

Aplicació amb tecnologia web: sistema de votació

Marc Jordana Vila

Resum– Avui en dia, hi ha moltes situacions en les quals s'ha de prendre una decisió en grup. Moltes vegades aquesta decisió es pren mitjançant un sistema de votació. Els sistemes de votació tradicionals no són gaire eficients, sobretot si es tracta d'un grup gran de persones. Alguns dels problemes de les votacions en grup són: s'ha de fer un recompte de vots, tothom ha d'estar en el mateix lloc, moltes vegades es pot veure què vota cada persona i el nombre de vots de cada opció. Aquesta situació pot condicionar les decisions dels votants. Aquest projecte pretén solucionar aquesta problemàtica, mitjançant una aplicació web que permetrà fer votacions en temps real.

Paraules clau– ReactJS, Firebase, NodeJS, Express, Cloud Firestore, Aplicació web, temps real, votació, Redux, Javascript, CSS

Resumen– Hoy en día, hay muchas situaciones en las cuales se debe tomar una decisión en grupo. Muchas veces esta decisión se toma mediante un sistema de votación. Los sistemas de votación tradicionales no son demasiado eficientes, sobre todo si se trata de un grupo grande de personas. Algunos de los problemas de las votaciones en grupo son: se tiene que hacer recuento de votos, todos los participantes tienen que estar en el mismo sitio, muchas veces se puede ver que vota la otra gente y el número de votaciones que tiene cada opción. Esta situación puede condicionar las decisiones de los votantes. Este proyecto pretende solucionar esta problemática, mediante una aplicación web que va a permitir hacer votaciones en tiempo real.

Palabras clave– ReactJS, Firebase, NodeJS, Express, Cloud Firestore, Aplicación web, temps real, votació, Redux, Javascript, CSS

Abstract– Nowadays, there are many situations in which you have to make a group decision. Many times this decision is made through a voting system. Traditional voting systems are not very efficient, especially if it is made by a large group of people. Some of the problems with group voting are: there has to be a vote count, it has to be in the same place, many times it is possible to see that each person votes and the number of votes for each option, this situation can condition the decisions of the voters. This project aims to solve this problem, using a web application that will allow you to vote in real-time.

Keywords– ReactJS, Firebase, NodeJS, Express, Cloud Firestore, Aplicación web, temps real, votació, Redux, Javascript, CSS



1 INTRODUCCIÓ

ACTUALMENT, hi ha aplicacions que permeten fer votacions, però no hi ha cap aplicació que ens ho permeti fer en temps real o si més no, no està enfocada a utilitzar-se

d'aquesta manera, a més a més, moltes d'aquestes aplicacions estan molt enfocades en un àmbit en concret, com pot ser el de l'educació. Per diferenciar la nostra aplicació de les que ja hi han al mercat, s'utilitzaran el maneig de dades en temps real i una interfície que sigui adequada a la majoria de situacions. El nostre projecte és una aplicació web, ja que d'aquesta manera, es podrà utilitzar en qualsevol dispositiu que tingui connexió a internet i un navegador. Gràcies al maneig de dades en temps real les votacions són més dinàmiques, ens permet conservar els avantatges de les votacions tradicionals i eliminar els inconvenients que

- Correo de contacto: marc.jordanav@e-campus.uab.cat
- Mención realizada: Tecnologías de la información
- Trabajo tutorizado por: Aaron Blanco Esteban
- Curso 2019/20

aquestes generen. L'aplicació s'ha realitzat utilitzant tecnologies actuals com poden ser ReactJS [1] i NodeJS [2]. També s'utilitza Firebase [3] com a sistema d'allotjament web i d'emmagatzemament de les dades.

2 MOTIVACIÓ

Des de fa cert temps tenia clar que volia escollir un projecte basat en tecnologia web, ja que havia fet algun projecte a la universitat i a la feina i m'havien resultat molt interessants, ja que a mesura que vas avançant, vas veient la interfície que s'està desenvolupant. Per tant l'únic que havia de decidir era a quin projecte m'inscrivía. Al començament dubtava entre diferents possibilitats, però finalment em vaig decidir per aquest projecte perquè s'havia d'utilitzar dades en temps real i vaig pensar que això seria una bona oportunitat per veure i aprendre com funcionava aquesta tecnologia.

3 ESTAT DE L'ART

Si pensem en aplicacions que tinguin una funcionalitat similar, a tothom li ve al cap eines com Kahoot [4] i Doodle [5]. Pel que fa a Kahoot és una aplicació que està molt enfocada en l'àmbit educatiu, amb una interfície que no és gaire adequada per altres situacions. En el cas de Doodle és una plataforma molt enfocada en decidir el dia que s'ha de produir un esdeveniment i no utilitza dades amb temps real. Hi ha altres aplicacions que tenen funcionalitats similars, però les ja mencionades són les més conegudes.

4 OBJECTIUS

L'objectiu principal del projecte és el desenvolupament d'una aplicació de votació. Aquesta aplicació ha d'utilitzar tecnologia web, preferiblement, actual. A més, el projecte ha d'utilitzar dades en temps real. Seguidament detallarem els objectius del projecte:

- Analitzar l'entorn, per trobar una solució que s'adapti a les necessitats reals de les persones i que permeti prendre decisions en grup, de manera més pràctica i efectiva que les votacions tradicionals.
- Configurar un servidor amb Firebase que permeti la comunicació en temps real entre l'aplicació i la base de dades.
- Desenvolupar un Back-End capaç de fer front a totes les connexions concurrents que hi puguin haver en un determinat moment.
- Desenvolupar una aplicació que permeti cobrir les necessitats dels usuaris que hàgim recollit en analitzar l'entorn, aquesta ha de ser en temps real i ràpida d'utilitzar.
- Que l'aplicació sigui fàcil d'utilitzar, perquè la puguin fer servir usuaris que no estiguin habituats en l'ús de les noves tecnologies.
- Que l'aplicació sigui funcional i es vegi bé en qualsevol pantalla, sigui des d'un ordinador o un dispositiu mòbil, donant prioritat a dispositius mòbils.

5 METODOLOGIA

La metodologia utilitzada pel desenvolupament del projecte és la del model iteratiu incremental. Aquesta metodologia consisteix a separar el projecte en petites parts, normalment funcionalitats i cada una d'aquestes parts es fa en una iteració. A cada iteració i sol haver-hi una fase de disseny, una fase d'implementació del que s'ha dissenyat i finalment una fase de test. Normalment quan s'acaba una d'aquestes iteracions, s'haurà creat una funcionalitat nova, que podrà ser testejada pel client. A partir dels tests d'aquestes funcionalitats es rebrà un 'feedback' i gràcies a això, es podrà corregir les parts que no acabin de funcionar correctament o les parts que no es comportin de la manera esperada. A més a més, com que el client pot anar provant funcionalitats i fent petites correccions, el projecte final, acaba sent més fidedigne al seu objectiu.

6 PLANIFICACIÓ

El projecte s'ha dividit en 8 iteracions, cada una enfocada amb una funcionalitat diferent. Com es pot veure a la TAULA 1, totes les iteracions tenen una durada aproximada de dues setmanes i mitja menys la primera. Es va escollir aquesta durada perquè es va calcular que per fer cada funcionalitat es necessitava com a mínim entre una i dues setmanes. A més a més, per donar una mica de marge, es va afegir mitja setmana al càlcul, d'aquesta manera també ens va permetre poder sincronitzar les iteracions amb la durada del projecte. El càlcul aproximat d'hores del projecte és de 254 hores sense incloure les reunions amb el tutor del projecte i els informes, tot i que finalment s'ha tardat més degut principalment a problemes amb el servidor que no es van tenir en compte al començament.

En general podem dividir el projecte en dues parts, la primera part, més dedicada a la planificació, al disseny i a preparar tota la infraestructura necessària pel desenvolupament del projecte i la segona part, enfocada totalment a desenvolupar les funcionalitats i a fer proves d'aquestes.

La primera iteració va ser una mica especial, ja que va començar abans que es fes la planificació. Es va decidir que aquesta iteració durés una setmana, perquè no hi havia tanta feina com en les altres i d'aquesta manera va permetre que la durada del projecte coincidís amb la durada de les iteracions. En aquesta iteració s'havia de fer la planificació del projecte, el disseny de l'aplicació utilitzant paper prototype i el disseny de la base de dades.

En la segona iteració s'havia de crear tota la infraestructura del projecte: crear el servidor amb Firebase, crear el repositori al Gitlab [6] i crear l'arquitectura del Back-End i del Front-End.

En la tercera iteració s'havia de crear la base de dades en el servidor, fer el desenvolupament de l'inici de sessió dels usuaris, tant per la part de Front-End, com per la part del Back-End i finalment testejar aquesta funcionalitat.

En la quarta iteració s'havia d'acabar tota la gestió d'usuaris; això incloïa el registre d'usuaris, que poguessin entrar usuaris convidats i eliminar la sessió d'usuari quan aquesta hagués caducat. També s'havia de fer la pàgina principal de la web.

Pel que fa a la cinquena iteració s'havia de fer totes les tasques relacionades amb la creació de formularis. Aques-

TAULA 1: PLANIFICACIÓ DEL PROJECTE

Iteracions			
Iteració	Inici	Final	Hores totals
1	22/02/20	29/02/20	15h
2	01/03/20	17/02/20	42h
3	18/03/20	04/04/20	40h
4	05/04/20	22/04/20	38h
5	23/04/20	10/05/20	38h
6	11/05/20	28/05/20	28h
7	29/05/20	16/06/20	23h
8	16/06/20	02/07/20	30h

tes tasques incloïen: crear i eliminar preguntes, crear i eliminar respostes i controlar quin tipus d'usuari podia fer-ho.

En la sisena iteració s'havia de fer tota la funcionalitat de resoldre els qüestionaris, això incloïa: fer una pàgina on es poguessin veure els qüestionaris, poder iniciar un qüestionari en concret i que els usuaris poguessin accedir-hi i contestar-lo.

La setena iteració era l'última en la qual s'havia d'implementar nova funcionalitat. La funcionalitat a realitzar en aquesta iteració era tota la relacionada amb la consulta de l'historial de qüestionaris, aquest historial ha de permetre veure tots els qüestionaris resolts i les seves preguntes i respostes.

En l'última iteració s'havia de dedicar tot el temps a fer proves de tota l'aplicació en general per assegurar-se que tot funcionés i en el cas que hi hagués algun error corregir-lo.

7 DESENVOLUPAMENT

En aquest apartat descriurem tot el que s'ha fet per aconseguir desenvolupar l'aplicació web. Com que aquesta secció és molt llarga la partirem en 4 parts, la configuració de Firebase, el control de versions, el desenvolupament del Front-End i finalment el del Back-End.

7.1 Configuració de Firebase

Per fer aquest projecte s'ha utilitzat Firebase, més concretament hem utilitzat els mòduls d'autenticació, de bases de dades, de hosting i de funcions. Per començar a utilitzar Firebase s'ha de crear el projecte i seguir els passos que posa a la documentació. Firebase té dos tipus de tarifes, la gratuïta anomenada Spark, que és la que hem utilitzat per desenvolupar aquest projecte i té la de pagament anomenada Blaze. La diferència entre les dues tarifes és que la gratuïta ho té tot limitat, tot i això els límits són força grans i no ens influiran en el desenvolupament del projecte.

En un principi no es tenia pensat utilitzar el sistema d'autenticació de Firebase, però per problemes que mencionarem més endavant, al final es va acabar utilitzant. Firebase ofereix una interfície de configuració d'usuari molt fàcil d'utilitzar i que et dona moltes opcions d'inici de sessió diferents. Per aquest projecte s'ha escollit que es podrà iniciar sessió de manera anònima i mitjançant correu electrònic i contrasenya. Firebase també et permet configurar un correu

electrònic automàtic per fer la validació del registre. En la figura 1 podem veure els dos tipus d'usuari que s'han configurat, el convidat i l'usuari normal.

Identificador	Proveïdors	Fecha de creación	Inicio de sesión	UID de usuario
marinavilla		10 abr. 2020	7 jun. 2020	1RGkRcM6i9Zu8z9c49bFTpkf0yp1
(anónimo)		21 abr. 2020	21 abr. 2020	2ASJ661hQseAAATwptSgvgNplq02

Fig. 1: Panell d'administració d'usuaris

Com a sistema de base de dades, Firebase ofereix dues opcions: Realtime Database [7] i Cloud Firestore [8] les dues són de tipus NoSQL. La diferència entre aquestes és que Cloud Firestore és una versió més recent de Realtime Database i inclou millores, per exemple, és més escalable i permet fer consultes més complexes. Per aquesta raó i perquè des de la mateixa documentació de Firebase o aconsellen, hem utilitzat Cloud Firestore. Per decidir el disseny de la base de dades es va fer un diagrama del model relacional com es pot observar en la figura 2. El diagrama de la figura 2 ha patit diverses modificacions, ja que el diagrama inicial li faltaven força coses perquè es pogués implementar tota la funcionalitat. La interfície oferida per Firebase, a part de moltes altres coses, permet veure les dades que hi ha inserides a la base de dades i configurar regles d'accés a la base de dades. Aquestes regles ens han portat problemes, ja que es poden modificar des de dos llocs diferents: des d'un fitxer que es genera quan crees el projecte a Firebase i des de la mateixa interfície web de Firebase. Al començament, com que no sabíem que en el projecte hi havia un fitxer de configuració, es van editar les regles des de la interfície web, però cada vegada que es penjava el codi al servidor, aquestes es sobreescrivien. Finalment s'ha configurat aquestes regles perquè qualsevol usuari amb una sessió vàlida pugui accedir a les dades.

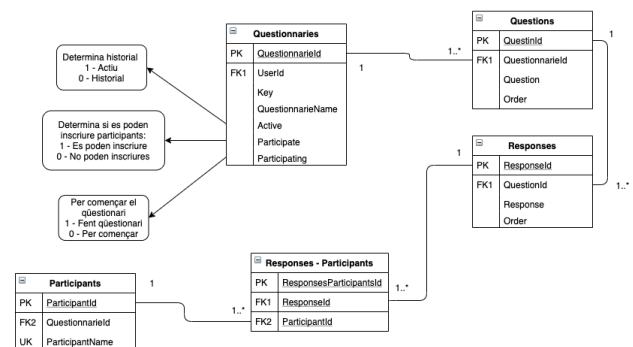


Fig. 2: Model relacional BD

Pel que fa al sistema de hosting no s'ha hagut de configurar res, simplement emmagatzemem els fitxers a la ruta on tens implementat Firebase, fas un build del Front-End i amb la comanda 'firebase deploy' es penegen els fitxers al servidor. Aquest servei també ofereix 2 dominis per defecte i et dona l'opció de crear-ne de personalitzats, però com que aquesta part de moment no tenia rellevància amb el projecte hem deixat els que ja oferia.

Finalment el mòdul de funcions és l'encarregat d'executar el Back-End. Fins al moment tots els sistemes de hosting

que s'havia vist es penjava el codi del Back-End al servidor i aquest es posava en funcionament mitjançant una comanda, però en aquest cas no és així. El mòdul de funcions emmagatzema només les funcions que reben les peticions, per tant només s'ha de desenvolupar les peticions que volem utilitzar. Aquest funcionament ens va portar problemes a l'inici del projecte, ja que es va començar el desenvolupament del Back-End com si es tractés d'un dels sistemes de hosting que s'havien utilitzat en altres projectes.

7.2 Control de versions

Per portar el control de versions s'ha utilitzat Gitlab. Al començament es tenia el dubte de si crear dos repositoris per separar el Back-End del Front-End o tenir-los junts en el mateix. Com que en aquest projecte només hi treballa una persona, no es va trobar cap avantatge significatiu en fer-ho per separat, per tant es va decidir crear només un repositori en el qual hi hagués tant el Back-End com el Front-End.

7.3 Front-End

Com ja hem dit anteriorment, per fer la interfície de la pàgina web hem fet servir ReactJS. Aquesta llibreria de JavaScript es basa en components. ReactJS té una gran quantitat de components que ja estan desenvolupats disponibles per qui els vulgui utilitzar. Per fer la nostra aplicació web hem utilitzat uns quants d'aquests components, ja que redueixen molt la quantitat de codi que s'ha de desenvolupar, seguidament s'explicarà els més importants i perquè serveixen.

Un dels components principals que hem utilitzat és React Router [9]. Aquest component s'encarrega de rendiritzar els altres components segons l'adreça web, aquesta funcionalitat és molt important, ja que ofereix una experiència molt similar a la de les pàgines web convencionals amb una velocitat molt superior, ja que ReactJS carrega tots els components al començament, per tant l'usuari no haurà d'esperar, ja que el canvi entre components serà instantani.

Un altre component important és el component de Firebase [10], aquest component ens permet fer consultes a la base de dades directament des del Front-End, això podria semblar que no és correcte, ja que se salta el paradigma de programació Model-Vista-Controlador. Com ja hem mencionat abans Firebase només utilitza les funcions de les peticions, per tant si volem aconseguir un servei de dades en temps real, no es pot connectar directament el Front-End i el Back-End mitjançant una connexió amb WebSockets [11]. El component de Firebase té una funcionalitat anomenada 'onSnapshot' que permet crear observadors que observen certs camps de la base de dades. Aquests observadors estan a l'espera que es produeixin canvis al camps que observen i quan se'n produeix un, el reben. Gràcies aquest funcionament podem utilitzar les dades en temps real en el Front-End.

Per acabar de mencionar els components més importants, hem utilitzat React Redux [12]. Redux és un contenidor d'estats que ens permet solucionar un dels principals problemes que té ReactJS, el maneig d'estats. Aquests estats es van heretant de components pares a components fills. Aquest problema es fa visible quan l'aplicació web és gran, ja que acaba havent-hi una herència d'estats que fa molt

diffícil el manteniment. Per fer el nostre projecte, s'han diferenciat dos contenidors d'estats, un que conté tots els estats d'usuari i un que conté els estats generals, d'aquesta manera és més fàcil de mantenir i d'afegir noves funcionalitats.

7.4 Back-End

Finalment, pel que fa a la part del Back-End com ja hem dit anteriorment s'ha utilitzat NodeJS amb Express [13], que és un framework minimalista compatible amb el sistema de Funcions de Firebase. També s'ha utilitzat paquets de Firebase que permeten connectar-nos a la base de dades i fer consultes i el paquet de CORS que permet habilitar el Cross-Origin Resource Sharing [14]. Com ja hem mencionat en altres apartats, només s'ha hagut de fer les funcions que s'encarreguen de recollir les peticions i un middleware que s'encarrega de validar l'usuari que fa la petició i comprova que el token no s'hagi caducat.

Per fer el desenvolupament del Back-End s'han utilitzat 3 tipus de peticions HTTP: el POST, el GET i el DELETE. Pel que fa a la petició del POST, s'ha utilitzat principalment per peticions que necessitaven més d'una dada per realitzar-se, normalment se'ls hi passava un objecte. D'altra banda s'ha utilitzat les peticions del GET, quan només s'havia d'enviar una dada al servidor, com pot ser l'identificador d'usuari. Finalment s'ha utilitzat la petició del DELETE, quan s'ha enviat una petició per esborrar alguna dada de la base de dades.

8 FUNCIONALITATS

En aquest apartat descriurem les funcionalitats que té l'aplicació i es farà una petita explicació del seu funcionament.

8.1 Registre

L'usuari ha de poder registrar-se a l'aplicació, com es pot veure en la figura 3, només farà falta un correu i una contrasenya. Una vegada l'usuari s'hagi registrat rebrà un email de confirmació del correu electrònic. En el cas que l'usuari posi un correu o una contrasenya que no compleixi els requisits mínims sortirà un avís per pantalla i no es podrà registrar fins que ho corregeixi. Un cop l'usuari s'hagi registrat s'inicia sessió directament i reconduïx l'usuari al panell principal.

Correu electrònic

exemple@example.com

Contrasenya

.....

Repeteix la contrasenya

.....

Registrar-se

Fig. 3: Formulari de registre

8.2 Iniciar sessió

L'usuari podrà iniciar sessió introduint la seva contrasenya i correu electrònic com es pot veure en la figura 4. En cas que la contrasenya o el correu no siguin correctes sortirà un avís per pantalla. També hi ha un altre tipus de sessió, l'usuari convidat. Aquest tipus d'usuari no podrà crear qüestionaris ni veure historials, només podrà participar en qüestionaris que hagin creat altres usuaris. Un cop s'hagi iniciat la sessió amb un usuari convidat s'anirà directament a la pàgina d'inscripció per participar en un qüestionari, d'altra banda, quan s'iniciï sessió amb un usuari normal, s'anirà al panell principal.

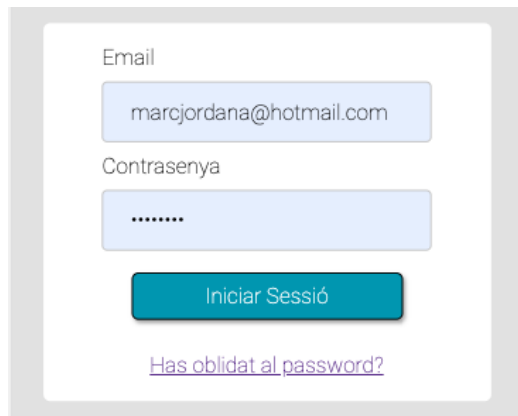
Aquesta imatge mostra un formulari web per iniciar sessió. A la part superior, hi ha un camp etiquetat "Email" amb el text "marcjordana@hotmail.com" introduït. Just a sota, hi ha un camp etiquetat "Contrasenya" amb sis punts "....." per ocultar el text. A continuació, hi ha un botó blau amb el text "Iniciar Sessió". Al final del formulari, hi ha un enllaç de text "Has oblidat al password?".

Fig. 4: Formulari per iniciar sessió

8.3 Panell principal

Una vegada un usuari registrat iniciï sessió, arribarà en la pantalla que es veu en la figura 5. Aquest panell ens permet iniciar totes les operatives que estan disponibles fer la nostra aplicació. Permet iniciar la creació d'un qüestionari, entrar en el panell de participació, entrar en el panell de qüestionaris, veure l'historial i finalment tancar la sessió. Encara que un usuari convidat no entri directament en aquesta pàgina, hi pot arribar, però com que no té permisos per fer moltes accions veurà totes les opcions deshabilitades, excepte la de participar en un qüestionari i la de tancar sessió.

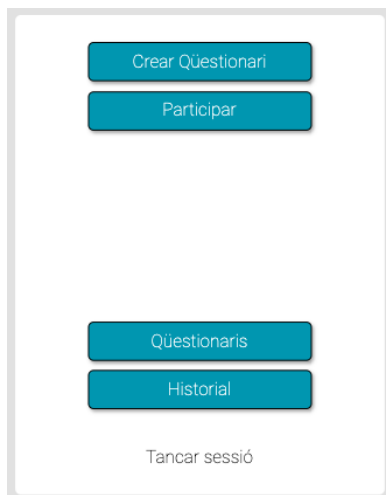
Aquesta imatge mostra el panell principal de l'aplicació. Conté un conjunt de botons blaus: "Crear Qüestionari" i "Participar" a la part superior; "Qüestionaris" i "Historial" a la part inferior; i "Tancar sessió" al centre de la part inferior.

Fig. 5: Panell principal

8.4 Creació de qüestionaris

L'usuari podrà crear els seus qüestionaris en el panell que podem veure en la figura 6. Per defecte un qüestionari té una pregunta, però en podrà escriure fins a cinc. L'usuari podrà posar un nom al qüestionari, per poder-lo identificar més fàcilment. Per poder escriure les preguntes, haurà de prémer el botó de la pregunta que vulgui editar i apareixerà un nou panell. En aquest panell hi podrà escriure la pregunta i afegir-hi respostes. Per defecte cada pregunta apareixerà amb dues respostes. Aquestes respostes tampoc es podran eliminar. L'usuari podrà afegir fins a 5 respostes a cada pregunta. Finalment, un cop s'hagi acabat d'escriure les preguntes i respostes, l'usuari podrà crear el qüestionari prement el botó de crear.

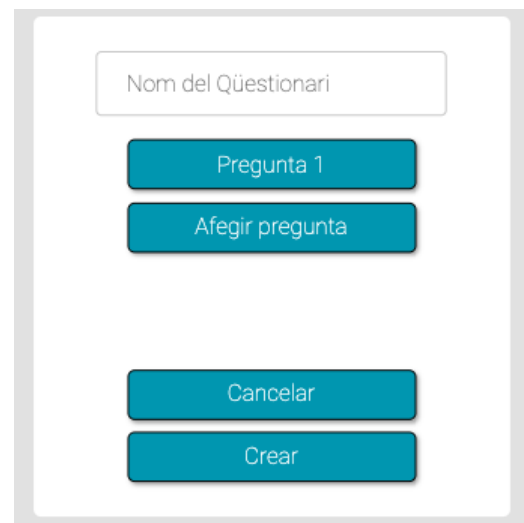
Aquesta imatge mostra el panell de creació de qüestionaris. A la part superior, hi ha un camp de text etiquetat "Nom del Qüestionari". A continuació, hi ha un botó "Pregunta 1" i un botó "Afegir pregunta". Al final, hi ha dos botons: "Cancelar" i "Crear".

Fig. 6: Creació de qüestionaris

8.5 Veure qüestionaris creats

A la pantalla que es pot observar en la figura 7, l'usuari podrà veure els qüestionaris que hagi creat i que estiguin en actiu. En el cas que vulgui començar a realitzar un qüestionari, haurà de prémer el botó del qüestionari que vulgui posar en funcionament.

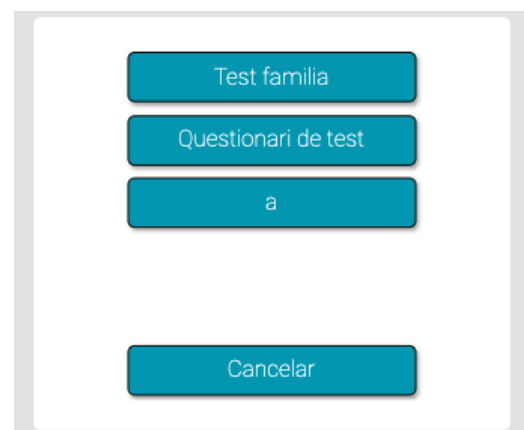
Aquesta imatge mostra el panell de qüestionaris creats. Conté un conjunt de botons blaus: "Test familia", "Questionari de test" i "a" a la part superior; i "Cancelar" a la part inferior.

Fig. 7: Panell de qüestionaris

8.6 Posar en funcionament un qüestionari

Quan un usuari vulgui posar en funcionament un qüestionari, com hem dit en el punt anterior, haurà d'estar en el menú de visualització de qüestionaris. Una vegada l'usuari premi el botó del qüestionari que vulgui començar, es trobarà amb una pantalla semblant a la figura 8. En aquesta pantalla podrà veure el títol del qüestionari i el codi d'inscripció.

Perquè els usuaris es puguin inscriure, s'haurà de prémer el botó d'inscripció d'usuari. Una vegada s'hagi premut, apareixerà un comptador, que recull el nombre d'usuaris que estan apuntats en el qüestionari en temps real.

Una vegada s'hagin inscrit tots els usuaris, es podrà posar en marxa el qüestionari. Quan es posi en funcionament, l'usuari veurà un panell en el qual hi haurà el nombre de respostes. Aquest nombre de resposta, anirà augmentant a mesura que els participants vagin responnent, ja que s'obtenen les dades en temps real.

Test familia

Codi del qüestionari 4IHQIzRMc5xS8I6t9XFW

Inscripció d'usuaris

Eliminar

Cancelar

Fig. 8: Pantalla per iniciar qüestionari

8.7 Participar en un qüestionari

En el panell de participació que podem veure en la figura 9, els usuaris es podran inscriure. Per fer-ho només és necessari el codi del qüestionari al qual es vulguin inscriure i un nom de participant que ha de ser únic.

En el cas que el nom de participant, el codi del qüestionari o que aquest qüestionari no estigui obert perquè s'inscriguin els participants, es mostrarà un missatge d'error.

D'altra banda si tot és correcte, s'arribarà a una pantalla d'espera fins que l'administrador del qüestionari l'iniciï. Una vegada iniciat, aniran apareixent les preguntes per pantalla amb les seves opcions de resposta corresponents. Per seleccionar una resposta, l'usuari haurà de prémer el botó de la resposta desitjada.

8.8 Restablir contrasenya

En el cas que un usuari no recordi la seva contrasenya, la podrà restablir. Com podem veure en la figura 10, necessitarà el correu electrònic. En el cas que el correu electrònic introduït sigui vàlid i estigui registrat a la nostra base de dades, es mostrarà un missatge de confirmació. En el cas contrari es mostrarà un missatge d'error.

Codi qüestionari

Nom participant

Participar

Cancelar

Fig. 9: Panell de participació

Correu electrònic

Confirmar

Cancel

Fig. 10: Restablir contrasenya

8.9 Veure l'historial de qüestionaris

Com es pot veure en la figura 11, en aquesta pantalla l'usuari podrà veure tots els seus qüestionaris que ja s'hagin respost. Si prem qualsevol dels botons d'aquests qüestionaris, entrarà en un altre menú on podrà veure les respostes que va obtenir.

Historial

Test familia

Questionari exemple

Test mobil

Questionari de test

Questionari de test 1

Test3

Cancelar

Fig. 11: Historial

9 DIFICULTATS

Durant el desenvolupament del projecte s'han trobat diverses dificultats. En aquesta secció les mencionarem i explicarem com s'han resolt.

La primera dificultat que es va trobar va ser, com ja hem dit en apartats anteriors, a l'hora d'executar el codi del servidor. Es va començar a desenvolupar el Back-End com s'havia fet en altres assignatures, pensant que penjant el codi al servidor i posant-lo en funcionament, aquest s'executaria, però Firebase no executa directament el codi del servidor, sinó que quan es fa el deploy de l'aplicació, busca el codi de les peticions i és el que realment es queda, ignorant tot el que quedi a fora o directament marcant-lo com error. Per aquesta raó es va haver de refer gran part del codi que s'havia escrit fins al moment.

Un altre problema al qual vam haver de fer front va ser en el tema de rebre dades en temps reals, ja que en aquest apartat no es tenia experiència prèvia i es va haver d'investigar més que en altres tasques. Des del començament del projecte, es va pensar a fer que el Back-End fos el que observés els canvis a la base de dades i els envies al Front-End mitjançant WebSockets, però pel motiu esmentat en el primer apartat de dificultats, no va poder ser així. La solució que es va trobar va ser directament rebre els canvis en el Front-End, ja que després de fer recerca per internet, es va veure que gran part de les solucions abordaven el problema d'aquesta manera.

Un dels errors que ens ha portat més temps de resoldre, és un problema de missatges d'error de Firebase. Hi ha errors que provoquen que aparegui un error de CORS encara que no sigui realment aquest problema. Es va perdre molt temps mirant de corregir aquest error, ja que en el codi del Back-End ja s'havia habilitat la política de Cross-Origin. Tot i això, ens seguia sortint el missatge d'error provocat per CORS. Finalment es va trobar un article [15] que explicava que l'error podia venir d'un altre problema, gràcies a això es va poder abordar el problema de manera diferent i solucionar-lo.

L'últim problema que es va haver de fer front és que quan es va fer una de les últimes pujades de codi al servidor van deixar de funcionar les peticions des de qualsevol dispositiu, exceptuant l'ordinador des d'on s'havia fet el projecte. Per solucionar-ho es van revisar els fitxers de configuració de Firebase, però no si va veure cap problema. També es va revisar la configuració de CORS, ja que des de certs navegadors ens marcava aquest error. Finalment es va veure que el problema venia del fet que les peticions del Front-End estaven apuntant a l'entorn de desenvolupament i no el de producció.

10 RESULTATS

Com hem pogut veure en els apartats anteriors, s'ha arribat als resultats esperats, ja que s'han fet totes les funcionalitats. Tot i això, en la part final del projecte es creia que no s'arribaria a poder desenvolupar-les totes. Entenem com a funcionalitats principals, la gestió de les sessions d'usuari, poder crear i eliminar qüestionaris, participar en qüestionaris i poder veure l'historial de qüestionaris realitzats.

Encara que com s'ha mencionat, s'han complert amb les funcionalitats principals, no s'ha pogut donar tota la im-

portància que mereix la interfície i l'experiència de l'usuari, ja que hi ha parts del projecte que ens han portat més temps del que s'havia calculat inicialment. Tot i això, el resultat és correcte, ja que l'aplicació web és funcional i intuïtiva.

11 FUTURES ACCIONS

Un dels canvis més importants, seria fer un codi de qüestionari més curt, ja que no és pràctic haver d'escriure un codi de vint dígit cada vegada que es vulgui participar en un qüestionari, a més a més la base de dades ja està preparada per emmagatzemar-lo. També seria important, que l'aplicació fos compatible amb més idiomes, com a mínim amb l'anglès i el castellà.

Com ja hem dit, un dels temes a millorar seria la interfície d'usuari i l'experiència que aquest rep en utilitzar l'aplicació. D'altra banda, també s'hauria de permetre reutilitzar els qüestionaris, ja que per falta de temps s'ha hagut de simplificar el funcionament i cada vegada que s'acaba un qüestionari aquest passa directament a l'historial i ja no es pot tornar a resoldre.

12 CONCLUSIONS

En aquest apartat s'explicaran les conclusions i reflexions que s'han obtingut a través de la realització d'aquest projecte.

L'objectiu d'aquest projecte era fer una aplicació web que permetés a l'usuari fer votacions en línia amb temps real. Encara que aquest objectiu s'hagi aconseguit, l'aplicació té molt marge de millora, ja que com hem dit en el punt anterior, l'experiència d'usuari es pot millorar força i actualment només és compatible amb un idioma.

La planificació que es va fer a l'inici del projecte al començament va ser correcte, només hi va haver una iteració en la qual no es van arribar als objectius. A mesura que ha anat avançant el projecte, la planificació ha sigut menys correcte, ja que s'han anat trobant problemes que han fet endarrerir la resta de tasques. Aquest projecte m'ha servit per veure la dificultat de planificar correctament el desenvolupament d'un projecte real, ja que hi ha molts factors que en un primer moment no es tenen en compte i que realment influeixen moltíssim en els temps d'entrega de funcionalitats. Les dependències entre funcionalitats tampoc es van tenir massa en compte, ja que hi ha hagut dos casos: la participació de qüestionaris i l'inici de sessió d'usuaris, que per dependències amb altres funcionalitats, han provocat que s'hagi hagut de modificar la planificació.

Fent aquest projecte he après que primer de tot s'ha de mirar com funcionen les interfícies que s'utilitzaran en el projecte i no donar per fet que aquestes funcionaran com altres en les que s'hagi treballat prèviament. En aquest treball s'ha vist que pot ser realment difícil compaginar el temps entre la feina, les diferents assignatures de la universitat i el projecte de final de grau.

Pel que fa a la part de desenvolupament crec que el resultat final del projecte ha estat bo, ja que s'han implementat totes les funcionalitats essencials.

Finalment, m'hagués agradat disposar de més temps per fer el projecte, ja que he hagut de deixar aspectes que m'hagués agradat implementar.

AGRAÏMENTS

Gràcies a Aaron Blanco, que m'ha tutoritzat el projecte i m'ha ajudat a resoldre dubtes. Sense els seus consells s'hauria tardat més temps a fer el projecte. Gràcies a l'Esther Fraile per haver fet el logotip de l'aplicació i finalment agrair el suport de la meua família, especialment el de la meua parella, al llarg de tota la carrera.

ÍNDEX DE FIGURES

1	Panell d'administració d'usuaris	3
2	Model relacional BD	3
3	Formulari de registre	4
4	Formulari per iniciar sessió	5
5	Panell principal	5
6	Creació de qüestionaris	5
7	Panell de qüestionaris	5
8	Pantalla per iniciar qüestionari	6
9	Panell de participació	6
10	Restablir contrasenya	6
11	Historial	6
12	Diagrama de Gantt	8

ÍNDEX DE TAULES

1	Planificació del projecte	3
---	-------------------------------------	---

REFERÈNCIES

- [1] ReactJS, és una llibreria per construir interfícies d'usuari amb JavaScript. 04/03/2020. Disponible: <https://reactjs.org/docs/getting-started.html>
- [2] NodeJS, és un entorn d'execució per JavaScript. 04/03/2020. Disponible: <https://nodejs.org/es/>
- [3] Firebase, és una plataforma d'allotjament web. 22/02/2020. Disponible: <https://firebase.google.com/docs>
- [4] Kahoot, és una aplicació per realitzar qüestionaris en línia. Disponible: <https://kahoot.com>
- [5] Doodle, és una web per programar reunions. Disponible: <https://doodle.com/es/>
- [6] Gitlab, és una eina per fer el control de versions. Disponible: <https://about.gitlab.com>
- [7] Realtime Database, és un tipus de bases de dades que ofereix Firebase. Disponible: <https://firebase.google.com/docs/database>
- [8] Cloud Firestore, és un tipus de bases de dades que ofereix Firebase. Disponible: <https://firebase.google.com/docs/firestore>
- [9] React Router, és un component de ReactJS. Disponible: <https://reacttraining.com/react-router/web/guides/quick-start>
- [10] Component de Firebase de ReactJS, és un component que ens permet connectar ReactJS amb Firbase. Disponible: <https://www.npmjs.com/package/firebase>

- [11] WebSockets, és una tecnologia que fa possible obrir una sessió de comunicació entre el navegador i el servidor. Disponible: https://developer.mozilla.org/es/docs/Web/API/WebSockets_API
- [12] React Redux, és un contenidor d'estats per aplicacions amb JavaScript. Disponible: <https://redux.js.org/introduction/getting-started>
- [13] Express, és un framework minimalista per NodeJS. Disponible: <http://expressjs.com>
- [14] Cross-Origin Resoure Sharing, és un mecanisme que utilitza les capçaleres HTTP per dir als navegadors l'origen de l'aplicació web. Disponible: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- [15] Article, aquest article explica si el servidor et retorna l'error de CORS, pot ser per altres errors. Disponible: <https://haha.world/firebase-cors/>

APÈNDIX

A.1 Diagrama de Gantt

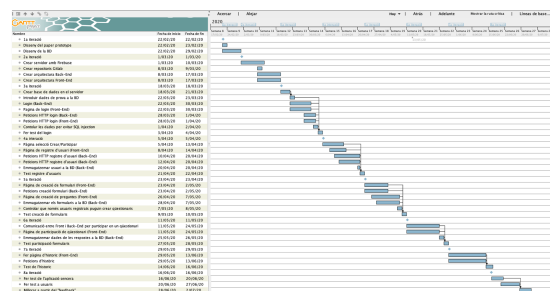


Fig. 12: Diagrama de Gantt

A.2 Enllaç a l'aplicació

<https://appquest-dcb91.firebaseio.com>