

Realització d'una PWA i una app híbrida amb un back-end sobre PHP

Joel Felip Cárdenas

Resum– El projecte pretén extreure conclusions sobre el debat entre tecnologies de desenvolupament web i mòbil actuals (*Native vs Hybrid vs Progressive*) basant-se en l'experiència assolida realitzant dos prototips d'una mateixa aplicació centrada en el suport a la formació esportiva. El següent document recull el procés de desenvolupament de les dues versions: una PWA en React i una Hybrid App en React Native fonamentades per un back-end en Lumen. Finalment, es realitza un estudi dels dos procediments valorant els pros i contres de cada alternativa i un petit sondeig a un conjunt d'usuaris per tal de contextualitzar el nivell de familiarització de la societat amb aquests tipus d'aplicacions.

Paraules clau– Aplicació Web Progressiva, Aplicació Web Híbrida, React, React Native, Laravel-Lumen, PHP, Javascript

Abstract– This project aims to draw conclusions about the debate between current web and mobile development technologies (*Native vs Hybrid vs Progressive*) based on the experience realizing two prototypes of an application focused on supporting racket sports training. The following document covers the development process of the two versions: a PWA in React and a Hybrid App in React Native based on a back-end in Lumen. And lastly, a study of the two procedures and a small survey are carried out assessing the pros and cons of each alternative in order to contextualize the level of familiarity of society with these kind of applications.

Keywords– Progressive Web App, Hybrid App, React, React Native, Laravel-Lumen, PHP, Javascript

1 INTRODUCCIÓ

LA diversitat d'opcions existents a l'hora d'orientar el desenvolupament d'una aplicació mòbil ha obert un debat amb multitud d'opinions al respecte sobre quina solució escau millor a cada projecte. L'objectiu del següent treball serà obtenir conclusions pròpies basades en l'experimentació durant un desenvolupament individual d'una PWA i la seva versió híbrida enfocada al món de la formació en l'esport.

Aquesta aplicació ha de resultar una eina d'ajuda per a tècnics esportius en el món del tennis dins i fora de les pistes. El principal objectiu recau en el foment d'entrenaments complets i amb diversitat d'exercicis. La monotonia en les sessions d'entrenament és un dels casos que més és produït en les escoles i clubs formatius en esports de raqueta. És per això, que el software inclourà un repositori d'exercicis indexat i accessible mitjançant filtres segons la tipologia per a que els tècnics puguin planificar i consultar-ne el contingut en qualsevol moment. Els exercicis emmagatzemats

continuaran el nom, la descripció del propi i un seguit d'informació addicional com el material necessari, un esquema de la distribució dels alumnes en pista, etc... D'altres eines que incorpora el software és el generador d'informes per obtenir en format PDF la planificació realitzada, un mòdul de dibuix per complementar les explicacions tècniques o un mòdul on realitzar proves de nivell.

Un cop realitzat el desenvolupament d'ambdues aplicacions es farà un estudi a nivell d'usuari sobre la velocitat de desplegament i temps de resposta i l'ús de la memòria caché del cercador. Per últim, es realitzarà un sondeig entre potencials usuaris de l'aplicació, donant a testejar les dues versions, per recollir l'opinió pública i la percepció de les dues tecnologies. La comparativa final valorarà aspectes clau de cadascuna de les tecnologies com la capacitat de funcionar offline, el nivell de familiarització dels usuaris amb el tipus d'aplicació o la usabilitat, per tal de comprovar com afecta l'adaptació d'una web al format natiu.

El projecte es desenvoluparà en ReactJs i React Native, dues de les tecnologies i competències més demandades del 2019, per tal de complementar la formació universitària en PHP i Javascript.

- E-mail de contacte: joel.felipc@e-campus.uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Joan Giner Miguelez
- Curs 2019/20

1.1 Estat de l'art

Per a que el conjunt del treball prengui sentit és vital que el producte o aplicatiu web final sigui funcional i que realment atengui necessitats reals al target definit d'usuari. Així doncs, per ajudar a definir el producte i prioritzar els requeriments s'ha comptat amb l'opinió de tècnics esportius en esports de raqueta i activitats en l'aire lliure en actiu.

Per altre banda, s'han cercat a internet solucions software que resolguin parcial o totalment les necessitats establertes a mode d'orientació. A continuació es descriurà l'alternativa que comparteix més similituds amb el nostre objectiu.

Powerplay Tennis Software [1] és un software de pagament orientat a la gestió, control i planificació d'entrenament esportiu. És molt extens i la interfície no és gaire atractiva. Actua realment com un sistema d'informació transversal on es mostren grans quantitats de dades i es pot monitoritzar l'evolució al llarg del temps dels usuaris. Compta amb una gran varietat d'opcions a l'hora de generar reports cosa que en dificulta lleugerament l'ús. En definitiva, és una opció vàlida però que per resultar més pràctica necessitaria una reestructuració de les funcionalitats acompanyada d'un redissenyat visual.

A continuació es detallaran els aspectes que es consideren ineficients en aquest software:

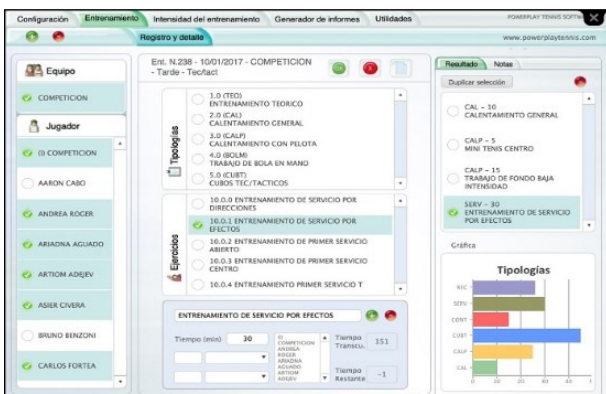


Fig. 1: Captura de "Registre i detalls" dels entrenaments a PowerPlay TS

El primer aspecte conflictiu en aquesta interfície és la gran quantitat de dades que es mostren en una sola pantalla i la diversitat de les pròpies dades. Per exemple, en la part esquerra es mostra el nom o tipologia del grup d'alumnes que s'està analitzant i els seus integrants, en el bloc central apareixen dades sobre el banc d'exercicis i a la dreta de la pantalla es veu un resum de l'entrenament complet junt amb una gràfica de barres sobre les tipologies dels exercicis escollits.

Si pensem en una solució ideada a mode de sistema d'informació / plataforma de gestió a gran escala totes aquestes dades poden resultar útils però si el resultat que volem obtenir ha d'estar orientat a dispositius mòbils i petit format s'ha de filtrar el que no és necessari. Eliminar la gràfica de tipologies degut a que cada grup d'entrenament necessita especial atenció en determinats tipus d'exercicis sense ser un tret perjudicial el fet de que es produeixin desproporcions puntuals en els gràfics. Mantenir un historial dels entrenaments realitzats ja és suficient per ajudar en la planificació

de les sessions futures. El format del filtre per tipologies situat al centre de la pantalla és poc pràctic i no seguiria un estàndard actual. Per últim, no caldria mostrar els noms i cognoms dels alumnes que han realitzat l'entrenament almenys en aquesta pestanya. Si es vol fer un control d'assistència o una avaluació d'un entrenament o competència en particular seria més intuïtiu si es fes en una altre pantalla.

La informació extreta de les xerrades amb els diferents tècnics focalitza l'atenció en dos principals aspectes d'usabilitat: l'eina perfecte ha de ser accessible des de dispositius mòbils o tablets, idealment sense necessitat de connexió a Internet, i ha de ser intuïtiva i fàcil d'utilitzar. Així doncs, tractarem d'evitar mostrar dades innecessàries o complexes, acumulació d'informació en pantalla o funcionalitats poc utilitzades i optarem per orientar l'eina al suport del entrenador tant dins com fora de les pistes. Els tècnics valoren positivament el fet de tenir accés a un repositori d'exercicis indexats i que es puguin filtrar per tipologia per tal de consultar de forma ràpida exercicis vinculats a un aspecte tècnic de l'esport i evitar la monotonia als entrenaments.

Un altre aspecte que, de forma més puntual o amb alumnes d'un nivell avançat, pot resultar pràctic és el de la pissarra tàctil. Troben que mostrar certs aspectes tècnics i tàctics com el direccionament dels cops, el posicionament en pista o la trajectòria dels desplaçaments més òptims des de la mateixa eina pot facilitar la interpretació de les explicacions. Les zones d'impacte en la raqueta i els diferents efectes proporcionats a la pilota és poden mostrar de forma visual i complementar els exemples en viu. La pissarra tàctica és una eina que s'utilitza en multitud d'esports com el bàsquet, l'handbol o el futbol i serveix als entrenadors per definir jugades i corregir errades durant els partits i entrenaments. L'eina ideal hauria de poder fer *switch* entre diferents imatges de fons (ex. pista de tennis - vista zenital, raqueta o pala, pilota...) sobre les que poder traçar línies o posicionar-hi objectes.

1.2 Objectius

Els objectius principals del projecte són:

- Recopilació d'informació i experimentació sobre tecnologies actuals referents en el món del desenvolupament web i mòbil com són React i React Native amb l'objectiu d'argumentar el debat existent entre quina opció encaixa millor en segons quin tipus de projectes.
- Completar un desenvolupament d'una web-app útil amb un back-end relativament bàsic sobre PHP. Realització tant de la versió PWA, com de la híbrida.
- Modelat complet de la base de dades del projecte i de la api de gestió i interacció en Laravel-Lumen.
- Disseny de les diferents interfícies del projecte. En un principi, s'usaran components de Material-UI o similar. (La prioritat principal recaurà en el funcionament correcte de la solució i es centrarà en l'anàlisi de les diferents versions obtingudes. Es poden considerar dur a terme millores de caire estètic en una etapa avançada del projecte).
- Contextualitzar i argumentar el creixement i aparició d'aplicacions híbrides vers les aplicacions natives tradicionals.

- Destacar pros i contres de cada solució i formular una opinió sobre aquestes tecnologies basada en l'experiència personal. Quin desenvolupament ha comportat una major dificultat o dedicació, quins són els requisits a complir si es vol convertir una web a una PWA descarregable, quins canvis pateix la API per al desenvolupament natiu per a dispositius mòbils, fins quin punt resulta reutilitzable el codi de la versió web per a la híbrida, comprovar quines llibreries de javascript mantenen suport de la comunitat i són compatibles amb el codi de React Native i aconseguir definir un target que es fonamenti en el tipus d'equip de desenvolupament i la finalitat del propi projecte per cadascuna de les tecnologies esmentades.

1.3 Descripció del producte

L'eina a desenvolupar durant aquest projecte serà una web-app orientada a la ajuda i suport en pista per entrenadors, monitors o preparadors físics en el món del tennis i esports de raqueta. L'eina pretén ser un instrument eficaç per a tècnics de qualsevol nivell des de iniciació a competició des d'on podran consultar recursos, planificar sessions d'entrenament i complementar les seves explicacions tècniques i tàctiques. El producte haurà de ser accessible des de dispositius mòbils de mitjà-gran format, tablets i ordinadors.

Els principals mòduls i funcionalitats amb els que comparà l'eina són:

- Visió de continguts i materials. Mòdul en el que l'usuari podrà accedir al repositori d'exercicis i veure tota la informació vinculada a un exercici (nom, descripció, material necessari, objectiu, duració, imatges o vídeos explicatius, recomanacions, aspectes a tenir en compte, etc...). Els exercicis es podran accedir introduint-ne el nom, cercant-los en un llistat o fent ús de filtres. Un usuari administrador podrà afegir un nou exercici al banc de continguts.

- Generador d'informes. Mòdul amb el que un tècnic podrà dissenyar sessions d'entrenament seleccionant exercicis del repositori. Aquests exercicis es podran filtrar segons diferents criteris (tipus d'exercici, etapa del entrenament on s'efectua, cop treballat...) per agilitzar-ne la cerca. Un cop finalitzada la planificació es podrà exportar l'informe en format PDF per tal de facilitar-ne la compartició amb altres tècnics. Els informes generats quedaran emmagatzemats en un històric accessible.

- Mode pissarra. Mòdul de dibuix interactiu amb el que un tècnic podrà fonamentar les seves explicacions tàctiques a la mateixa pista.

- Registre de jugadors i control d'assistència a l'entrenament. Un tècnic podrà definir un grup o equip d'alumnes i validar la seva assistència als entrenaments.

- Proves de nivell. El tècnic pot avaluar les capacitats d'un alumne realitzant una prova de nivell i anotant els resultats en el sistema. Actualment aquestes proves de nivell s'acostumen a realitzar anotant els resultats en en paper. Mantenir un registre d'aquestes proves pot permetre en un futur poder comparar-les i comprovar el progrés i evolució dels alumnes.

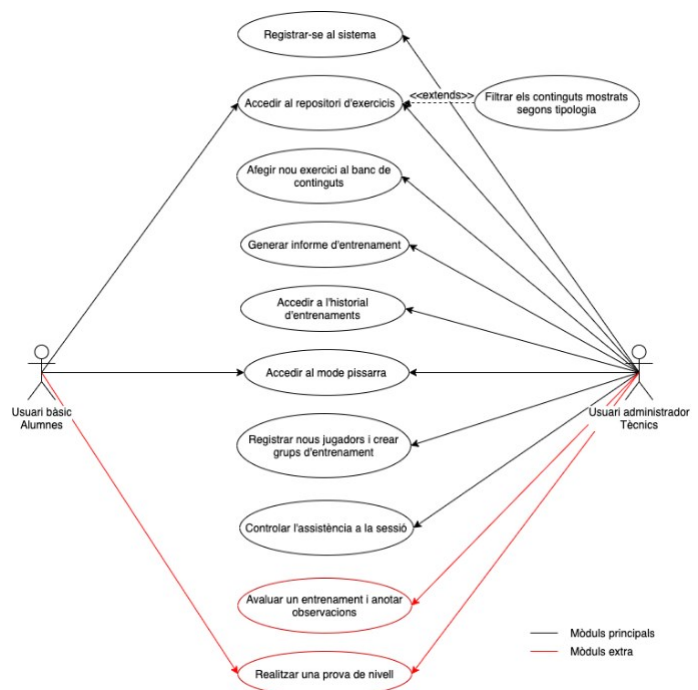


Fig. 2: Diagrama de casos d'ús per als dos rols establerts d'usuaris

2 PLANIFICACIÓ

Les tasques s'han dividit en tres blocs principals: un orientat a la realització del informe inicial i el set up del projecte, un segon centrat en el desenvolupament complet de les dues versions del producte tenint en compte les fases naturals de disseny, implementació i test, i per últim un darrer bloc basat en la redacció exhaustiva del procediment i l'extracció de conclusions de cara el document final.

Donat que les tecnologies escollides per realitzar el desenvolupament són relativament noves i no es té cap experiència prèvia la primera fase del projecte resulta especialment important. La cerca d'informació sobre el framework i les llibreries utilitzades és clau per tal de fer-se una idea sobre quins reptes ens trobarem durant la codificació. Una vegada s'obté un *background* prou sòlid s'inicia el desenvolupament (9/3/2020) part principal del projecte que té una previsió d'allargar-se fins a finals de maig i que inclou des del disseny de les vistes prototipus de la webapp fins el modelat de la petita base de dades o la confecció de la API Lumen i els dos front-ends: PWA i Hybrid. Finalment, com a part del bloc d'user testing i extracció de conclusions s'han fet públiques les dues versions de l'aplicació per demanar l'opinió de possibles usuaris en un petit formulari i s'han generat gràfics de rendiment per fonamentar la comparativa.

3 METODOLOGIA

Per desenvolupar el seguit de tasques plantejades en el projecte s'utilitzarà la metodologia *Waterfall*. Metodologia que consta de sis etapes diferenciades que esdevenen una rere l'altre a mesura que es van finalitzant. Aquestes etapes són: l'anàlisi de requisits, disseny del sistema, disseny de l'app, codificació, test i implantació o manteniment [2].

S'ha escollit aquesta metodologia vers altres alternatives del tipus àgil degut a que el desenvolupament del projecte corre a càrrec d'un únic individu i la paral·lelització de les tasques és pràcticament inexistent.

El fet d'aplicar una metodologia no cíclica o iterativa té els seus inconvenients. La presa de requisits inicial ha de ser el més acurada possible per tal d'estalviar temps i esforç en etapes més avançades del projecte. De la mateixa forma, començar a dissenyar la webApp sense donar per finalitzada i validada la presa de requisits funcionals i secundaris pot produir la inversió de temps en dissenys i prototipus finalment descartats.

En aquest projecte en particular prenen especial rellevància les etapes de disseny de la webApp i codificació degut a que es volen obtenir dues versions d'un mateix producte. La versió nativa pot presentar variacions estètiques respecte la versió web degut a que està pensada per ser utilitzada en dispositius mòbils o de mida reduïda. En quant a la codificació, l'utilització de la tecnologia React Native presenta certs canvis respecte ReactJs, com els nous components o la generació de *stylesheets*, que requereixen d'una formació bàsica prèvia.

4 DESENVOLUPAMENT

Per tal d'afavorir la reutilització del codi, la escalabilitat del projecte i de seguir amb la planificació plantejada anteriorment el desenvolupament de les aplicacions Itennis s'ha realitzat de forma modular. A continuació es detalla breument el funcionament de cadascuna de les implementacions.

4.1 Lumen API

El projecte es fonamenta sobre una API generada en Laravel-Lumen. Un cop ideades les principals vistes de l'aplicatiu web i definits els continguts a mostrar és pot iniciar els desenvolupament en local del projecte. El primer pas realitzat ha estat la creació d'una petita base de dades via phpMyAdmin amb un recull d'exercicis de prova i d'usuaris per poder testear les crides a la API. Seguidament s'ha procedit a definir els *endpoints* de la API on s'emmagatzemaran les dades recollides de la base de dades. La sintaxi escollida de les rutes de cada *endpoint* segueix el següent patró: `$router->routerMethod('path', Controller@Method)`; on el *routerMethod* fa referència al tipus de request HTTP, el *path* on es publicaran les dades recollides i per últim el controlador i la funció encarregades de realitzar la consulta i retornar dades en la resposta [3]. Per a que aquest ecosistema funcioni correctament s'ha de configurar prèviament el fitxer `.env` situat a la carpeta arrel de tot projecte en Larval-Lumen. Aquest fitxer administra algunes variables de configuració com són el tipus de connexió a la base de dades, la definició del host, el port associat, el nom de la base de dades i les credencials d'accés.

4.1.1 Axios - Obtenció de les dades de la API

El client HTTP triat per obtenir les dades de cadascun dels *endpoint* de la API ha estat Axios. Client basat en *promises* que gestiona aquest aspecte de forma senzilla i que és funcional tant en navegadors com en ecosistemes de node.js [4]. S'instal·la i integra en un projecte com qualsevol altre

```
$router->get('/', function () use ($router) {
    return $router->app->version();
});

$router->get('exs', 'ReportController@showExs');
$router->get('exlist', 'ExsController@getExs');
$router->get('tipologies', 'NewExController@showTips');
$router->get('newEx', 'NewExController@newExToDB');
$router->get('selected', 'ReportController@getExsFromIds');
```

```
public function showExs(){
    $list_data = [];
    $list_data = DB::select("SELECT Exercise.exID ,Exercise.exName , Tipology.tipName
                            FROM Exercise
                            JOIN Exercise_Tipology ON Exercise.exID = Exercise_Tipology.exID
                            JOIN Tipology ON Exercise_Tipology.tipID = Tipology.tipID ");
    foreach($list_data as $exercise){
        $exercise->id = $exercise->exID;
        $exercise->label = $exercise->exName;
        $exercise->group = $exercise->tipName;
        unset($exercise->exID,$exercise->exName, $exercise->tipName );
    }
    return $list_data;
}
```

Fig. 3: Rutes dels *endpoints* i exemple de recuperació de dades via consulta a base de dades

llibreria o component. Ha jugat un paper fonamental en les crides a `componentDidMount()`, les dades desitjades arriben en la resposta de la petició GET i només cal formatjar-les per a que el component les accepti.

4.2 Creació de l'aplicació web amb React

4.2.1 Mòdul de registre i Login

Mòdul tradicional d'accés al web i registre en base de dades realitzat per via de formularis. Un dels aspectes a destacar en aquesta implementació és el tractament de la contrasenya com a dada privada. És per això, que aquesta no pot es pot incorporar en una petició HTTP si no es troba xifrada. En el prototip s'ha optat per un hash SHA-256 de forma provisional.

4.2.2 Mòdul de consulta - Repositori d'exercicis

Per aquest mòdul s'ha usat un altre component estàndard. La vista conté un filtre de cerca d'exercicis per nom i a continuació un llistat interactiu de Collapsibles. Component utilitzat per mostrar i ocultar continguts amb un sol click i que ens permet veure la vista en detall dels exercicis que vulguem consultar. El filtre per nom s'ha realitzat a partir d'una llista de mida variable que inclou els exercicis on el nom concorda amb l'input proporcionat per l'usuari.

4.2.3 Mòdul Generador d'informes d'entrenament

Mòdul encarregat de generar un report d'un possible entrenament confeccionat per l'usuari. A partir d'un component `MultiSelect` amb els exercicis indexats per tipologia l'usuari selecciona els exercicis que vol incorporar al informe. Les ID's dels exercicis s'afegeixen a la url de la request i són interpretades pel controlador per tal de recupera l'informació necessària per elaborar el document. A continuació, les dades s'incorporen a un *template* en HTML del document i s'envia com data de la resposta. A la banda del client el *package* `react-to-pdf` crea el document a partir d'un *screenshot* d'un component que renderitza el codi HTML. Aquesta aproximació és limitada i per això els informes resultants tenen una extensió màxima d'una pàgina.

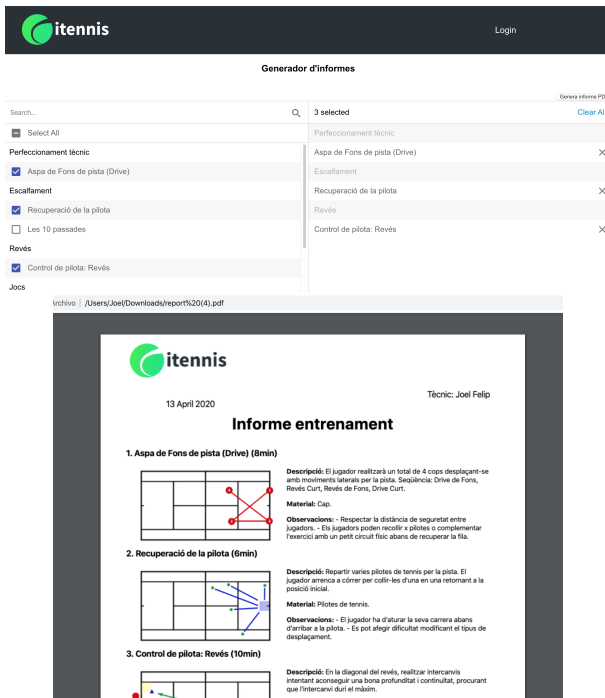


Fig. 4: Component multiSelect i exemple d'un report generat

4.2.4 Mòdul Pissarra tàctica-Canvas

En aquest mòdul s'ha fet ús de dos components. Un canvas com a llenç i d'un ColorPicker com a paleta de colors pels traçats. La associació del color seleccionat en la picker amb el color del traç del canvas és fa actualitzant el prop *brushColor*. Per altre banda, la vista permet canviar el *background* del canvas seleccionant qualsevol de les imatges proporcionades. Aquesta funcionalitat està implementada ocultant el fons del component canvas i renderitzant en el seu lloc la imatge corresponent.

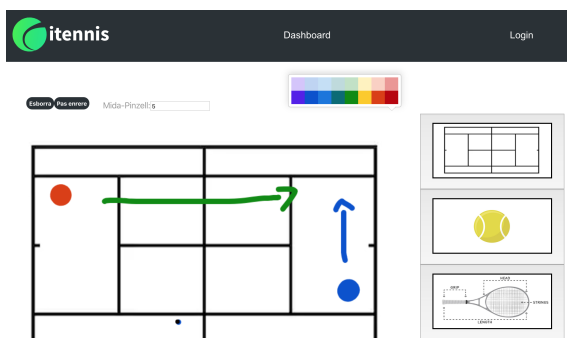


Fig. 5: Vista del mòdul de dibuix - Canvas, ColorPicker, Backgrounds

4.2.5 Mòdul Proves de Nivell

Aquest mòdul pretén adaptar una prova de nivell convencional en paper al format digital. Un dels principals handicaps que presenta aquest tipus de proves en el tennis és que es basen en la repetició de colpejos i conseqüentment el nombre de inputs i ítems a avaluar augmenta ràpidament. Per mantenir una quantitat de variables reduïda en pantalla s'ha fraccionat la prova per fases. A cadascuna de les

fases es pot accedir i navegar mitjançant els botons "anterior" "següent" d'un component *Stepper*. Component popular en aplicacions mòbils degut a que permet mostrar de forma progressiva el contingut i no sobrecarregar les interfícies. Les notes de d'execució tècnica i control de cada colpeig s'introdueixen en cadascuna de les files d'una taula via input numèric convencional i via Dropdown respectivament. Al finalitzar la prova es mostra un missatge a l'usuari en funció de si s'han complert o no els criteris per superar-la i permet reinicialitzar-la novament. Resulta un mòdul que tant pot servir als tècnics per efectuar un control esporàdic del nivell dels alumnes, com una eina autoavaluativa.

4.3 Conversió d'una web a una PWA

Una PWA és una responsive web dotada d'un manifest i d'un service worker. El manifest és un fitxer JSON amb un seguit de propietats que serveix per informar al navegador quins seran els elements que es mostraran en cada vista de la pàgina web i quin estil rebran. També s'especifica quina serà la pantalla d'inici en la navegació o controlar els icons i títols amb els que es podrà accedir a la PWA des de l'escriptori o dispositiu mòbil. Aquest manifest.json ha de ser cridat en el `<head>` del codi HTML de la vista principal de la pàgina web i per comprovar-ne el correcte funcionament es poden utilitzar les Chrome Dev Tools i consultar l'apartat Manifest un cop accedim a una web dotada de d'aquesta informació [5].

Per altre banda, el service worker és un fitxer javascript que s'executa en segon pla i que actua com a proxy de xarxa. Permeten controlar la forma en la que es processen les request en la teva web app i emmagatzemar en memòria cau vistes i dades sense que l'usuari se'n doni compte. Estan basats en *promises* de javascript. Possibiliten l'execució offline de la web, milloren la velocitat general del sistema i afavoreix l'estalvi de recursos precisament els objectius core de les PWA [6].

El cicle de vida d'un service worker consta de tres fases principals: el registre, l'instal·lació i l'activació. En la fase de registre es comunica al cercador on s'ubica la nostra web i s'inicia la descàrrega en segon pla. Registrar un service worker en la carpeta arrel d'un projecte el dotarà d'accés a tots els recursos. Durant el procés d'instal·lació es poden executar altres tasques i un cop instal·lat romandrà en espera fins rebre l'event d'activació. Tant si el procés d'instal·lació o el d'activació fallen el service worker passa a un estat redundat producte del rejepte de les *promises*.

4.4 Creació d'una aplicació híbrida a partir d'una PWA

A diferència de React, React-Native no utilitza HTML per renderitzar l'aplicació. Native proporciona components propis que funcionen de forma similar i són compatibles per a iOS i Android simultàniament. Per exemple, un tag `<div>` de React podria substituir-se per un component `View` de Native o un tag `<p>` per un component `Text` respectivament. El fet de que el codi no sigui renderitzat en una pàgina HTML provoca que no es puguin reutilitzar moltes de les llibreries usades en React. Aquestes s'han de substituir per unes de compatibles i s'ha d'adaptar el codi per adoptar els nous components. La adaptació d'un aplicació

a aquesta tecnologia és un procés que sovint comporta modificacions en el format de les dades recollides de la API. En referència al disseny i edició visual dels components, React-Native fa servir *stylesheets* en Javascript. Les característiques principals dels *stylesheets* són:

- Notació camelCase en les propietats d'estil.
- Utilització d'unitats de mesura estàndard interpretades posteriorment com a píxels.
- Es pot utilitzar codi dins la definició dels estils.
- Permet reutilitzar estils fent referència al ID atorgat a l'estil que volem re-aprofitar.
- La Flexbox funciona de la mateixa forma que en CSS amb alguna excepció. El paràmetre flex només accepta un nombre i la flexDirection està declarada per defecte com a column.

Una de les característiques principals de les aplicacions mòbils és el seu propi sistema de routing i navegació interna. Mentre que en una web es possible navegar pels continguts de la mateixa realitzant modificacions de la URL base en una aplicació no és possible realitzar-ho d'aquesta manera. En la versió híbrida de l'aplicació Itennis s'ha optat per realitzar una navegació basada en el renderitzat condicional. Degut a la senzillesa de l'aplicació i la estructuració principal del menú en format grid permet agrupar cada vista en un sol component i renderitzar-lo en funció d'una variable estat. D'aquesta forma per navegar entre vistes només cal modificar el valor de l'estat que controla la vista actual. Per a navegacions més complexes generalment s'utilitza un stack de navegació per tal de simular una navegació en web. Els components/vistes es guarden en una pila mentre s'accedeix a ells mitjançant els mètodes de *push* o *replace*. En cas de que un objecte hagi estat invocat per via d'un *push* es podrà retornar a la pàgina anterior realitzant un *pop*. Si la invocació ha estat un *replace* la pila ignora les vistes apilades anteriorment al *stack* i quedaran inaccessible mentre duri la sessió de l'usuari [7].

4.4.1 Paper de l'algorisme Flexbox en la confecció de layouts

Partint de la base que, en dispositius mòbils, la gran majoria d'aplicacions suporten *layouts* en format horitzontal i vertical és convenient conèixer aquesta propietat junt amb els seus props principals. Per defecte, flexbox funciona en React Native de la mateixa forma que ho fa en CSS orientat a web però el valor base de la direcció del contingut està inicialitzat com a column. En el cas d'Itennis el disseny del *layout* mòbil està ideat en un format horitzontal i els components són distribuïts realitzant combinacions de les props: *flexDirection*, *alignItems* i *justifyContent*. A internet es troben multitud de guies i documentació visual que ajuda a comprendre el concepte i les seves propietats [8].

5 PUBLICACIÓ DE LES SOLUCIONS

5.1 Deploy

Per fer públiques les aplicacions s'ha de realitzar un *deploy* tant de la API com de les interfícies. En el cas

d'Itennis s'ha optat per fer servir el servei Laravel Forge, una plataforma que permet desplegar projectes web en PHP de forma còmode [9]. El procediment és senzill i es basa en els següents passos: registre i adquisició d'algun dels plans oferts, triar el tipus de servidor desitjat (AWS), configurar-lo i afegir-hi un parell de *keys SSH* per accedir-hi posteriorment, vincular el compte Forge amb Github o el repositori on tinguem emmagatzemat el codi, crear el site i per últim, entrar a la base de dades creada per defecte en el servidor via SSH (Sequel Pro) i fer l'import de la BD treballada de forma local en phpMyAdmin o similar.

Si bé el procediment és força simple i a Internet es troben nombroses guies hi ha un seguit d'aspectes clau a tenir en compte a l'hora de configurar l'ecosistema. Si la interfície es troba allotjada en un domini segur dotat d'un certificat SSL la API també ha de estar allotjada en un domini segur per a que el navegador no bloquegi les connexions. Per altre banda, cal revisar el fitxer *.env* on es guarden les variables d'entorn per comprovar que les credencials d'accés a la BD són correctes en tot moment. Per últim, s'ha d'habilitar, si és necessari, l'intercanvi de recursos d'origen creuat (CORS). La política de seguretat SOP prohibeix que es carreguin dades provinents de servidors aliens, és per això que s'ha d'especificar explícitament aquesta autorització en les capçaleres de les respostes HTTP. Si ambdós administradors web coneixen i aproven l'intercanvi de dades no té sentit que aquestes request siguin denegades. Tots els navegadors actuals suporten el CORS i l'enviament de sol·licituds i respostes succeeix prou ràpid per a que l'experiència d'usuari no es vegi afectada.

En cas que es vulgui fer pública l'aplicació híbrida s'haurà de considerar revisar la implementació i les dependències dels components de React-Native seguint la documentació proporcionada per cada creador per tal d'adaptar la build al sistema operatiu desitjat sigui iOS o Android. Cal tenir present que les configuracions per defecte de disseny per a Android i iOS no són exactament iguals (*backgroundColor*..).

5.2 Publicació

El primer pas a realitzar quan es decideix fer pública una aplicació és escollir en quina o quines plataformes volem llistar-la. Actualment els tres alternatives més conegudes són Google Play, iOS App Store i la Windows Store.

Mentre subscriure's a la iOS App Store té un cost de 99 euros anuals, fer-ho a Google Play suposa una sola cuota de 25 euros i el registre a la plataforma de Microsoft és gratuït. Només aquestes dues últimes plataformes permeten la publicació de PWA creades per desenvolupadors freelance, si es vol publicar en la iOS Store s'ha de realitzar un procediment previ per tal d'acreditar la legalitat del negoci o entitat publicadora, tràmit que en ocasions es pot allargar diversos dies.

Un cop registrat a les pertinents plataformes només cal crear el app package i pujar-lo als repositoris. PWABuilder és una eina gratuïta que en funció de la URL proporciona retorna el seguit d'instruccions i fitxers necessaris per completar el package (certificats de seguretat, icones de diferents mides...) i deixar-lo llest per publicar. En cas de

no disposar d'un dispositiu Mac per tractar el xCode project sempre es pot contractar un servei Mac en cloud i fer el built del projecte remotament [10].

Un cop publicades, les PWA passen un filtre per tal de confirmar que el projecte no sigui un plagí, contingui dades sensibles o la informació i imatges estiguin en els formats establerts. Mentre que a Microsoft i Apple el filtre sempre ha estat humà complementat per tests automàtics, a Google no va ser fins al 2015 quan degut a una sobrecàrrega d'aplicacions de baixa qualitat o d'intencions malicioses l'empresa va considerar establir un equip de moderadors que revisessin més estrictament les propostes dels desenvolupadors [11].

6 RESULTATS DEL PROJECTE

En el següent apartat realitzarem un anàlisi comparatiu del comportament de la PWA i la versió Híbrida en un mateix dispositiu Android. Analitzarem el pes de cadascuna de les apps, els recursos que consumeixen, la fluïdesa i les diferències entre les respectives interfícies.

En primer lloc hem descarregat ambdues aplicacions. La app híbrida ha estat descarregada des del market (Google Play) i la PWA accedint a la web de itennis des del navegador Google Chrome dirigint-nos al botó 'Menú' i seleccionant l'opció 'Afegir a la pantalla d'inici'. Aquest procés crearà un accés directe a la pàgina web en l'escriptori del dispositiu Android o bé instal·larà l'aplicació com a PWA si la web ho permet.

El temps de descàrrega i instal·lació de l'aplicació híbrida és superior al de la PWA. Mentre l'aplicació híbrida triga 15 segons en descarregar-se del market i uns 20 segons en instal·lar-se en el dispositiu la PWA triga poc més de 5 segons en descarregar-se, instal·lar-se i estar operativa. Això es deu, a que l'aplicació híbrida incorpora els recursos estàtics que necessitarà mentre que la PWA només requereix l'estructura bàsica i va descarregant el contingut a mesura que li fa falta.

En referència al pes que ocupen les dues versions en el dispositiu, la PWA surt clarament guanyadora. L'aplicació híbrida pesa 112MB mentre que la versió PWA ocupa només 479KB. El pes de la versió PWA es pot incrementar a mesura que el navegador emmagatzemi en memòria cache certs continguts però, tot i així, sempre serà inferior a la versió híbrida.

Un cop utilitzades les apps durant uns minuts procedim a comprovar el consum de recursos de cada alternativa (Figura 6).

En quant a fluïdesa, els dos prototips responen correctament i els continguts es mostren prou ràpid com per a que l'experiència de l'usuari no es vegi entorpidida. En referència a les interfícies, s'ha optat per dotar a la PWA una imatge més estàndard en format web mentre que la versió híbrida s'adapta més als canons de les aplicacions mòbil. Una de les principals virtuts que presenten les PWA és que poden pràcticament emular l'estètica de les aplicacions híbrides fent que els usuaris no aconseguixin diferenciar-les a simple vista.

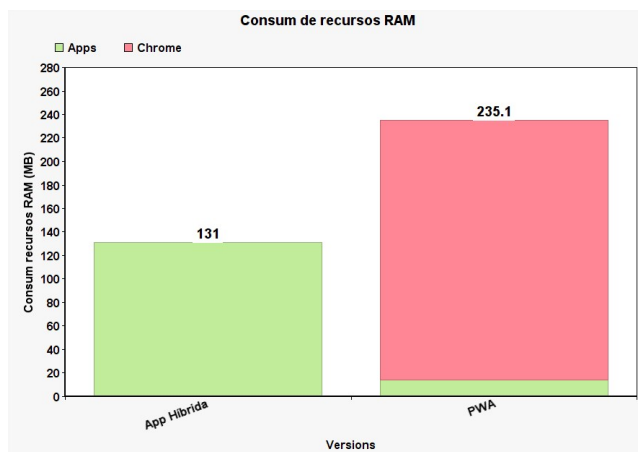


Fig. 6: Comparativa entre el consum de memòria RAM

Com podem observar en la Figura 6, la PWA atribueix el consum de memòria RAM al navegador Chrome tot i que aquest es trobi tancat. És per això, que cal incloure'l i tenir-ho en compte en la comparativa.

7 VALORACIÓ DE LES SOLUCIONS

En la següent secció es valoraran els pros i contres que comporten aquest tipus de desenvolupaments. Es destacaran aquelles característiques que han resultat més rellevants.

El temps invertit en desenvolupar una PWA o una aplicació híbrida des de zero és aproximadament el mateix. Les dues versions necessiten coneixements previs en desenvolupament web, UI/UX mòbil i gestió d'API's. Una alternativa per agilitzar el procediment seria partir d'una web ja existent i realitzar la conversió a PWA. El cost que comportaria realitzar cadascun dels desenvolupaments seria també força similar, tenint en compte que el producte final són eines accessibles des de diverses plataformes.

Les capacitats d'interacció de les PWA amb el hardware/software dels dispositius ve limitada per la seva pròpia arquitectura mentre que les aplicacions híbrides hi tenen accés per via de *plugins*. Cada any els navegadors incorporen funcionalitats per dotar a les PWA aquestes capacitats d'interacció.



Fig. 7: PWA vs Apps Híbrides [12]

Per últim, l'accessibilitat que tenen les aplicacions híbrides està limitada a que només es poden trobar al market on estan publicades mentre que les PWA són, a més a més, accessibles i descarregables des dels navegadors web. Aquest fet comporta que el propietari de l'aplicació no cal que registri ni tramiti la publicació del software més enllà del *deploy* en un servidor i gestió d'un domini. Per contra, cal valorar que dispositius amb cercadors des-actualitzats poden afectar negativament el rendiment de les PWA provocant inclús el tancament inesperat de l'aplicació.

8 RESULTATS DE L'ENQUESTA

Per tal de contextualitzar l'ús que li dona la societat als seus dispositius mòbils i més concretament sobre quin tipus d'aplicacions usen amb més freqüència, s'ha realitzat una enquesta amb els següents resultats:

- Un 24% dels enquestats utilitzen dispositius Apple. Dada que concorda aproximadament amb el 26,72% que proporciona gs.statcounter.com d'usuaris de iOS en tot el món.
- El 19% dels usuaris enquestats disposa de menys de 5GB lliures d'emmagatzemament mentre que un 55% té més de 20GB encara disponibles.
- D'entre les webs més visitades destaquen: la xarxa social Twitter, les plataformes de continguts audiovisuals Youtube i Twitch, el serveis de correu electrònic Gmail i Outlook, webs de premsa digital i de consulta d'informació.
- Respecte a les aplicacions més utilitzades destaquen: Whatsapp (Nativa), Instagram (React Native - Híbrida), Twitter (Híbrida - Nativa), Spotify (disposa de versió PWA), Youtube i Gmail (Híbrida).
- El 57% dels enquestats utilitza el navegador del mòbil de forma ocasional (20% o inferior del temps d'ús diari total).
- El 67% dels enquestats prefereix realitzar compres online des d'un ordinador. un 17% prefereix fer-ho des del mòbil i un 14% li és indiferent. El 2% restant no acostuma a realitzar compres online.

Tot i que realitzar compres online des dels dispositius mòbils és una pràctica que es troba en creixement degut a l'ús cada cop més estès d'aquests dispositius, hi ha estudis que demostren que comprar articles des d'un ordinador provoca prendre millors decisions sobre quins articles estem adquirint i redueix el nombre de productes comprats per error o retornats. La principal raó que fonamenta aquest plantejament és que les versions per a mòbil d'aquestes botigues no presenten la informació dels productes de la mateixa manera que és fa en una versió per a *desktop*. Ho fan d'una forma reduïda, informació accessible via desplegable o botons que mostren continguts inicialment ocults. La solució a aquest conflicte pot semblar aparentment senzilla. A simple vista, mostrar la mateixa informació en totes les versions de l'aplicació per a que els usuaris puguin fonamentar les seves decisions de la mateixa

forma en totes les plataformes resolldria aquest conflicte. Però a la pràctica, l'experiència d'usuari i la usabilitat de les aplicacions mòbil juguen un paper massa rellevant com per convertir les aplicacions en eines més difícils d'usar, interpretar o recarregades d'informació en pantalla.

- Els criteris d'usabilitat i d'User Experience que prioritzen els consultats en pàgines web són: els temps de càrrega mínims (81,4%), les mesures de seguretat (verificacions, controls d'accés..)(69,8%) i l'estètica dels llocs web (58,1%).
- Quant a aplicacions mòbils ens referim els resultats varien lleugerament: els temps de càrrega mínims (83,7%), una navegació intuïtiva i simple (74,4%) i l'estètica i disseny de les apps (60,5%).

De les dades recollides en les enquestes podem extreure que els usuaris no es decanten per una app o per un altre basant-se en quina tecnologia ha estat implementada o quina estructura presenta internament. Al cap i a la fi, l'usuari cerca un rendiment òptim en tot tipus de condicions i dispositius i una aparença treballada alhora que actualitzada. El fet de que tant les PWA com les apps híbrides puguin incorporar *Splash Screens* o ser instal·lades de la mateixa forma, accedint al market o store i descarregant-les en els nostres dispositius, oculta encara més les diferències que puguin presentar les dues versions i les unifiquen sota el mateix concepte d'aplicació mòbil.

9 CONCLUSIONS PROVISIONALS I LÍNIES OBERTES

9.1 Conclusions

A continuació s'exposaran els conceptes extrets en clar producte de l'experiència desenvolupant ambdós prototips. Amb aquestes conclusions s'intenta definir un marc que ajudi a desenvolupadors software, emprenedors i negocis a prendre decisions sobre la definició de les seves aplicacions en funció del *target* dels clients, les capacitats econòmiques i de recursos humans disponibles i el temps que es desitgi invertir. En l'actualitat la planificació i la correcta orientació dels projectes és un factor clau que acompanya la presa de requeriments i funcionalitats de qualsevol idea de negoci. Dedicar-hi estona pot aconseguir escurçar els temps d'entrega i evitar desestimar aproximacions que es vegin limitades a mig procés.

Tant la llibreria React com el framework React-Native són tecnologies que precisen d'un coneixement previ i una certa familiarització amb el llenguatge javascript. Usuaris amb una base conceptual sòlida en javascript es poden adaptar fàcilment a la notació i desenvolupar els seu propis projectes en un curt espai de temps. D'altre banda cal destacar tres factors que afecten directament el desenvolupament de software amb aquestes eines:

- La documentació: Són tecnologies que han guanyat popularitat en els últims anys i es troben en un procés de creixement constant. És per això, que accedir a una documentació completa, oficial i estable és complexa. Sovint, membres de la comunitat generen les seves

guies i redacten les instruccions per fer ús dels seus components.

- Les versions: Relacionat amb el punt anterior, el framework rep actualitzacions periòdicament que poden comportar problemes de compatibilitat entre components o modificacions en la notació utilitzada.
- Front-end only: ReactJS es basa en cobrir les necessitats dels usuaris a nivell d'interfície. Per a qualsevol projecte web serà necessari complementar-lo amb altres tecnologies encarregades de la part administrativa o back-end.

Com s'ha pogut veure en la secció de desenvolupament, la creació del prototip híbrid ha resultat ser fins a un 30% més ràpida que el desenvolupament web. Aquest fenomen és produït gràcies a la reusabilitat parcial del codi que comparteixen les tecnologies. La reutilització de la comunicació amb la API i de la lògica de les vistes ha permès una adaptació als components nadius més eficient.

Així doncs, les PWA i les aplicacions híbrides són clares candidates a competir ben de prop amb els desenvolupaments nadius. Les dues alternatives exposades en aquest projecte serien enfocades a projectes o aplicacions on l'ús dels recursos externs del dispositiu com la càmera, el lector d'empremtes dactilars o el GPS no fora necessari. Altres criteris que poden ajudar a prendre una decisió ferma poden ser a quin tipus de dispositius específics va dirigida la nostra aplicació o si el servei final es vol monetitzar d'alguna forma sigui via càrrec per la descàrrega o de si permetria la integració d'anuncis. Una hipotètica versió final del software presentat Itennis seria enfocada en un ús en dispositius mòbils/phablets i tablets i podria incorporar anuncis relacionats amb el àmbit esportiu intentant que siguin poc intrusius o afectin el menys possible l'experiència de l'usuari.

Les tendències actuals indiquen que de tot el temps que dediquen diàriament els usuaris als seus telèfons mòbils només un 10-15% correspon a la consulta de continguts des d'un cercador web. Els usuaris cerquen solucions estables i amb un bon rendiment en tot tipus de situació i en aquest sentit les aplicacions híbrides es situen un pas per davant. En canvi, plantejaments com els que presenta el sector del e-commerce o els videojocs encaixen amb el concepte de les PWA. Aconseguir que una aplicació web convencional pugui ser adaptada al format mòbil i adopti el comportament i característiques del dispositiu és el que pretenen moltes empreses que volen créixer en la compra-venta online. El fet de poder enviar una notificació a un usuari amb informació sobre nous productes, restocks o ofertes és un clar reclam per que aquest accedeixi novament a la aplicació i realitzi una compra. Aquest sistema pot substituir a les tradicionals trucades per telefon o a l'enviament de e-mails que tenen un risc més elevat de resultar ignorats.

En l'apèndix de l'article es poden trobar els enllaços per visitar o provar les dues versions desenvolupades.

9.2 Línies obertes

En aquesta secció s'exposen tots aquells apartats, funcionalitats no implementades o defectes trobats durant el desenvolupament que podrien donar continuïtat al projecte fins

obtenir un resultat sòlid. Una clara millora de cara a un futur seria el acabar la funcionalitat del generador d'informes. L'eina utilitzada en la solució de React limita la creació d'informes *single page* mentre que la opció triada en la versió híbrida permet generar informes de més d'una pàgina. Per altre banda, els exercicis podrien presentar-se de forma ordenada segons el desenvolupament general d'un entrenament (escalfament, bloc tècnic, bloc tàctic, tornada a la calma) així com incorporar informació addicional al report com l'autor o dades identificatives del club al que pertany. Per tal d'afectar al mínim l'experiència de l'usuari i el rendiment de la solució, seria convenient migrar la generació del document PDF a la banda del servidor. A curt termini s'haurien de perfeccionar el sistema de login i accés als continguts així com millorar els filtres per tipologia en el repositori d'exercicis. Tenint en compte que el temps de cerca serà en moltes ocasions força limitat, un factor clau és que els continguts es mantinguin ben classificats i la cerca per filtres resulti realment efectiva. El mòdul que permeti realitzar el control d'assistència dels alumnes als entrenaments seria una altre implementació que restaria per produir-se. Dins una eina pensada per utilitzar-se en pista, té sentit dur un control dels alumnes a tots els nivells possibles, tant administratiu i coordinatiu com de seguiment i avaluació del progrés formatiu.

A nivell estètic, ambdues versions són clarament millorables. El prototip PWA té sentit si pot ser utilitzat per diferents dispositius. Per obtenir això, la solució ha de ser plenament responsive i la visualització dels continguts ha de ser òptima en gairebé totes les plataformes.

10 AGRAÏMENTS

En primer lloc, agraeixo al meu tutor de projecte Joan Giner la disposició i llibertat donada a l'hora de planificar i desenvolupar el treball de forma autònoma. Per altra banda també agraeixo a l'exalumne Nil Domene el temps invertit en compartir amb mi la seva experiència en el camp del desenvolupament mòbil i publicació d'aplicacions en diferents stores.

REFERÈNCIES

- [1] Powerplay Tennis Coaching, "Tennis Software" [En línia]. Disponible a: <http://powerplaytennis.com/tennis-software/> Consultat: 16 de Febrer de 2020.
- [2] Lucidchart, "What the Waterfall Project Management Methodology Can (and Can't) Do for You". [En línia]. <https://www.lucidchart.com/blog/waterfall-project-management-methodology> Consultat el 15 de febrer de 2020.
- [3] Lumen - Laravel, "HTTP Routing". [En línia]. <https://lumen.laravel.com/docs/7.x/routing> Consultat el 25 de març de 2020.
- [4] S. Kumar, "Managing API Requests in React Native Using Axios", 3 de desembre de 2019. [En línia]. <https://medium.com/better-programming/managing-api-requests-http-https-in-react-native-using-axios-9ebf75cbca9b> Consultat el 26 de març de 2020.

- [5] NC, “Que es y Como Configurar Manifest JSON en una Pàgina Web”, 29 de març de 2020. [En línia]. <https://blog.nubecollectiva.com/que-es-y-como-configurar-manifest-json-en-una-pagina-web/> Consultat el 2 d'abril de 2020.
- [6] jdonsan, “PWA: Conceptos básicos sobre Service Workers”, 17 de novembre de 2017. [En línia]. <https://elabismodennull.wordpress.com/2017/11/17/pwa-conceptos-basicos-sobre-service-workers/> Consultat el 10 d'abril de 2020.
- [7] T. Maroudas, ”Routing in React Native apps and how to configure your project with React-Navigation library”, 17 de setembre de 2018. [En línia]. <https://medium.com/building-with-react-native/routing-in-react-native-apps-and-how-to-configure-your-project-with-react-navigation-library-d8d58005bfe9> Consultat el 10 de maig de 2020.
- [8] D. Biran, “The Full React Native Layout Cheat Sheet”, 6 de desembre de 2017. [En línia]. <https://medium.com/wix-engineering/the-full-react-native-layout-cheat-sheet-a4147802405c> Consultat el 22 de maig de 2020.
- [9] Laravel Forge - #Deployments. [En línia]. <https://forge.laravel.com/docs/1.0/sites/deployments.html> Consultat el 6 de juny de 2020.
- [10] J. Gabriel Himango, “I built a Progressive Web App and published it in 3 app stores. Here’s what I learned”, 26 d'abril de 2018. [En línia]. <https://medium.com/free-code-camp/i-built-a-pwa-and-published-it-in-3-app-stores-heres-what-i-learned-7cb3f56daf9b> Consultat el 4 d'abril de 2020.
- [11] S. Perez, “Android App reviews may slow to over a week due to COVID-19 impacts, Google warns developers”, 17 de març de 2020. [En línia]. <https://techcrunch.com/2020/03/17/android-app-reviews-may-slow-to-over-a-week-due-to-covid-19-impacts-google-warns-developers/> Consultat el 11 d'abril de 2020.
- [12] K. Arshuk-Popov I. Stetsiuk, “Progressive Web Apps: Comparison of PWA with Other Approaches to Build Mobile User Experience”, 27 d' abril de 2020. [En línia]. <https://blog.griddynamics.com/pwa-research/> Consultat el 12 de juny de 2020.

APÈNDIX

A.1 Model de la base de dades

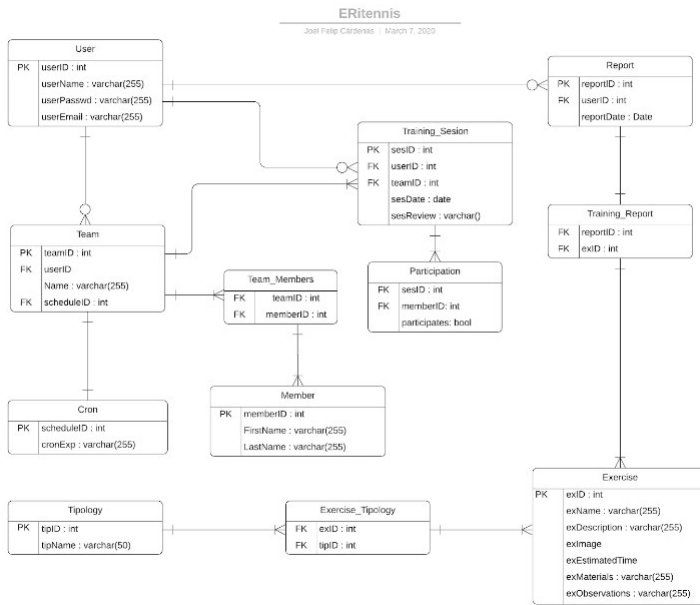


Fig. 8: Model de la base de dades

A.2 Planificació i Diagrama de Gantt

ID	Nombre	Duracion	Inicio	Terminado
1	TFG PWA+HybridApp	58,125 days?	14/02/20 8:00	4/03/20 17:00
2	Realitzar informe inicial	58,125 days?	14/02/20 8:00	4/03/20 17:00
3	Cercar info. marc teòric. Debat PWA VS HybridApp	43,125 days?	14/02/20 8:00	28/02/20 17:00
4	Estudi eines de desenvolupament - Laravel (Lumen)	46,125 days?	14/02/20 8:00	29/02/20 17:00
5	Model de dades del projecte (UML)	25,125 days?	25/02/20 8:00	4/03/20 17:00
6	Documentació de la planificació	12,125 days?	24/02/20 16:00	28/02/20 17:00
7	Milestone - Lliurament informe inicial	1,125 days?	8/03/20 8:00	8/03/20 17:00
8	Desenvolupament	331,125 days?	9/03/20 8:00	27/06/20 17:00
9	Disseny	58,125 days?	9/03/20 8:00	28/03/20 17:00
10	Disseny i prototip de vistes de la webapp	16,125 days?	9/03/20 8:00	14/03/20 17:00
11	Modelat de base de dades i generació de continguts	37,125 days?	16/03/20 8:00	28/03/20 17:00
12	Implementació	174 days?	28/03/20 17:00	25/05/20 17:00
13	Desenvolupament PWA	54 days?	28/03/20 17:00	15/04/20 17:00
14	Desenvolupament API d'interacció amb base de dades	30 days?	15/04/20 17:00	25/04/20 17:00
15	Documentació referent a la API. Argumentació querys	12 days?	25/04/20 17:00	29/04/20 17:00
16	Maquetació de les vistes. Front-end (Material)*	27 days?	29/04/20 17:00	8/05/20 17:00
17	Desenvolupament de la Hybrid Version	15 days?	8/05/20 17:00	13/05/20 17:00
18	Port a aplicació híbrida via React Native	30 days?	13/05/20 17:00	23/05/20 17:00
19	Modificacions en la maquetació	6 days?	23/05/20 17:00	25/05/20 17:00
20	Test	22,25 days?	25/05/20 7:00	1/06/20 17:00
21	User testing	22,25 days?	25/05/20 7:00	1/06/20 17:00
22	Realitzar Informe Final	78 days?	1/06/20 17:00	27/06/20 17:00
23	Documentació del procés i inconvenients	22,25 days?	1/06/20 17:00	9/06/20 3:00
24	Estudis bàsics de rendiment	22,25 days?	8/06/20 7:00	15/06/20 17:00
25	Extracció de conclusions, comparativa de versions	37,25 days?	15/06/20 7:00	27/06/20 17:00
26	Milestone - Lliurament Dossier + Proposta de presentació	1 day	28/06/20 7:00	28/06/20 15:00
27	Realitzar la presentació i el póster	10,25 days?	29/06/20 7:00	2/07/20 17:00
28	Milestone - Presentació a Tribunal TFG	1,25 days?	3/07/20 7:00	3/07/20 17:00

Fig. 9: Planificació temporal de les tasques

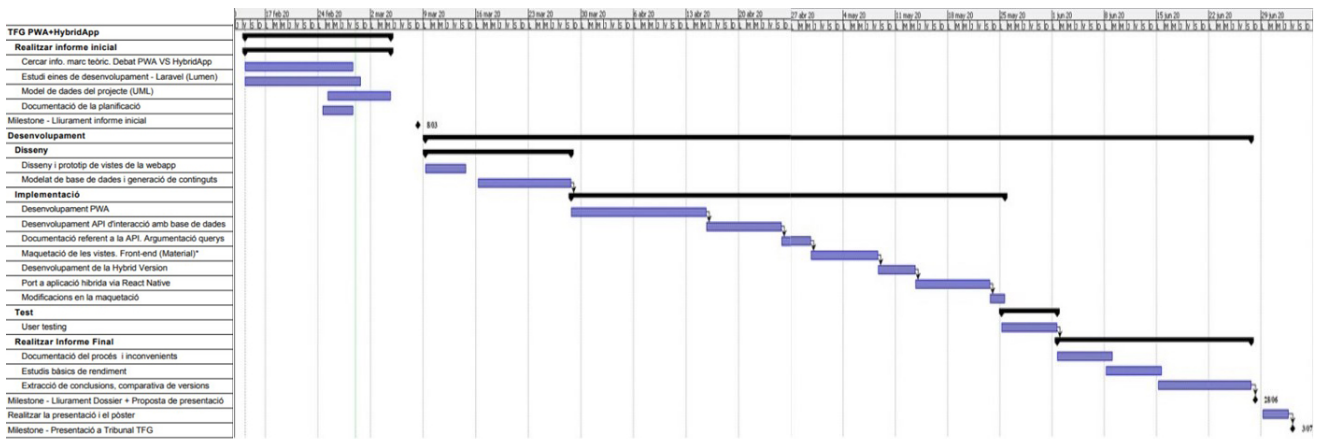


Fig. 10: Diagrama de Gantt de la planificació

A.3 Preguntes de la enquesta: Ús de tecnologies web i aplicacions mòbil

1. Edat:

- 0-18 anys
- 19-30 anys
- 30-50 anys
- 50-70 anys
- +70 anys

2. Sexe:

- Dona
- Home

3. Model de dispositiu mòbil personal: (Resposta oberta)

4. Memòria lliure en el dispositiu mòbil:

- 0-5 GB
- 5-10 GB
- 10-20 GB
- +20 GB

5. Top 3 de pàgines web més visitades: (Resposta oberta)

6. Top 3 d'aplicacions mòbil més utilitzades: (Resposta oberta)

7. Estimació aproximada en % del temps que utilitzes el navegador web en el mòbil: (respecte al temps total d'ús)

8. Si has de realitzar una compra on-line des de quin dispositiu prefereixes fer-ho?

- Mòbil/Tablet
- Ordinador
- Indistintament
- No acostumo a comprar online

9. Marca quins dels següents criteris d'usabilitat i UX valors més en una pàgina web (màxim de 3):

- Disseny/Estètica
- Velocitat (temps de càrrega mínims, evitar animacions...)
- Navegació intuïtiva
- Seguretat (accés, verificacions d'usuaris en registres, certificació...)
- Compatible en múltiples navegadors
- Contingut actualitzat
- Multilinguatge
- Rebre informació del sistema (missatges específics d'error, feedback, guia d'ús, documentació...)
- Altres...

10. Marca quins dels següents criteris d'usabilitat i UX valors més en una aplicació mòbil (màxim de 3):

- Disseny/Estètica
- Velocitat (temps de càrrega mínims, evitar animacions...)
- Navegació intuïtiva
- Incorpori les últimes tendències (interacció basada en veu, storytelling, gràfics 3D...)
- Actualitzacions freqüents
- Multilinguatge
- Capacitat de compartir informació mitjançant altres aplicacions
- Rebre informació del sistema (missatges específics d'error, feedback, guia d'ús, documentació...)
- Altres...

A.4 Visita les aplicacions

Versió PWA: <https://itennistool.herokuapp.com/>

Versió Híbrida: <https://play.google.com/store/apps/details?id=com.itennisnative>