més que a nivell d'usuari. El sistema operatiu d'Apple, que com a usuari ja sabia que havia millorat molt, també ho ha fet en l'entorn de desenvolupament que ofereix. Possiblement si no haguessin presentat Swift al 2014 aquest treball no hagués sigut possible tenint en compte tota la càrrega de treball extra que suposava aprendre Objective-C.

He patit bastant al final al anar a contrarellotge degut a alguns errors comesos en la fase de disseny i també degut a aquest Handicap que suposa aprendre com funciona internament una plataforma nova i la formació necessària per a poder treballar correctament sobre ella.

Però mirant enrere i comparant els anys que porto de desenvolupador Android i el present treball amb iOS he de dir que tot i que si que és més complicat aprendre a fer les coses tal i com està pensat per Apple que es facin ( hi ha menys documentació) al final et trobes també amb tot un seguit d'avantatges. Algunes les he anat comentant al llarg de la memòria com a pinzellada però realment s'agraeix molt el fet de poder tenir la total seguretat que el codi que piques com a universal, és realment UNIVERSAL. La majoria de test els he fet en un iPhone 6S Plus(5,5") (Que és el que utilitzen al gimnàs), però he tingut la oportunitat de probar-ho també en un iPhone 6S (4,7") i en un iPod Touch amb la pantalla encara més petita i al haver estat treballant amb el que ells anomenen Disseny Inferred per a aplicacions Universals totes les vistes a excepció d'una estaven adaptades sense requerir cap canvi. Per a la vista mencionada va ser necessari únicament revisar uns *constraints* de la interfície visual que no estaven del tot ajustats.

Un cop li agafes la pràctica, no crec que es pugui dir que el desenvolupament en iOS sigui més costós en quant a temps del que és a Android, i més si tenim en compte l'aspecte ja comentat del testing i els bugs segons dispositiu, problema arxiconegut d'Android.

Pel que fa referència al llenguatge, Swift com qualsevol altre té les seves manies, bones pràctiques i dificultats, però des d'Apple han fet un gran treball. Tenint en compte que és un llenguatge amb només 2 anys de vida, 6 dels quals ha estat ja sent codi lliure (Apple ho va fer públic fa aproximadament 6 mesos), he de dir tant el desenvolupament com el debugging es mil vegades millor al que hi havia amb Objective-C, llenguatge que porta al mercat moltíssim més temps i al que Apple dóna suport des d'abans que sortís el primer iPhone al 2008.

I per últim, m'agradaria acabar aquest treball tal com l'he començat remarcant els agraïments que he fet constar a la primera secció de la memòria, no hagués sigut possible arribar fins aquí sense ells. Un cop més, moltes gràcies a tots!