

Treball Final de Grau
Desenvolupament d'Aplicacions per Dispositius Mòbils

MEMÒRIA



Alumne: Daniel Domínguez Bender
Pla d'estudis: Grau Enginyeria Informàtica
Àrea de treball final: Desenvolupament d'Aplicacions per Dispositius Mòbils

Consultor: Carles Sànchez Rosa

Data lliurament: 04.06.2021

Índex

Capítol 1 - El pla de treball	3
1.1 Context i justificació del Treball	3
1.2 Objectius del Treball.....	4
1.3 Enfocament i mètode seguit	4
1.4 Planificació del treball.....	4
Capítol 2 - Investigació	7
Justificació de la tria del mètode d'indagació:.....	7
Procés plantejament:	7
Desenvolupament:	7
Fitxa perfils usuaris:.....	8
Prototipatge	10
Capítol 3 - Desenvolupament.....	13
3.1 - Eines, Editors, APIs.....	13
Què és Ionic?	13
Què es Github?.....	14
Què es Firebase?	15
Què es Stripe?	15
Què és Heroku?	15
3.2 - Preparar aplicació per ferla servir en un entorn de treball.....	20
Capítol 4 - Anàlisis aplicació.....	21
4.1 - Funcionalitats.....	21
4.2 - Explicació del codi	36
Capítol 5 - Proves aplicació.....	39
Capítol 6 - Problemes	42
Capítol 7 - Conclusions.....	50
Capítol 8 - Agraïments	51
Capítol 9 - Bibliografia	52

Capítol 1 - El pla de treball

1.1 Context i justificació del Treball

En aquests temps en que el món ha vist com han canviat moltes coses i d'altres s'han hagut d'adaptar degut a l'aparició del COVID-19, moltes persones han optat per sortir de la seva zona de confort degut al confinament i han decidit començar a fer esport.

Algunes persones han optat per sortir a passejar o a córrer i d'altres a començar a anar en bicicleta. És en aquest col·lectiu al qual ens enfocarem en aquest TFG.

Per a persones que no han practicat gaire aquest esport i d'altres que ho volguessin practicar en un altre lloc que no sigui al caos de les grans (i per què no també petites) ciutats, se'ls vol oferir l'oportunitat de poder re-descobrir el món del ciclisme.

Una família que vulgui passar uns dies fora i gaudir de la naturalesa fent excursions o rutes en bicicleta, s'enfronta moltes vegades amb la situació de que no tenen bicicletes per a tots el membres familiars, no tenen la capacitat de transportar tant equip en el seu cotxe per falta d'espai o ambdues coses a la vegada.

La necessitat a cobrir aquí seria la possibilitat d'oferir estances per a poder retornar després de gaudir d'hores damunt d'una bicicleta, la possibilitat de llogar bicicletes per a tots el membres familiars en cas que no es tingui o no es pugui transportar en cotxe al lloc de destí, la possibilitat de fer una ruta en família i gaudir de la naturalesa i trencar d'aquesta manera amb la monotonia d'estar a la ciutat.

Si que és cert que existeixen apps que s'adrecin a parts que s'han comentat amb anterioritat, com el tema estances, en aquí podríem citar algunes conegudes com Booking o Airbnb. Una altre app no tan coneguda que ofereix allotjament seria Warmshowers, persones que ofereixen la seva casa o fins i tot el seu jardí a cicli-turistes. Però en cas d'anar en família, igual podria ser una opció poc idònia per falta d'espai a l'allotjament.

Aquestes apps ofereixen només l'estança, i no la possibilitat d'alquilar bicicletes ni d'oferir una ruta per a poder passejar en bicicleta.

Aplicacions existents que ofereixin o mostrin rutes serien per exemple Biki Citizens, que calcula rutes en temps real i ofereix mapes de calor de diferents ciutats.

L'aplicació a desenvolupar és per a l'empresa GPS (Girona Pro Sport). Una empresa dedicada a oferir serveis tant a ciclistes o equips ciclistes professionals i amateurs com a entorns no professionals com nuclis familiars, parelles i/o *singles*. Els serveis son oferits en forma de «packs» que inclouen estances, circuits / rutes d'entrenament, visites / tours per Girona, informació de locals per a ciclistes, lloguer de material i la compra de roba esportiva per a ciclistes. La idea és que l'aplicació sigui capaç de realitzar tot això que es pot fer a través de la web.

1.2 Objectius del Treball

L'objectiu general d'aquest projecte, és crear una app multi plataforma per a Smartphones (Android & iOS). Els objectius específics que es volen assolir per a l'aplicació mòbil són:

- Adquirir *packs* / serveis
- Veure activitats / rutes
- Compra de material

Encara que GPS és una empresa que ofereix els serveis a Girona, el seu mercat està dirigit a països d'Europa Central com Eslovàquia. Volem aprofitar per fer una versió Catalana de l'aplicació. El *handycap* aquí és traduir els textos existents de l'Eslovac al Català / Castellà.

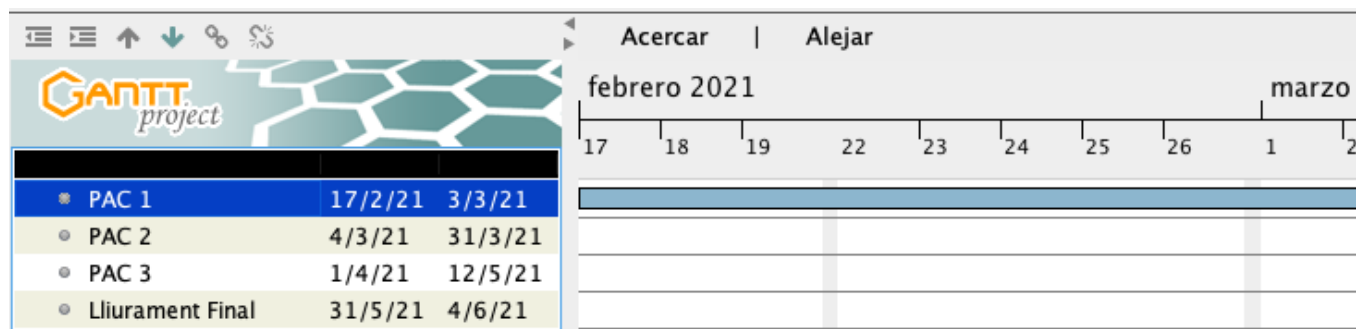
1.3 Enfocament i mètode seguit

Com ja ha sigut mencionat en un apartat anterior, aquí s'ha decidit de crear una app des de zero. Degut als requeriments que necessita Girona Pro Sport, creiem que la millor solució és crear una nova aplicació en comptes d'agafar una existent i adaptar-la.

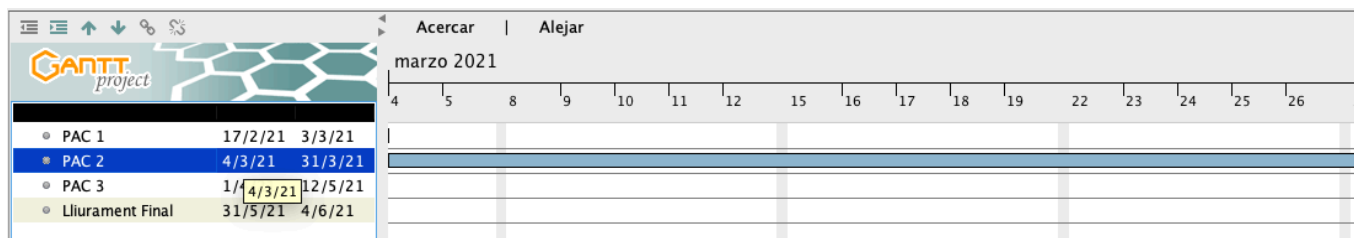
1.4 Planificació del treball

Realitzarem el projecte en 4 fases. Per poder treballar de manera més àgil i efectiva la idea és fer entregues parcials del producte a desenvolupar cada dues setmanes. Aquestes entregues parcials començaran després d'un període inicial de 15 dies (a mode de training camp) durant la fase 2 del projecte. Aquests dies d'autoaprenentatge els farem servir per aprendre a servir Ionic & Angular, que serà lo que farem servir per el desenvolupament de l'aplicació. També aprofitarem per deixar llest l'entorn per a desenvolupar l'aplicació.

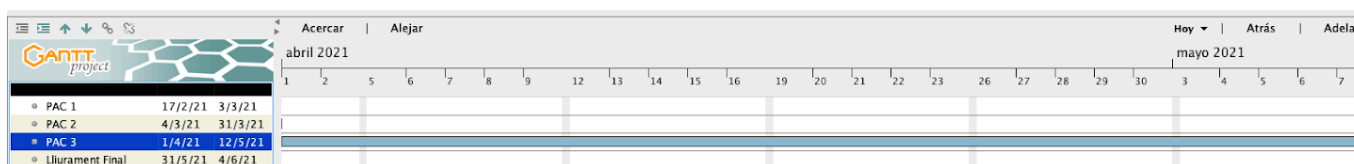
Les 4 fases principals seran les dades que estan preestablertes per a les diferents Pacs, però dintre de cada fase principal en tindrem de subfases per dedicar un temps a una certa cosa concreta.



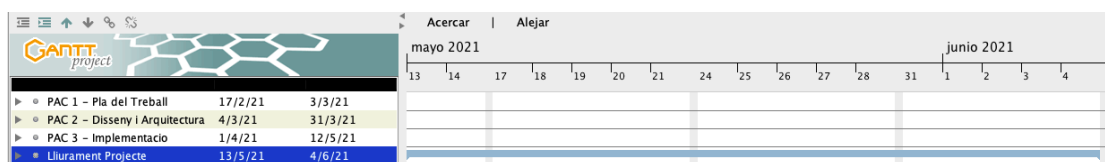
La PAC 1 és bàsicament cerca informació per al TFG.



A la PAC 2 dedicariem temps a instal·lar l'entorn i a l'aprenentatge de Ionic & Angular. L'objectiu en aquesta fase seria tenir llest l'entorn i poder haver fet algunes proves amb Ionic. En aquesta fase també començarem a tenir reunions setmanals amb l'empresa GPS per confirmar que tot va com s'ha plantejat inicialment. La comunicació amb el nostre tutor també serà habitual a partir d'aquesta fase.



La PAC 3 conté la implementació del projecte. En aquí serà sovint el contacte entre nosaltres i l'empresa GPS per mostrar l'aplicació i fer els canvis que es demanin. L'objectiu final aquí és tenir llesta l'aplicació.



El lliurament final hem decidit deixar per acabat el projecte per el dia 12 de Maig, de manera que farem servir els dies previs a l'entrega (4 de Juny) per adaptar la Memòria a entregar. Objectiu aquí és tenir llesta la Memòria, el Producte i la Presentació.

Degut al temps que s'ha de dedicar al projecte i tenint en compte les condicions a les quals està sotmès l'autor del projecte (feina a jornada completa 39h/setmana), creiem que unes condicions raonables per assolir l'objectiu és poder dedicar 2 hores diàries (de mitjana) i algunes hores extra els cap de setmana al TFG. La intenció aquí és dedicar 20 hores setmanals. En alguns casos es podria incrementar temporalment la dedicació per poder assolir l'objectiu final.

Si fem càlculs dona a 20 hores la setmana i calculant que des del dia 17 Febrer al 4 de Juny son unes 15 setmanes, fa un total de 300 hores. Tenint en compte que el TFG són 12 crèdits, creiem que és un volum d'hores factible per a dedicar-li a un projecte d'aquesta envergadura. A continuació el desglossament de la dedicació al treball segons les fases (pot canviar de cara a l'entrega final).

▼ • PAC 1 – Pla del Treball	17/2/21	3/3/21
• Cerca informació	17/2/21	3/3/21
▼ • PAC 2 – Disseny i Arquitectura	4/3/21	31/3/21
• Instal·lació entorn	4/3/21	8/3/21
• Training Camp	8/3/21	22/3/21
• Disseny	22/3/21	31/3/21
▼ • PAC 3 – Implementació	1/4/21	12/5/21
• Desenvolupament	1/4/21	5/5/21
• Testing	5/5/21	12/5/21
▼ • Lliurament Projecte	13/5/21	4/6/21
• Producte	13/5/21	19/5/21
• Presentació	19/5/21	25/5/21
• Memòria	25/5/21	4/6/21

Segons el que hem calculat, el que més temps ens portarà en quant a dedicació és la fase d'aprenentatge (Training Camp) i la fase de Desenvolupament del producte.

La part del lliurament final volem deixar enllestit el producte per tenir temps de dedicar-li a la Memòria i la Presentació.

Capítol 2 - Investigació

L'aplicació a desenvolupar es basa en la versió WEB de l'empresa GironaProSport. La idea es desenvolupar la versió app, en la qual es puguin realitzar les mateixes accions (o el més fidel possible) que a la versió WEB. Per a poder realitzar això, haurem de fer prèviament un estudi i triar un mètode de indagació apropiat per al desenvolupament de l'aplicació.

Justificació de la tria del mètode d'indagació:

Com el que volem desenvolupar és una aplicació mòbil, lo que equival a que els usuaris no estaran parats en el mateix lloc, per exemple el seu escriptori de treball davant del seu ordinador, sinó que poden estar en qualsevol lloc de casa seva o fora d'ella, creiem que el *Shadowing* seria el mètode de indagació més adient en aquest cas. Aquest mètode ens pot ajudar millor a conèixer:

- les característiques dels usuaris
- les seves necessitats
- els seus objectius
- el context d'ús
-

El *Shadowing* és una tècnica d'observació en el qual s'anoten les rutines del subjecte a observar (podeu llegir una mica més sobre aquest mètode en el següent [link](#)). En ser una tècnica intrusiva, intentarem no intervenir gaire amb els usuaris de manera que no es vegi molt afectada la seva rutina diària.

Procés plantejament:

Com que el que volem desenvolupar es una aplicació mòbil, el que farem serà acompanyar als usuaris durant les seves activitats diàries anotant les seves rutines. Aquest procés de *Shadowing* el durarem a terme durant una setmana sencera, de dilluns a diumenge, perquè ens interessa saber també els hàbits dels usuaris durant el seu temps lliure. De manera que veurem quines costums tenen els usuaris a l'hora de fer servir el seu *Smartphone* i consultar o fer servir diverses aplicacions. Quan i com fan servir els seus dispositius? Quanta estona dediquen a mirar coses als seus dispositius? On ho fan? Aprofiten una parada a un semàfor o uns minuts de la feina per mirar el mòbil?

Son informacions que ens agradaria poder esbrinar per poder desenvolupar més tard la nostra aplicació.

Desenvolupament:

Acompanyarem a un parell d'usuaris durant tota una setmana per veure els seus costums en fer servir el seu dispositiu mòbil. Creiem que això ens ajudarà a desenvolupar millor molts aspectes de l'aplicació a desenvolupar, com podria ser per exemple donar més importància a l'aspecte vistós de manera que l'usuari dediqui una mica més de temps en fer servir l'aplicació en cas de fer-la servir quan no està assegut o un temps de càrrega millor per aprofitar el temps que l'usuari està parat esperant a l'autobús. Els usuaris que hem acompanyat son de perfils ben diferents ja que un és molt més actiu en quant es refereix a fer servir el seu dispositiu mòbil mentre que l'altre subjecte està més enfocat en la seva feina i fa servir el dispositiu mòbil més durant els descansos i temps lliures.

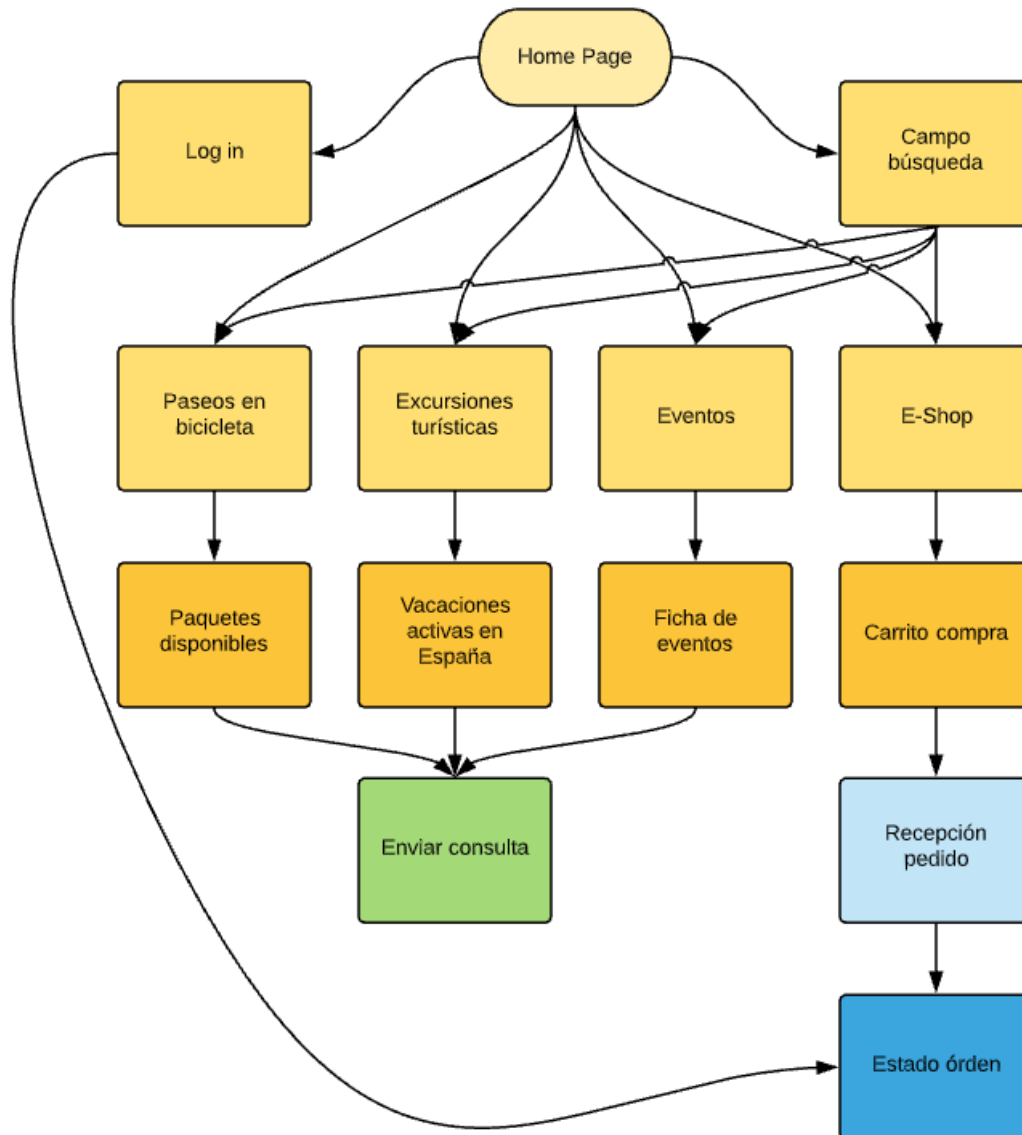
Donem pas a un breu resum de cadascun dels usuaris.

Fitxa perfils usuaris:

Fitxa usuari: Jana Kunecova		Context d'ús	
Perfil		<p>Jana farà ús del seu propi <i>smartphone</i> més en els moments que tingui temps a parer a dedicar-li. Això vol dir a casa després de sopar o al llit abans d'anar a dormir. El <i>smartphone</i> empresarial el farà servir o a casa (<i>remote working</i>), quan treballa al <i>Co-Working</i> o l'únic dia de la setmana que ha d'anar a l'oficina. Son llocs ben il·luminats i amb connexió WIFI.</p>	
Edat	38 anys		
Professió	E-Commerce		
Ubicació	Girona		
Interessos	Esport i naturalesa		
Motivacions	Viatjar Activitats familiars		
Experiència ús tecnologies	Nivell usuari	Anàlisis de tasques	
<p>Un temps de càrrega alt. Una interfície fàcil de fer servir.</p>		<p>Característiques i/o elements de la interfície En no tenir gaire temps per a dedicar-li en fer servir <i>apps</i> mòbils, volem enfocar-nos en fer una aplicació lleugera amb temps de càrrega curta per a que usuaris com Jana puguin fer servir l'<i>app</i> durant els pocs moments lliures dels qual disposen.</p>	

Fitxa usuari: Patrick Domínguez		Context d'ús	
Perfil		<p>Patrick fa ús del mòbil a quasi qualsevol lloc i a qualsevol hora. Com es passa gran part del temps durant el dia en transport públic, l'ús que farà al mòbil serà mentre es mou. Degut a la situació de COVID, treballa 2 setmanes al mes des de casa i les altres 2 a l'oficina. Son llocs molt ben il·luminats i amb connexions WIFI.</p>	
Edat	36 anys		
Professió	Webmaster i SEO		
Ubicació	Castelldefels Viatjar i últimes tecnologies		
Interessos	Activitats esportives		
Motivacions	Conèixer gent nova		
Experiència ús tecnologies	Usuari avançat	Anàlisis de tasques	
<p>Una bona experiència d'usuari. Que el UX estigui ben pensat.</p>		<p>Característiques i/o elements de la interfície En passar-se'l dia davant la pantalla de l'ordinador o la pantalla del seu <i>smartphone</i>, no dedica gaire temps a pàgines web en concret. Dedicar més temps a coses que li entren pels ulls. Això ens ha fet pensar en donar a l'aplicació un disseny atractiu, una presentació que inviti a l'usuari a voler dedicar més temps en fer servir l'<i>app</i>.</p>	

A continuació es mostrarà el flux d'interacció que tindrà l'aplicació.



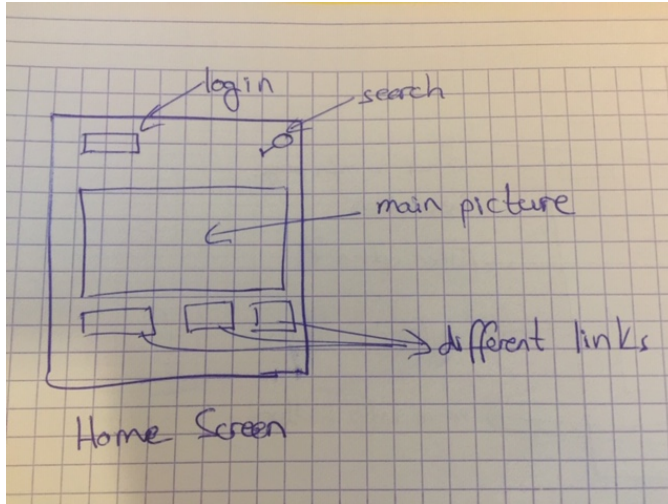
S'observa que l'aplicació disposarà d'una HOME PAGE des de la qual s'accedeixen a les demés funcionalitats. La pàgina de LOGIN, una secció de SEARCH i les 4 seccions principals:

- Paseos en bicicleta
- Escursiones
- Eventos
- E-Shop

Per poder veure l'estat d'una comanda s'ha d'estar registrat prèviament.

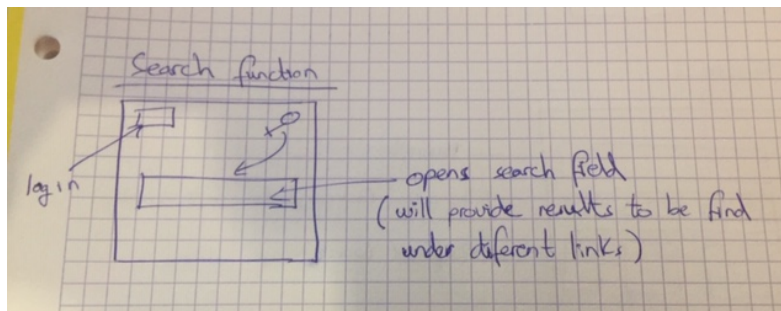
Prototipatge

A continuació es presenten els primers Sketches realitzats a mà alçada. La pàgina principal seria més o menys així:

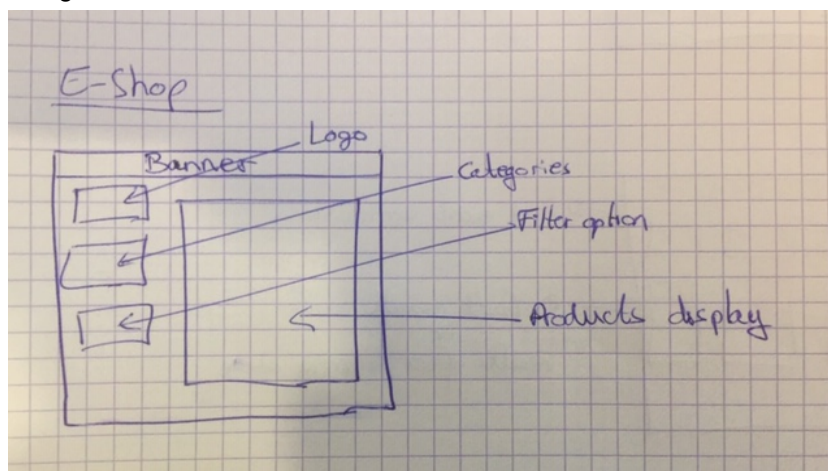


Des de la pàgina principal es podrien accedir a les diferents parts de l'aplicació, com poden ser l'apartat d'Events, dels Tours ofertos o de la pàgina de compra de material esportiu.

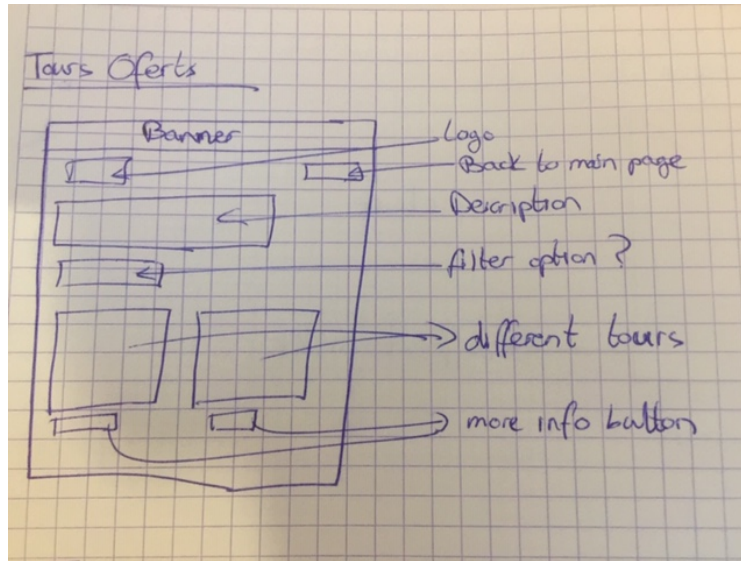
La funcionalitat de cerca es vol que sigui global, que mostri els resultats de la paraula cerca si es troba en tots els apartats.



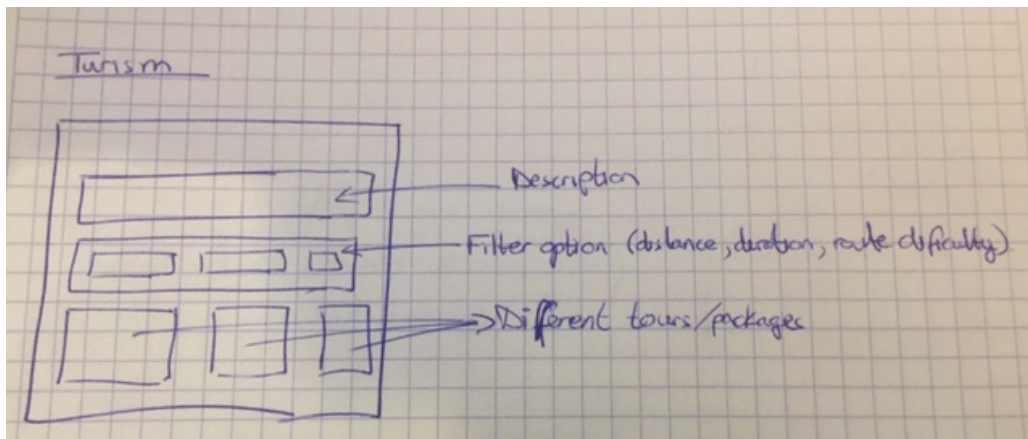
L'apartat de compra de material esportiu podria ser més o menys així. Tenint l'opció de filtrar el material categories, colors, ...



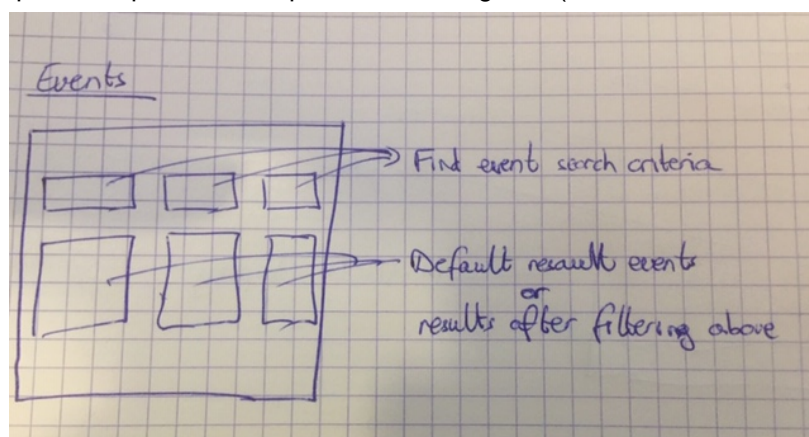
L'apartat dels tours. Començar amb una breu descripció i després oferir l'opció de filtrar, per exemple, dificultat de la ruta.



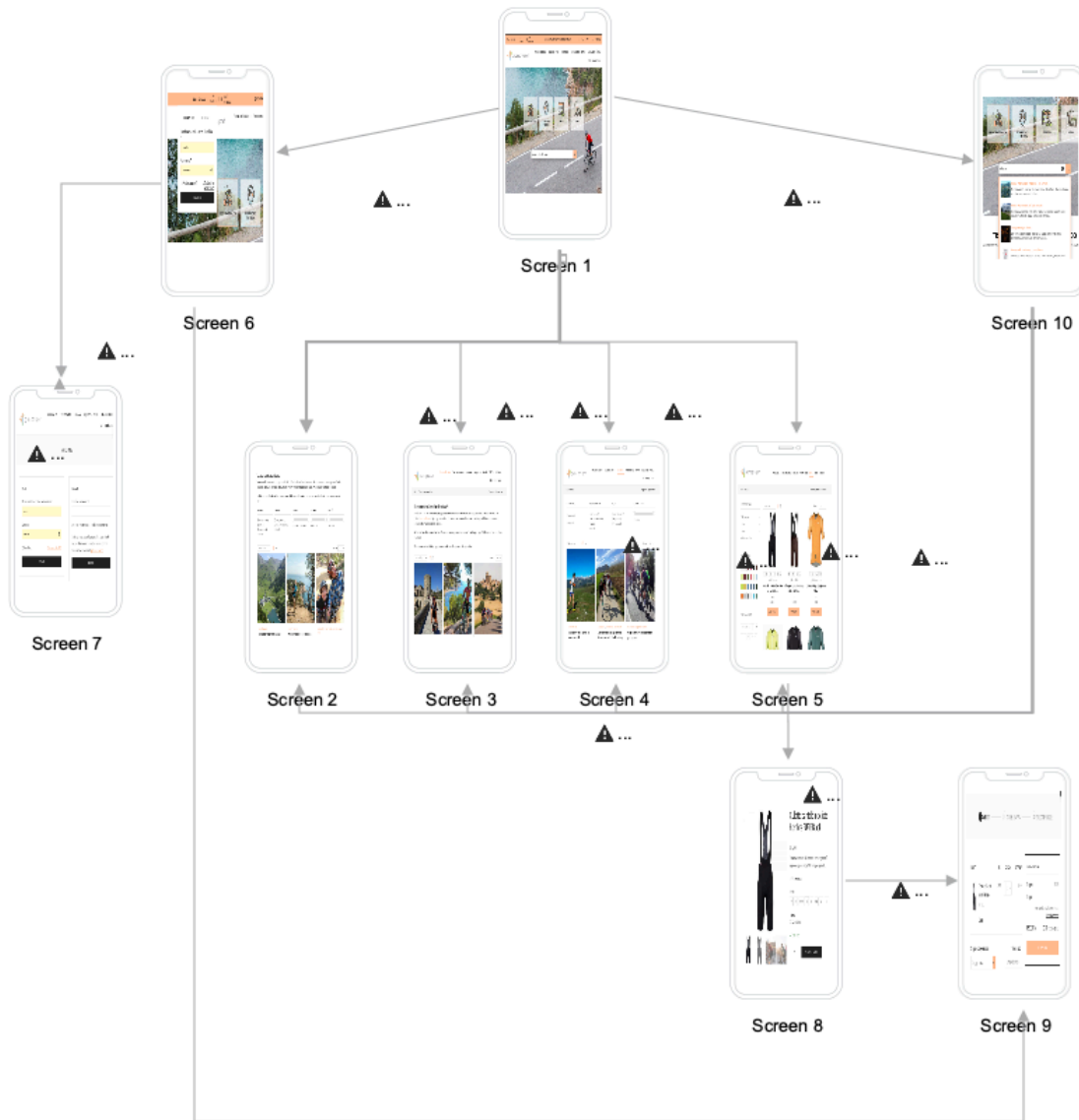
L'apartat de turisme, semblant a la pantalla anterior. Una descripció i l'opció de filtrar.



Els events esportius. Opció de filtrar per dates o categories (ciclisme, triatló, training camp, ...).



El prototipatge horitzontal tindria la següent forma.



Al prototipatge es pot observar les diferents pantalles a les quals es pot accedir.

La principal (Screen 1) es la Home Page. Des d'aquesta es pot accedir a la de Login (Screen 6) la de Cerca (Screen 10), Paseos en bicicleta (Screen 2) Excursiones Turísticas (Screen 3), Eventos (Screen 4), E-Shop (Screen 5).

Des del Login (Screen 6) es pot accedir a la de Registre (Screen 7). Per veure l'estat d'una comanda (Screen 9) es obligatori estar registrat.

En fer una Cerca (Screen 10) es pot retornar la paraula cercada i accedir al diferents llocs de la web, en l'exemple s'ha cercat Girona i s'han obtingut resultats de Tours com també d'una camiseta (E-Shop) amb la paraula clau Girona.

Capítol 3 - Desenvolupament

3.1 - Eines, Editors, APIs

Dispositiu empleat per el desenvolupament del projecte:

- MacBook Pro Late 2011
- OSX 10.15.16 (MacOS Catalina)

Frameworks i aplicacions:

- NodeJS 4.16.0 LTS
- Visual Studio Code
- Xcode (ios) / Android Studio (android)
- Ionic 5.4.16

```
[danieldominguez@MacBookPro bikeTours-api % ionic -v  
5.4.16
```

- Firebase
- Heroku
- Stripe
- Github

A continuació explicarem una mica alguns dels Frameworks i eines que hem utilitzat al llarg d'aquest projecte de final de grau.

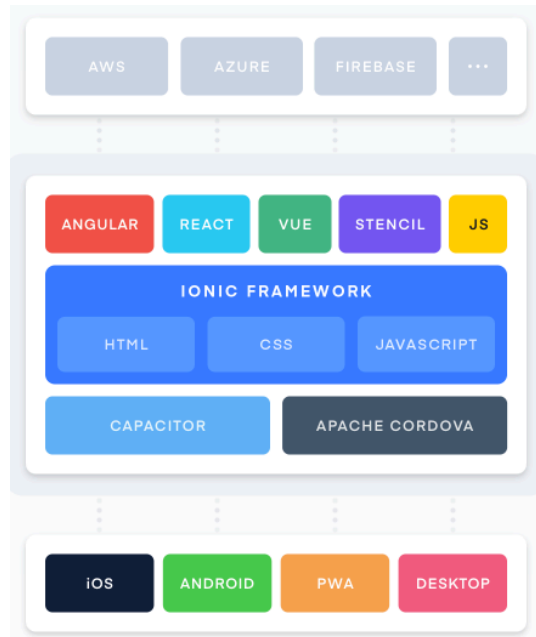
Començarem per donar una mica d'informació sobre Ionic i després comentarem una mica què són Github, Firebase i Stripe.

Què és Ionic?

És un SDK de codi obert que facilita la creació d'aplicacions web mòbils i progressives (PWA) d'alt rendiment per a qualsevol plataforma i dispositiu. En altres paraules, es utilitza per al desenvolupament d'aplicacions mòbils híbrides.



Consta d'una rica biblioteca de blocs de construcció i components d'interfície d'usuari que faciliten el disseny de l'aplicació.



Imatge extreta de ionicframework.com

Ionic utilitza Cordova i més recentment Capacitor plugins per dotar a sistemes operatius host amb característiques com Càmera, GPS, ... A més de l'SDK, Ionic també proporciona serveis que els desenvolupadors poden utilitzar per habilitar funcions, com ara desplegaments de codi, compilacions automatitzades. Ionic també proporciona el seu propi IDE conegut com Ionic Studio (encara que nosaltres hem fet servir Visual Studio Code). Ionic també proporciona una interfície de línia d'ordres (CLI) per crear projectes. La CLI també permet als desenvolupadors afegir connectors de Cordova i paquets front-end addicionals, habilitar notificacions push, generar icones d'aplicacions i pantalles Splash, crear binaris natius...

Què es Github?

Viquipèdia dona la següent descripció:

Es tracta d'un servei de *hosting* de repositoris Git, el qual ofereix tota la funcionalitat de Git de control de revisió distribuït i administració de codi de la font (SCM) així com afegint les seves característiques pròpies. A diferència de Git, el qual és estrictament una eina de línia d'ordres, GitHub proporciona una interfície gràfica basada en web i escriptori així com integració del mòbil. També proporciona control d'accés i diverses característiques de col·laboració com bug tracking, administració de tasques, i wikis per cada projecte.

Els projectes a GitHub poden ser accedits i manipulats utilitzant l'estàndard d'interfície de consola d'ordres Git i totes les ordres estàndard Git funcionen. GitHub també permet navegar als usuaris no registrats pels repositoris públics, donant la possibilitat de descarregar codi i/o projectes d'altres usuaris per el desenvolupament de codi. En quant a les seves llicències, les condicions d'ús de GitHub no requereixen que projectes de programari públic hosted a GitHub coneguin la Definició de Codi obert. En qualsevol cas és aconsellable per usuaris i desenvolupadors que pretenen utilitzar una peça del programari trobat en GitHub que llegeixin la llicència de programari en el repositori (normalment trobat en un arxiu situat entre els primers anomenat "LICENSE.txt" o similar) per determinar si accepta les condicions.

Què es Firebase?

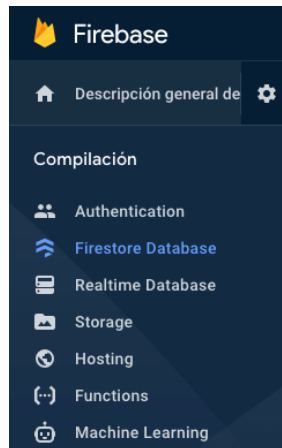


És una plataforma per al desenvolupament de aplicacions web i aplicacions mòbils llançada el 2011 i adquirida per Google tres anys més tard: Ubicada al núvol (integrada amb Google Cloud Platform), que utilitza un conjunt d'eines per a la creació i sincronització de projectes que seran dotats d'alta qualitat.

Els avantatges en utilitzar aquest entorn serien:

- Sincronitzar fàcilment les dades dels projectes sense haver de administrar connexions o escriure lògica de sincronització complexa.
- Utilitzar un conjunt de eines multi plataforma: fàcil integració per a plataformes web com en aplicacions mòbils, compatible amb grans plataformes com IOS, Android i aplicacions web.
- Fer servir infraestructura de Google.
- Crear projectes sense necessitat d'un servidor ja que les eines s'inclouen en els SDK per als dispositius mòbils i web.

Com a serveis incorpora Firebase Analytics, Firebase Cloud Messaging, Firebase Auth, Real Time DB, Firebase Storage ...



Què es Stripe?



Stripe és una companyia tecnològica el qual el seu programari permet a individus i negocis rebre pagaments per internet. Proporciona la infraestructura tècnica, de prevenció de frau i bancària necessària per operar sistemes de pagament en línia. És una eina molt útil que la hem integrat a la nostra aplicació per tenir un *overview* de quines comandes han realitzat segons quins usuaris.

Què és Heroku?



És una PAAS de computació al núvol que suporta diferents llenguatges de programació com son Java, node.js, Python, Php ...

Per al desenvolupament de l'aplicació també hem fet servir APIs que hem creat a la web de [Heroku](#).

Les APIs que hem creat han sigut:

Biketours: <https://bike-tours-api.herokuapp.com/bikeTours>

Tourism: <https://tourism-api-project.herokuapp.com/tourism>

Events: <https://event-api-project.herokuapp.com/event>

El motiu per el qual hem optat per fer servir APIs per les diferents activitats es per no carregar el nostre codi amb línies addicionals. Fent que l'app mostri les dades que necessita en connectar-se a una de les APIs facilita molt codificar certes parts. Inicialment no havíem pensat fer servir APIs, per hem vist com es reduïa molt les línies de codi en segons quins apartats.

Per exemple, per mostrar a la pantalla inicial els tours en bicicleta, el turisme i els events el codi inicialment quedava de la següent forma:

```

<ion-img src="../../assets/thumbnail/homeScreen/img_1.jpg"/></ion-img>
<ion-card-header>
<ion-card-subtitle>Road cycling bike, tours packages, bike tours Girona</ion-card-subtitle>
<ion-card-title>Girona bike rides - STANDARD package</ion-card-title>
<ion-card-subtitle>price from 516.00 €</ion-card-subtitle>
<p>Bike ride STANDARD - The optimal package for enjoying cycling in Girona using PRO ...</p>
<ion-item color="primary">GUIDE AVAILABLE: Yes</ion-item>
<ion-item color="primary">COACH AVAILABLE: No</ion-item>
</ion-card-header>
</ion-card>
<ion-card color="primary" button>
<ion-img src="../../assets/thumbnail/homeScreen/img_2.jpg"/></ion-img>
<ion-card-header>
<ion-card-subtitle>Cycling, mountain bike / gravel / cyclocross, with children</ion-card-subtitle>
<ion-card-title>Costa Brava + Griona with childrne 3 days</ion-card-title>
<ion-card-subtitle>price from 545.00 €</ion-card-subtitle>
<p>A short family trip suitable for children from 10 years, enjoy cycling without traffic with the whole ...</p>
<ion-item color="primary">DISTANCE: 140 KM</ion-item>
<ion-item color="primary">DIFFICULTY: Easy</ion-item>
</ion-card-header>
</ion-card>

```

Per mostrar 2 imatges amb les seves informacions hem necessitat quasi 20 línies de codi.

Mirem ara el mateix (o quasi el mateix) fent servir una API:

```

<ion-card-header>
<ion-card-title>Bike Tours</ion-card-title>
</ion-card-header>
<!-- make sure to have created before the biketoursList on TAB1.page.ts file -->
<ion-card [routerLink]="['biketours-detail', biketours.id]" button *ngFor="let biketours of (biketoursList | async)">
<ion-img [src]="biketours.imageUrl"></ion-img>

```



```
<ion-card-header>
<ion-card-subtitle>{{biketours.title}}</ion-card-subtitle>
</ion-card-header>
```

Gràcies a la API i amb la comanda `*ngFOR` podem cridar de manera recurrent les dades que volem mostrar per pantalla. Hem fet servir la meitat de codi i podem mostrar els 4 BikeTours que tenim a l'api, mentre que sense fer servir una API hauríem necessitat (a ull) unes 40 línies.

Si que es cert que inicialment mostràvem més dades a la pantalla inicial, com son els preus i la disponibilitat de guia o de coach (això es mostra ara als detalls de cada activitat) però encara així s'ha aconseguit reduir la quantitat de codi.

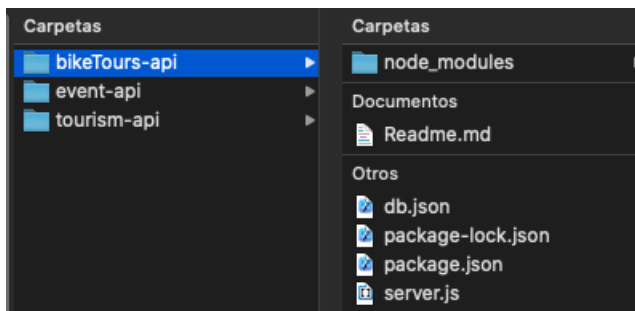
Les APIs que hem fet servir les podem recuperar dels següent repositoris creats a Github:

<https://github.com/dadobe/bikeTours-api>

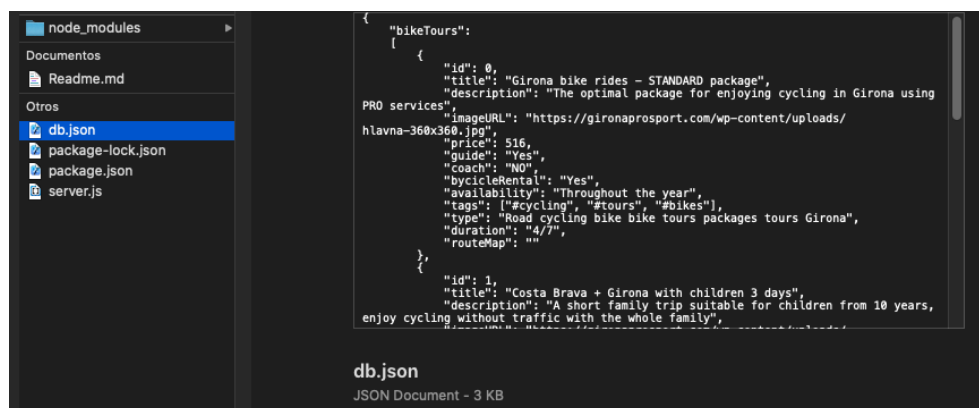
<https://github.com/dadobe/tourism-api>

<https://github.com/dadobe/event-api>

El procediment que hem seguit per a la creació de les APIs es descriurà a continuació:
Per a cada API hem creat carpetes individuals amb el mateix contingut.



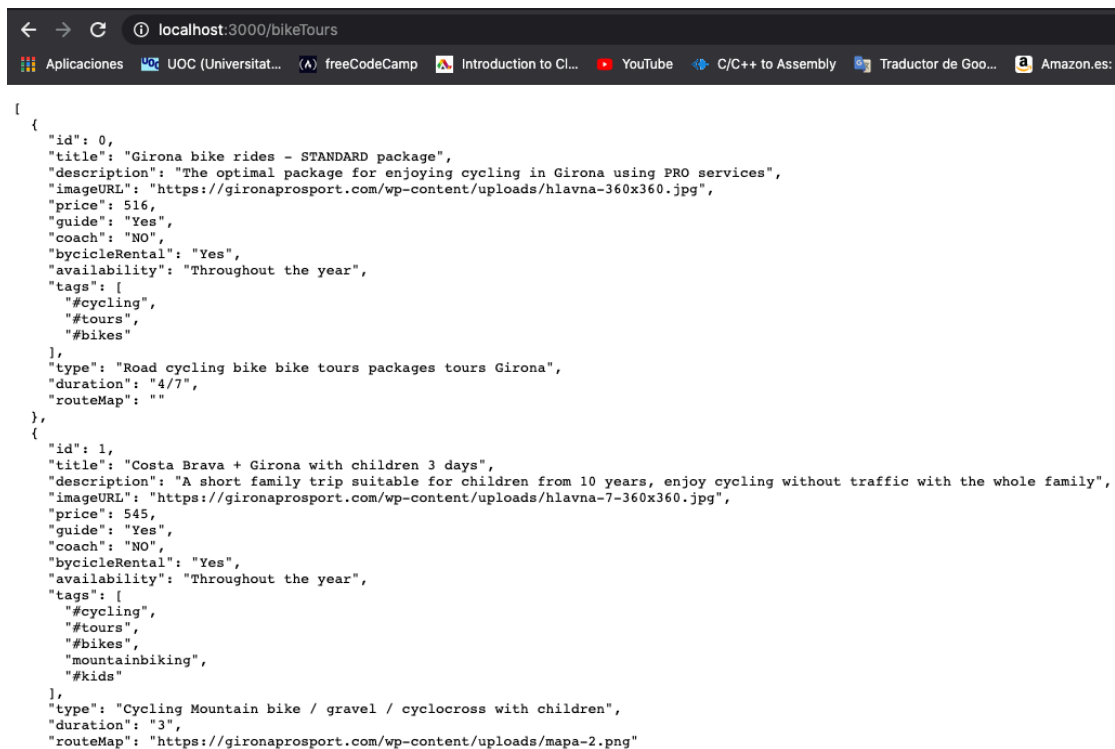
A l'arxiu «db.json» i tenim tota la informació de la nostra API, com poden ser la ID, el TITLE, DESCRIPTION, IMAGEURL, ...



Per veure un exemple en local de la nostra API hem de escriure la següent comanda:
`% json-server --watch db.json` (dintre del arxiu a on tenim la DB.json).

```
[danieldominguez@MacBookPro bikeTours-api % json-server --watch db.json
\{^_^\}/ hi!
Loading db.json
Done
Resources
http://localhost:3000/bikeTours
Home
http://localhost:3000
Type s + enter at any time to create a snapshot of the database
Watching...
GET /bikeTours 200 42.759 ms - -
```

I en obrir la URL de RESOURCES podem veure la nostra API.

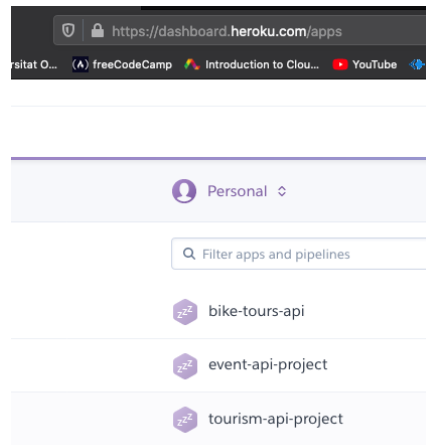


```
{
  {
    "id": 0,
    "title": "Girona bike rides - STANDARD package",
    "description": "The optimal package for enjoying cycling in Girona using PRO services",
    "imageUrl": "https://gironaproport.com/wp-content/uploads/hlavna-360x360.jpg",
    "price": 516,
    "guide": "Yes",
    "coach": "NO",
    "bicycleRental": "Yes",
    "availability": "Throughout the year",
    "tags": [
      "#cycling",
      "#tours",
      "#bikes"
    ],
    "type": "Road cycling bike bike tours packages tours Girona",
    "duration": "4/7",
    "routeMap": ""
  },
  {
    "id": 1,
    "title": "Costa Brava + Girona with children 3 days",
    "description": "A short family trip suitable for children from 10 years, enjoy cycling without traffic with the whole family",
    "imageUrl": "https://gironaproport.com/wp-content/uploads/hlavna-7-360x360.jpg",
    "price": 545,
    "guide": "Yes",
    "coach": "NO",
    "bicycleRental": "Yes",
    "availability": "Throughout the year",
    "tags": [
      "#cycling",
      "#tours",
      "#bikes",
      "mountainbiking",
      "#kids"
    ],
    "type": "Cycling Mountain bike / gravel / cyclocross with children",
    "duration": "3",
    "routeMap": "https://gironaproport.com/wp-content/uploads/mapa-2.png"
  }
}
```

Per a poder carregar la nostra API a Heroku, primer hem de tenir instal·lat a la nostra màquina i també tenir un compte creat. Després amb una mica de **màgia** al *backend* (package.json) i amb l'ajuda d'uns *script* (server.js) podem pujar els arxius a Heroku.

Per a instal·lar Heroku al MacBookPro hem fet servir BREW:
`% brew tap heroku/brew && brew install heroku`

Un cop ho tenim tot instal·lat només cal crear els projectes a Heroku:
`% heroku create bikeTours-api`



Una completa explicació del procediment es pot veure al [vídeo](#) de Ania Kubow.

Per a poder veure els canvis aplicats durant tota l'etapa de desenvolupament, hem optat per l'opció d'un LIVE RELOAD.

També ens ha facilitat molt el fet de treballar amb Capacitor a Ionic en comptes de Cordova, ja que el desplegament a Xcode ha anat sense problemes (amb Cordova de vegades no s'aplicaven els canvis implementats al codi).

Per a poder treballar amb Capacitor, dintre de la terminal del IDE (Visual Studio Code) hem creat el projecte de la següent forma:

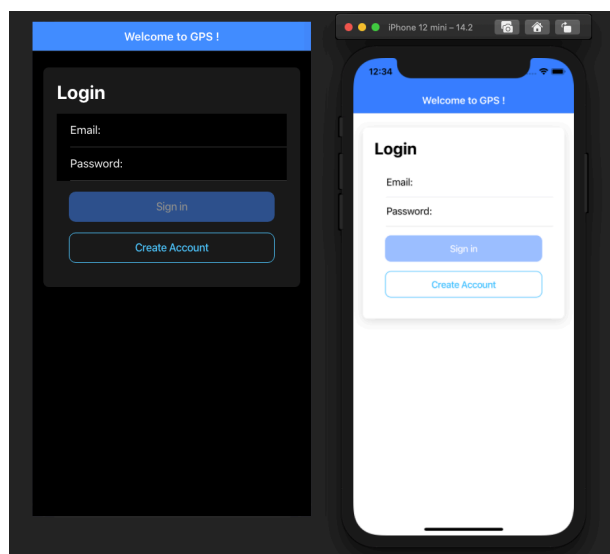
```
# Crear la aplicació
ionic start gironaps --type=angular --capacitor --package-id=com.uoc.gironaps

# Generar la part WEB de la app
ionic build

# Generar les plataformes
npx cap add ios/android

# Procediment estandard cada cop que s'instal·len nous mòduls
ionic build && npx cap copy

# Livereload
ionic cap run ios -l --external
```



Comparació de l'aplicació en WEB i en iOS Simulator

Per preparar l'entorn a Xcode simplement s'ha d'executar «`ionic cap run ios`» O «`ionic cap run ios -l --external`» si també es vol fer ús del *LIVERELOAD*.

3.2 – Preparar aplicació per ferla servir en un entorn de treball

Lo més ràpid serà clonar el projecte des del repositori de Github:

<https://github.com/dadobe/gironaps>

```
$ git clone [URL del repositori]
```

Un cop clonat, s'obre amb el IDE que es vol fer servir, nosaltres hem fet servir Visual Studio Code però també n'hi ha d'altres com WebStorm (de JetBrains).

Si el projecte no tingués creades les versions per a iOS i Android, es poden crear utilitzant les següents comandes:

```
ios: ionic capacitor add ios
```

```
Android: ionic capacitor add android
```

Per provar l'aplicació en un entorn per a iOS o Android, s'ha d'obrir per a Xcode o Android Studio.

```
Xcode: ionic capacitor open ios
```

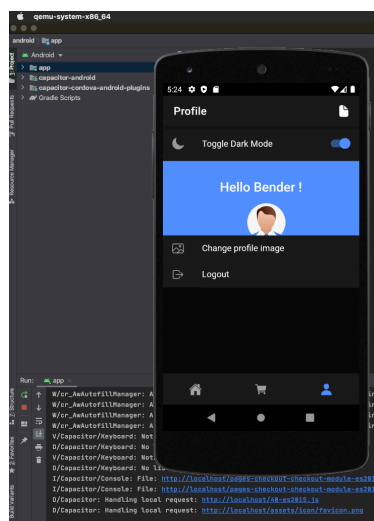
```
Android Studio: ionic capacitor run android
```

Per fer servir el Live Reload:

```
ios: ionic cap run ios -l --external
```

```
Android: ionic capacitor run android -l --host=YOUR_IP_ADDRESS
```

Veiem a continuació un exemple de simular el projecte en entorn Android.



Capítol 4 - Anàlisi aplicació

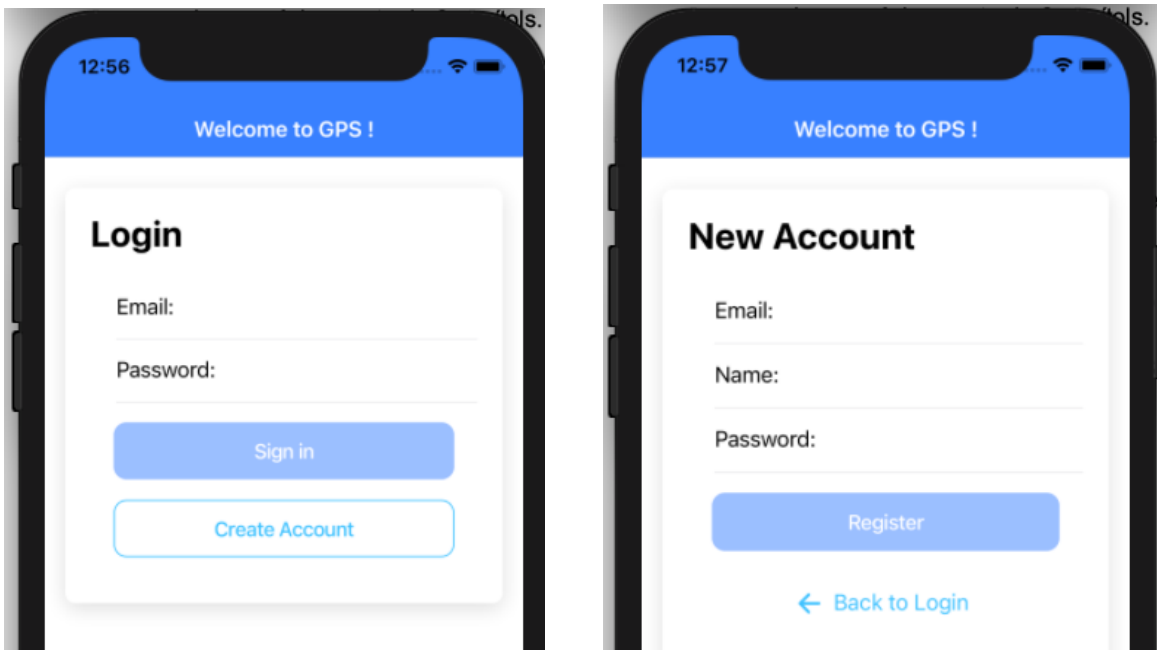
4.1 – Funcionalitats

L'aplicació consta d'una pantalla inicial, el LOGIN, a través de la qual també es pot anar a REGISTER.

Un cop loguejat, es veu una aplicació de tipus TABS. Aquesta consta de 3 tabs diferents:

- Tab 1 - seria el HOME, en aquí es poden veure les diferents activitats
- Tab 2 - seria el E-Shop, en aquí es poden adquirir diferents materials per a ciclisme
- Tab 3 - seria el Profile, en aquí l'usuari pot canviar el seu avatar, canviar a *DARKMODE* i veure les seves comandes.

Encara que inicialment no s'havia previst tenir un registre d'usuaris, s'ha decidit implementar ja que no ens desviava gaire del temps de programació inicialment establert.



Per a poder provar la funcionalitat del LOGIN, es va crear manualment un usuari DUMMY a Firebase, una plataforma situada en el núvol, integrada amb Google Cloud Platform, que utilitza un conjunt d'eines per a la creació i sincronització de projectes. Per facilitar la part d'autenticació, hem fet ús de Firebase Auth, que és un servei que pot autenticar els usuaris utilitzant únicament codi de la banda de el client. Inclou l'autenticació mitjançant proveïdors d'inici de sessió com Facebook, GitHub, Twitter, Google ... encara que nosaltres hem fet servir l'inici de sessió mitjançant correu electrònic únicament.

dummy@dummy.com



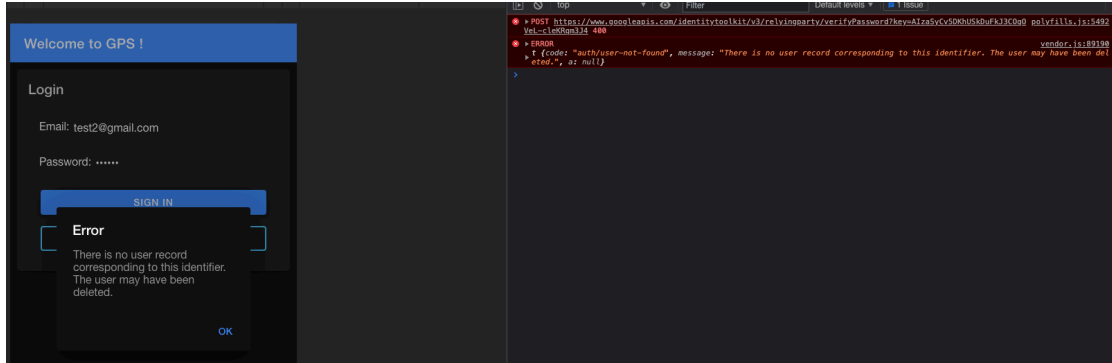
10 may. 2021

12 may. 2021

SVNZ4ncCeVcXgdtB3IHMXr093up2

Si l'usuari no es existeix perquè és la primera vegada que accedeix a l'aplicació té l'opció de donar-se d'alta a REGISTER. Un cop registrat, es crea una nova entrada a Firebase amb les dades i el *password* del nou usuari registrat.

Si intentem entrar amb un usuari que no existeix, el sistema respon amb una notificació explicant la raó de l'error.



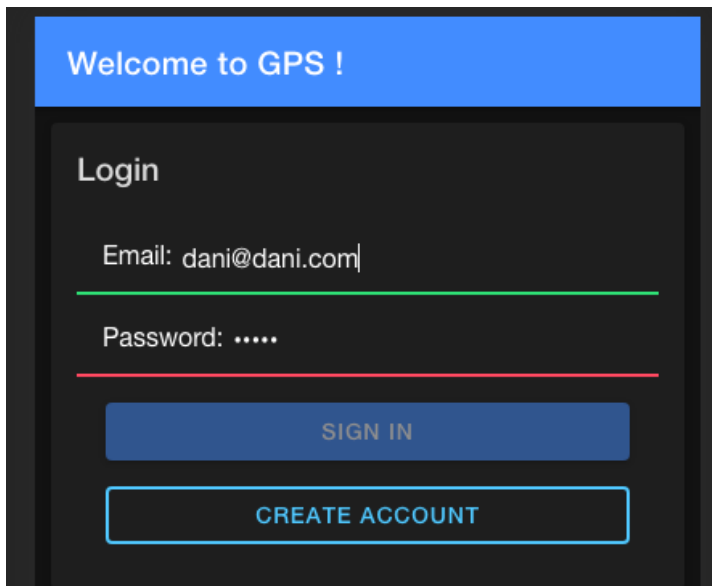
Al codi s'ha implementat com a validadors que la mida mínim d'una contrasenya sigui 6.

```

this.registerForm = this.fb.group({
  email: ['', [Validators.required, Validators.email]],
  password: ['', [Validators.required, Validators.minLength(6)]],
  fullname: ['', Validators.required]
});

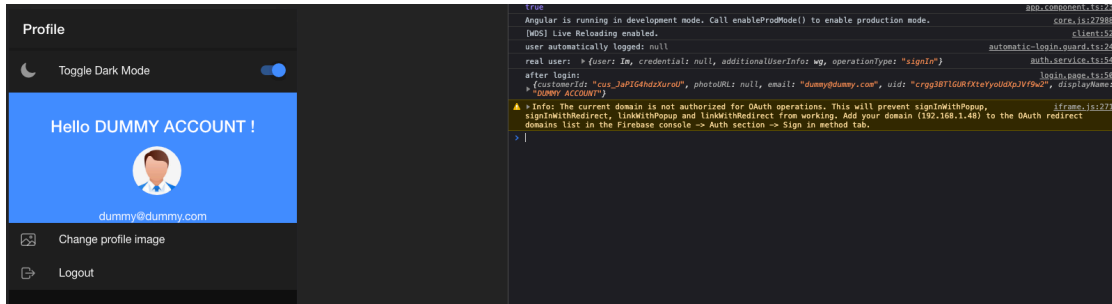
this.loginForm = this.fb.group({
  email: ['', [Validators.required, Validators.email]],
  password: ['', [Validators.required, Validators.minLength(6)]]
});
    
```

Si no s'assoleix aquest mínim, la finestra del camp contrasenya no es mostra com a activa (queda subratllat en vermell) i el botó de *SIGN IN* es mostra desactivat.

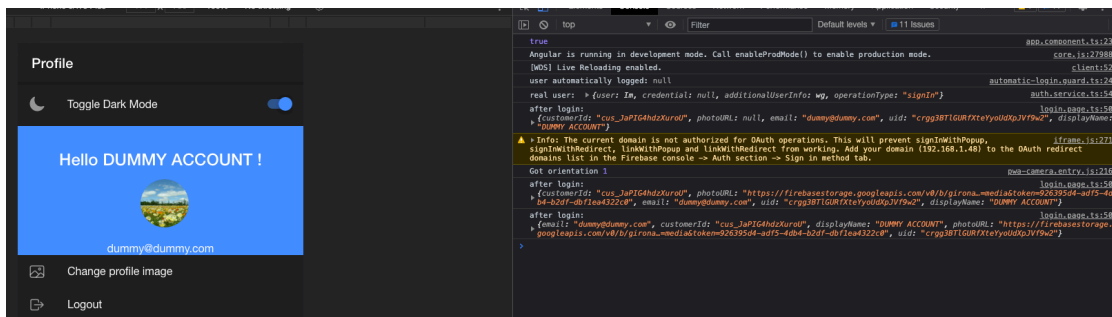
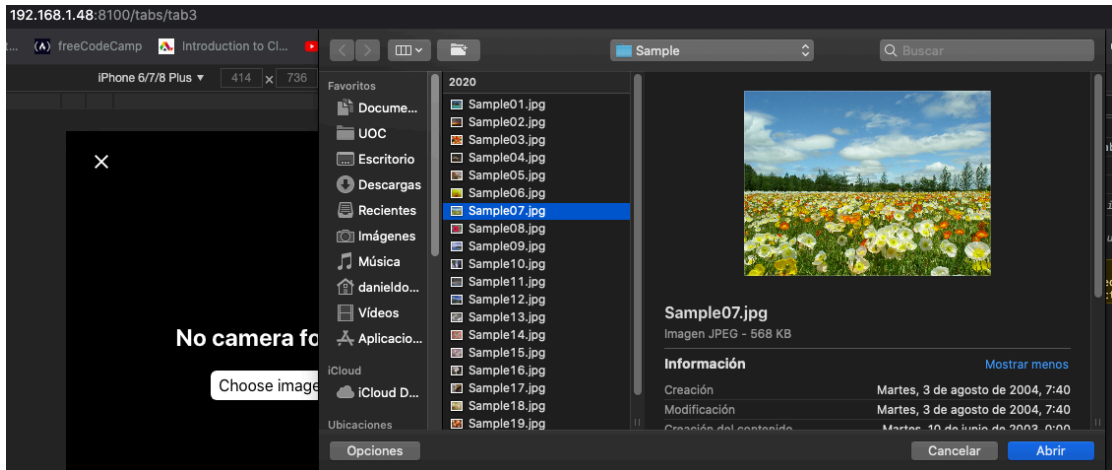


Per a donar d'alta un nou usuari s'ha de clicar a **CREATE ACCOUNT**. Aquí, mitjançant un *flipper* creat amb SCSS hem implementat una petita acció que fa que la pantalla de LOGIN es giri i mostri la de REGISTER. Hem cregut que seria un «nice feature» a implementar que no ens ha costat gaire temps. La primera vegada que accedeix un nou usuari, al tab de Perfil (el tab 3) es mostren dades com el nom, el correu electrònic i una *dummy image*.

Com es pot veure al «console.log», no es passa cap imatge, per això s'utilitza un avatar.



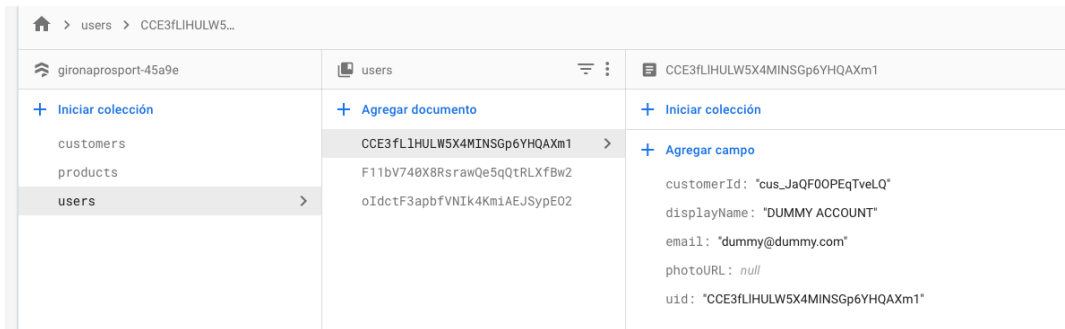
Per a canviar la imatge s'ha de clicar a **CHANGE PROFILE IMAGE**.



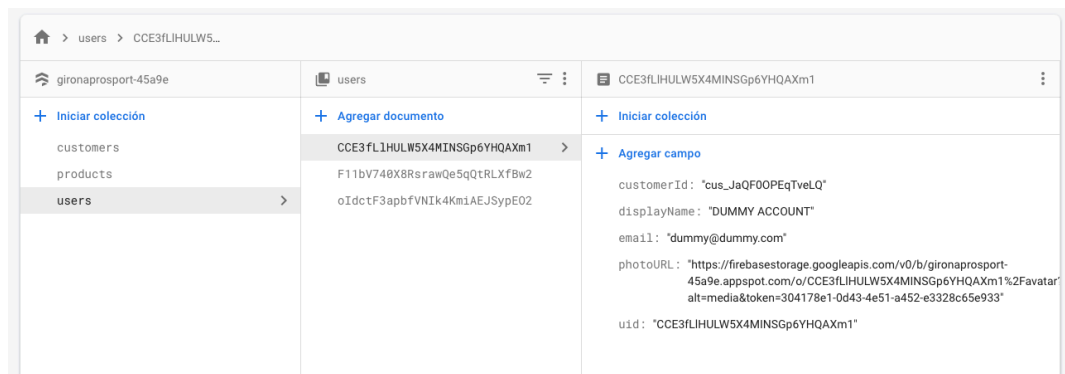
Com es veu al «console.log» es puja la imatge a Firebase i «photoURL» ara si que té un paràmetre.

Si mirem a Firebase, la col·lecció «users» té a *photoURL* la imatge que hem seleccionat fa uns instants. Veiem una comparativa d'abans i de després de canviar de l'avatar a una imatge.

Abans:

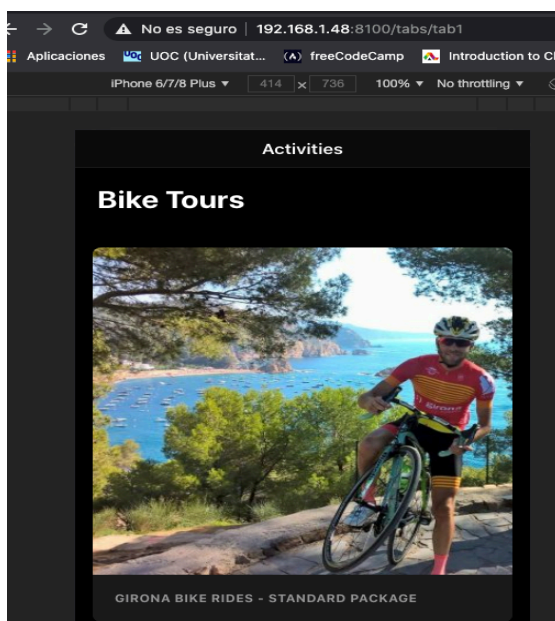


Després:

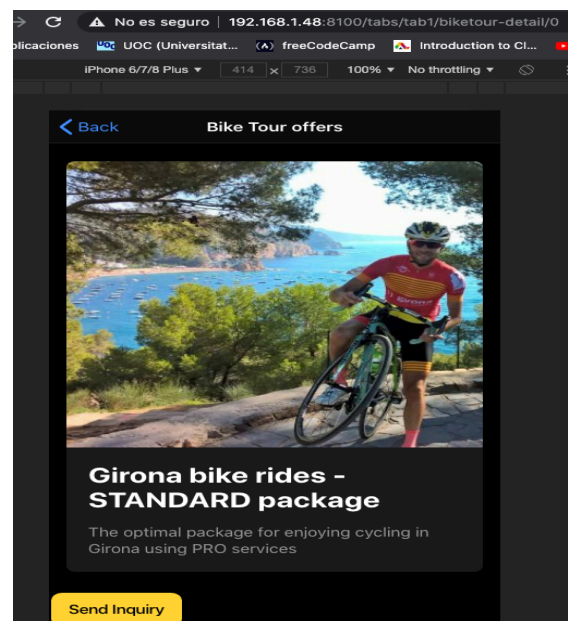


Un cop l'usuari s'ha registrat i accedeix, apareix a la pantalla inicial (Tab1) on es mostren les diferents activitats.

Si clica a alguna, l'usuari es guiat als detalls de l'activitat (aquest *routing* es fa a través de la ID de l'activitat).



Pàgina principal de Bike Tours



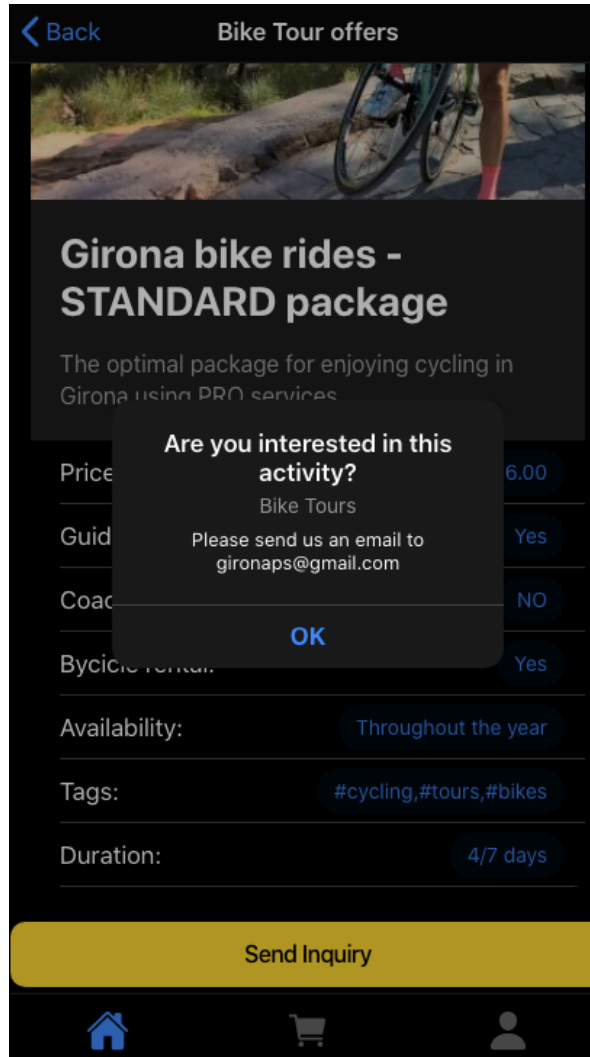
Pàgina de l'activitat amb ID=0 de Bike Tours

El mateix succeeix amb les altres activitats com *Tourism* i *Events*.

En cas que l'usuari estigui interessat en rebre més informació d'una activitat en concreta, s'ha afegit un botó «Send Inquiry» al final de l'activitat en qüestió.

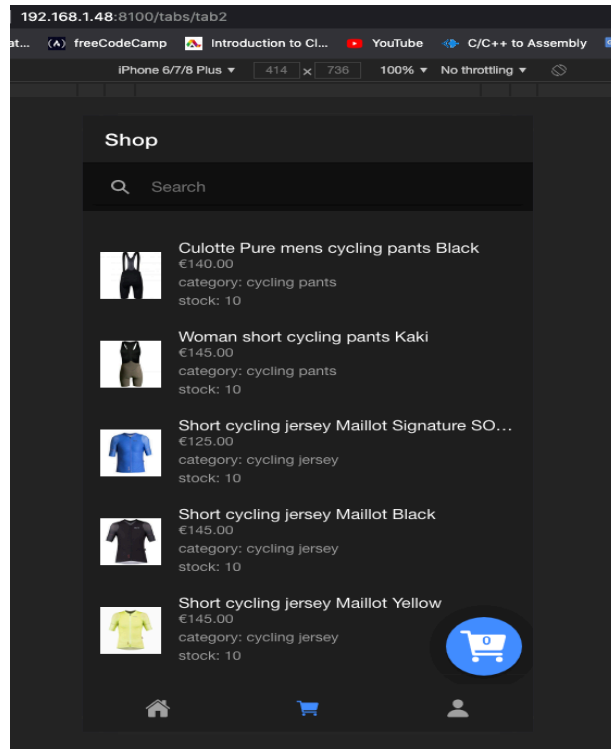
La idea inicial era que s'obris l'app nativa de correu electrònic, però degut a incompatibilitats entre paquets de Capacitor i Cordova, no s'ha aconseguit crear aquesta opció.

Com a «bypass» s'ha optat de convertir aquest botó en un *AllertControler* que mostri per pantalla un missatge.

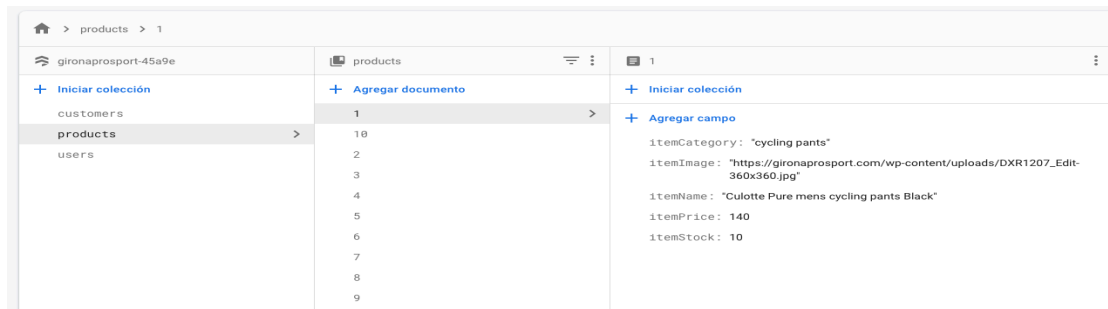


El mateix s'ha implementat per a la resta d'activitats a on el missatge canvia lleugerament depenent si es un Bike Tour, un Tourism o un Event.

Al Tab 2, es pot veure el E-Shop, el qual incorpora un *SEARCH BAR* molt simple, el qual per falta de temps no s'ha pogut implementar amb les funcionalitats desitjades, com poder clicar directament sobre la cerca i acabar en la secció «Item Details».



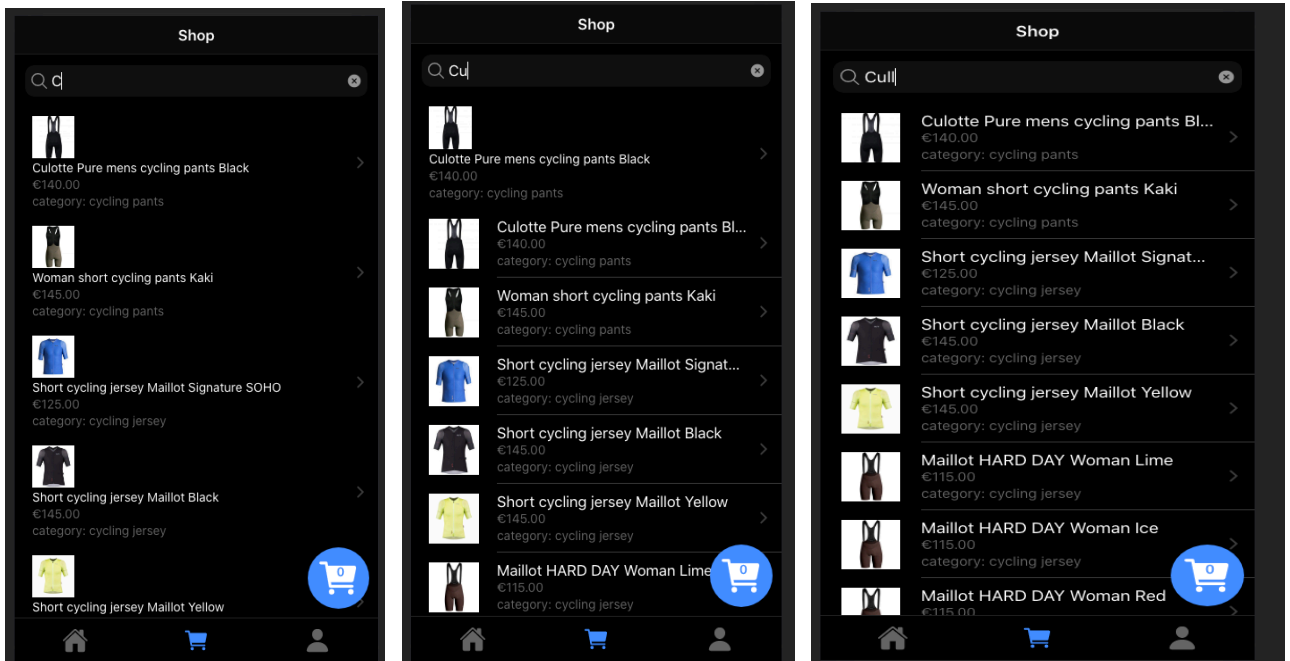
En el Tab 2 podem veure tots els productes que tenim allotjats a Firebase. Per facilitar la feina s'havia provat de pujar els productes a Firebase mitjançant un Script, però no ha acabat de funcionar i s'han hagut de crear els productes manualment.



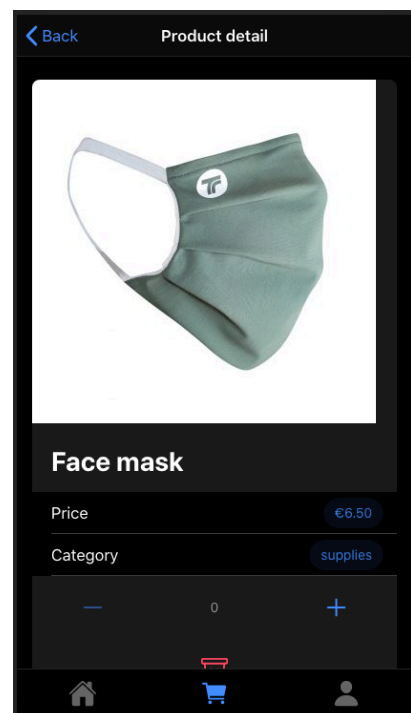
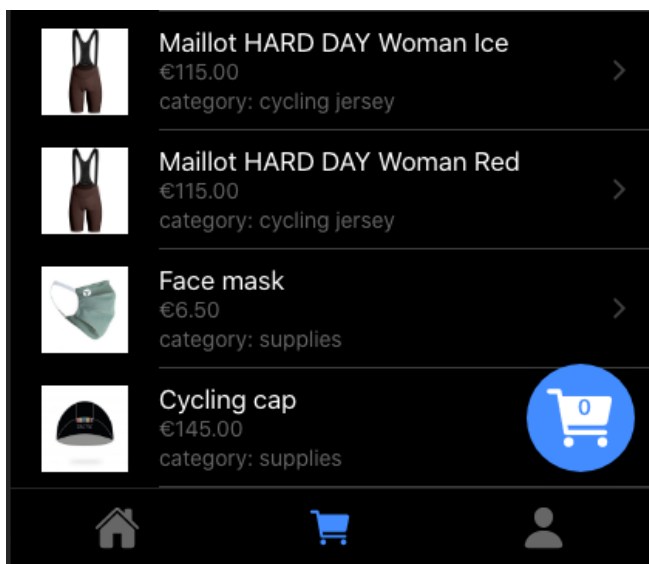
Com es pot veure, es disposa d'un catàleg curt però variat a on els camps importants de cada *ITEM* són:

- *itemCategory*: descriu la categoria del producte
- *itemImage*: la imatge del producte
- *itemName*: el nom del producte
- *itemPrice*: el preu del producte
- *itemStock*: el stock actual del producte

En el E-Shop també s'ha implementat un *Searchbar*, que filtra els productes per *itemName*. Si es realitza una cerca amb la lletra 'C', es mostren tots el productes que continguin aquest lletra en el *itemName*. Si cerquem per 'Cu', mostra l'únic *item* «cullotte» i si ampliem la cerca amb «Cull» no mostra resultats per que no existeix cap producte amb aquest nom.



Per realitzar una compra d'un producte, s'ha de clicar el producte que es vol adquirir. Això portarà a l'usuari a la pàgina de *DETAILS* del producte clicat. Per exemple, seleccionem el producte «Face Mask» i en clicar en aquest producte es veu seu detall.




En aquí es pot veure el nom del producte, el seu preu, la seva categoria i uns botons que serveixen per a afegir o treure el producte al *ShoppingCart*.

Es pot veure que el '-' està des habilitat, això s'ha implementat per evitar que l'usuari tregui de la cistella de la compra més productes dels que ha afegit.

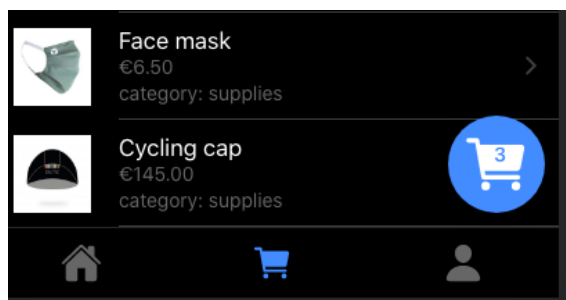
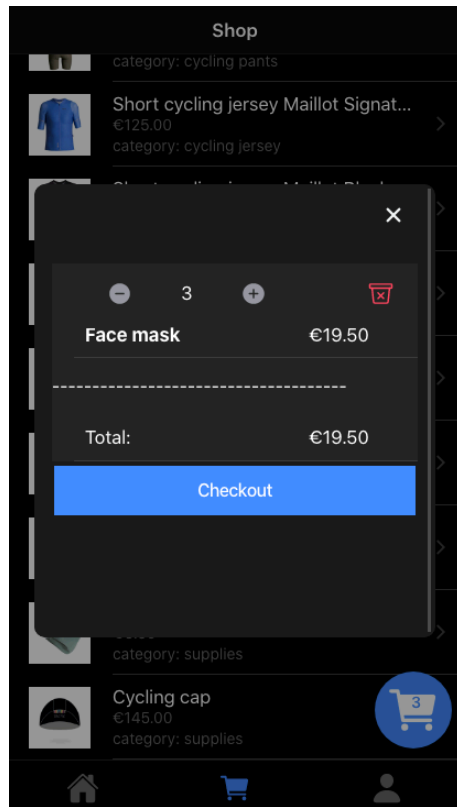
```
<ion-button fill="clear" (click)="removeItemFromCart()" [disabled]="amount == 0"> <!-- disables MINUS option when amount = 0 -->
```

```
<ion-icon name="remove" slot="icon-only"></ion-icon>
```

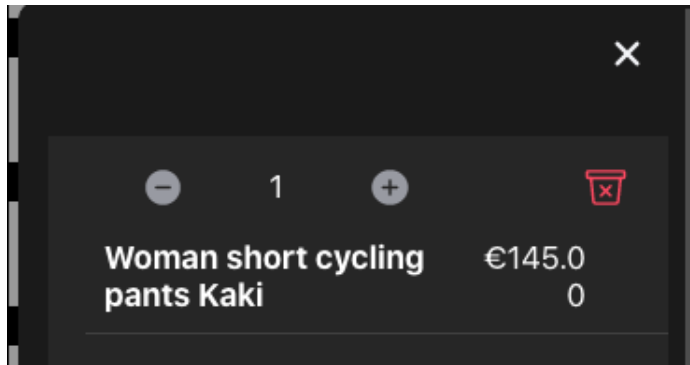
L'usuari també té l'opció de buidar completament la cistella clicant sobre la icona: 

Un cop s'ha seleccionat la quantitat del producte, es pot tornar a la pantalla anterior amb el *BACK BUTTON* que es troba a dalt a l'esquerra. Des d'aquí es pot tornar a seleccionar un nou producte i afegir-lo a la cistella.

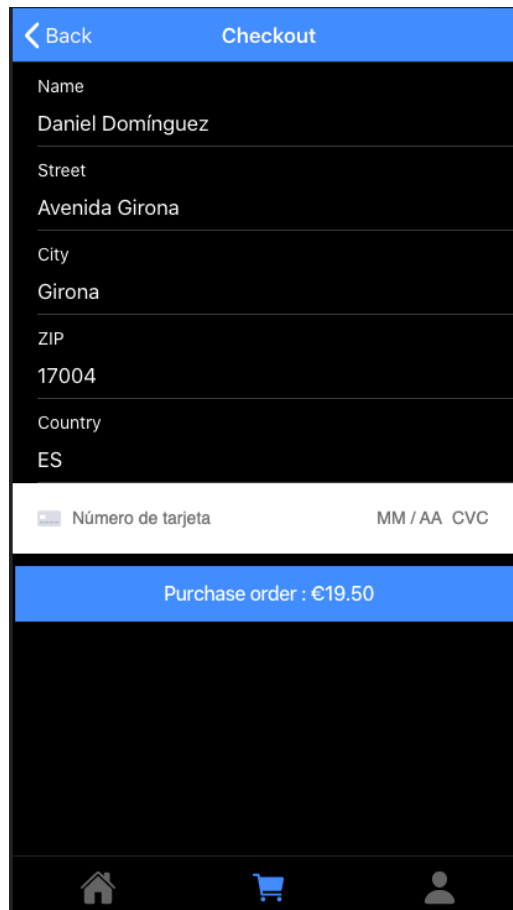
Un cop l'usuari ja té seleccionats tots els productes que vol comprar, ha de clicar a la icona del *ShoppingCart* abaix a la dreta per fer el *Checkout*. Dins de la icona del *ShoppingCart* es veu també la quantitat de productes que s'han afegit.



També es té l'opció de buidar la cistella de la compra completament en aquest punt, clicant a la icona vermella.





Un cop s'ha decidit què comprar, en fer el *Checkout*, l'usuari és guiat a la pàgina de pagaments (afegir les dades de *Shippment* i el número de la targeta).



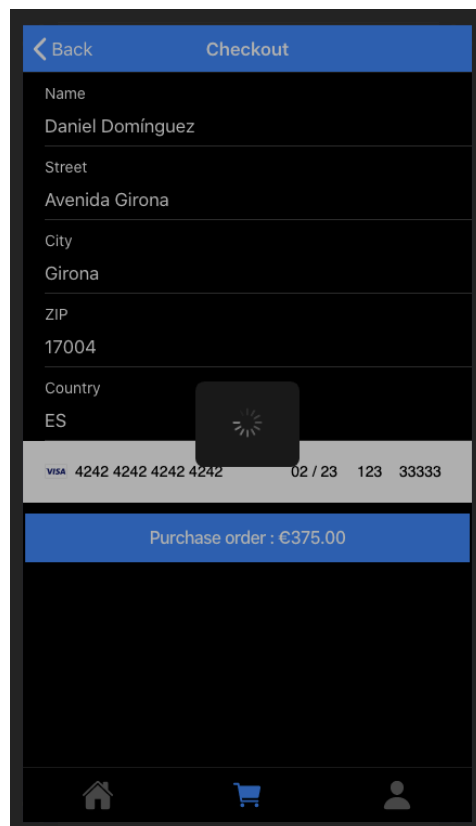
Per provar aquesta part de l'aplicació, s'ha fet servir un número de targeta que es pot trobar a *StripeDocs* sota «Test your integration». Farem servir la 4242 4242 4242 4242 que es veu a [Basic Test Card Number](#).

Basic test card numbers

Genuine card information cannot be used in test mode. Instead, use any of the following test card numbers, a valid expiration date in the future, and any random CVC number, to create a successful payment. Each basic test card's billing country is set to U.S. If you need to create test card payments using cards for other billing countries, use our [international test cards](#).

Card numbers	Tokens	PaymentMethods	
NUMBER	BRAND	CVC	DATE
4242 4242 4242 4242 	Visa	Any 3 digits	Any future date
4000 0566 5566 5556 	Visa (debit)	Any 3 digits	Any future date

L'usuari afegeix les dades per al *Shippment* i el número de la targeta per terminar la compra.

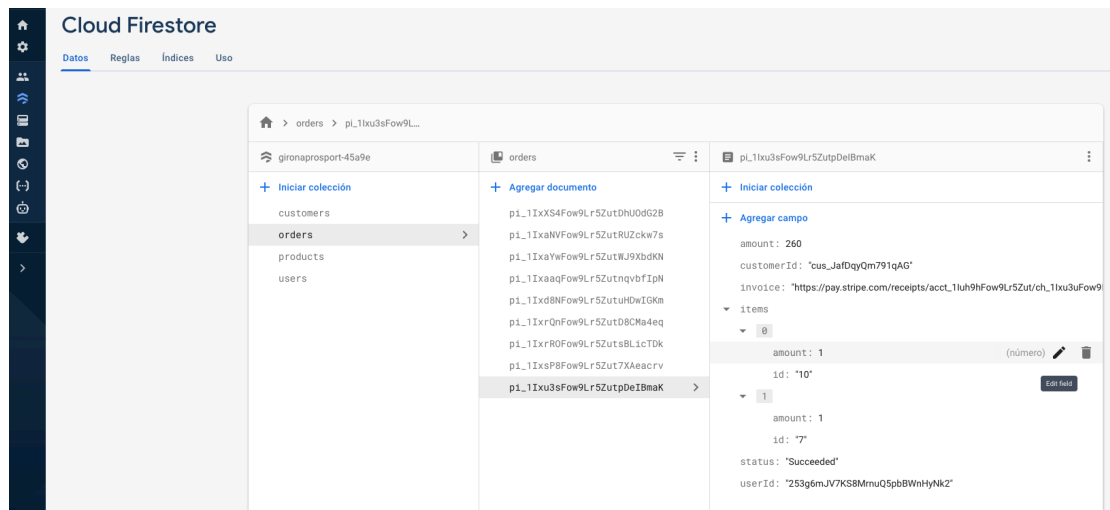


En cas que un dels camps a omplir es deixessin en blanc, no es podrà efectuar la compra ja que el botó estarà des habilitat.

Un cop realitzat el *PURCHASE*, l'usuari torna a la pantalla del *ESHOP*.

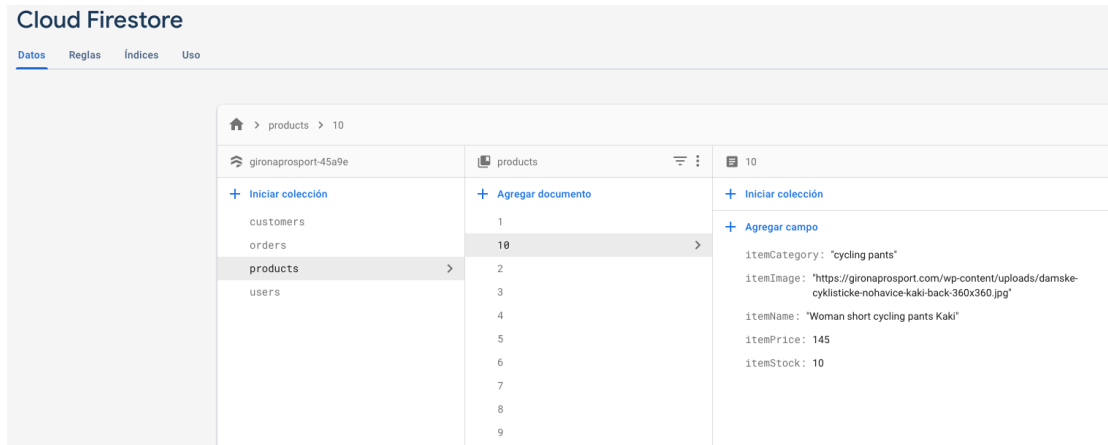
Si donem un cop d'ull al nostre backend, veurem com s'han creat els registres de la comanda.

A Firebase:

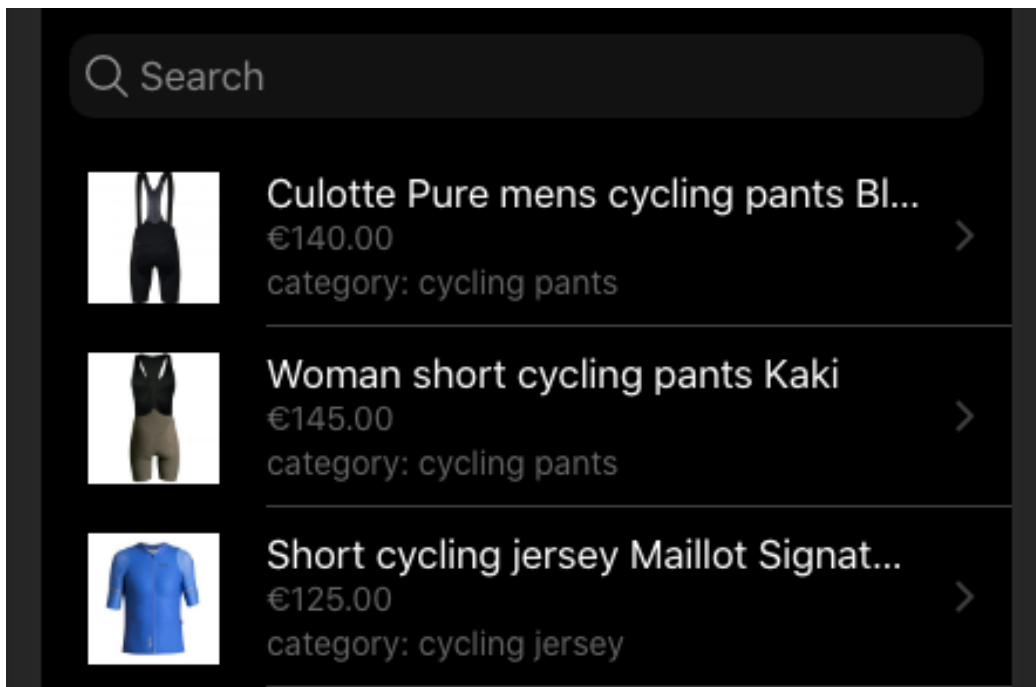


En aquí es veu com un client ha realitzat una compra per valor (amount) de 260 havent efectuat una compra del producte amb ID=10 i el producte amb ID=7.

Mirant a la col·lecció de PRODUCTS podem identificar aquest productes.



El producte amb ID=10 és un «Woman short cycling pants Kaki». Que és en la nostra base de dades el segon producte de la llista. A l'aplicació també apareix en aquella posició.



Si mirem a STRIPE, veurem les dades de la comanda i les de l'usuari que la ha realitzada.

Payments

Filter Export Create payment

AMOUNT	DESCRIPTION	CUSTOMER	DATE
€260.00	Succeeded ✓ pi_1Ixu3sFow9Lr5ZutpDeIBmaK	patrick@gmail.com	2 Jun, 15:24
€145.00	Incomplete pi_1Ixu1IFow9Lr5Zut5ujYb6zV		2 Jun, 15:22
€375.00	Succeeded ✓ pi_1IxsP8Fow9Lr5Zut7XAeacrV	patrick@gmail.com	2 Jun, 13:38
€125.00	Succeeded ✓ pi_1IxrR0Fow9Lr5ZutsBLicTDk	patrick@gmail.com	2 Jun, 12:36
€125.00	Succeeded ✓ pi_1IxrQnFow9Lr5ZutD8Cma4eq	patrick@gmail.com	2 Jun, 12:36
€19.50	Incomplete pi_1IxrKrFow9Lr5ZutPv1PFkLD		2 Jun, 12:30

Es veu un *PAYMENT* de 260.00 € que correspon al *customer* que estava connectat mentre es realitzava la comanda. En aquest cas «patrick@gmail.com». També es veu la data en la qual es va efectuar la comanda, el 2 de Juny a les 15:24 hores.

Si es clica a la fila d'aquesta comanda, es veu una nova pantalla amb més informació sobre la compra realitzada.

TEST DATA

PAYMENT

€260.00 EUR Succeeded ✓

Date	Customer	Payment method	Risk evaluation
2 Jun, 15:24	patrick@gmail.com	visa **** 4242	0 Normal

Timeline

- Payment succeeded
2 Jun 2021, 15:24
- Payment started
2 Jun 2021, 15:24

Payment details

Statement descriptor	GIRONA PRO SPORT
Amount	€260.00
Fee	€7.79
Net	€252.21
Status	Succeeded
Description	No description Edit

Payment method

ID	pm_1Ixu3tFow9Lr5ZutfLCzsDeY	Owner	Patrick Dominguez
Number	**** 4242	Owner email	patrick@gmail.com
Fingerprint	KUDCowNm4GLXZ8SB	Address	Avenida Girona
Expires	04 / 2022		Girona, 33333, ES

A la part de *PAYMENT METHOD* es veuen les dades que l'usuari ha introduït a l finestra de CHECKOUT abans de la compra: nom, correu electrònic, adreça ...

A la secció de *customers* de Stripe, es poden veure tots els usuaris existents des de la seva creació a la pàgina de REGISTER (alguns apareixen repetits per que han sigut esborrats a Firebase directament i creats de nou a l'aplicació).

Customers Filter Export + Add customer

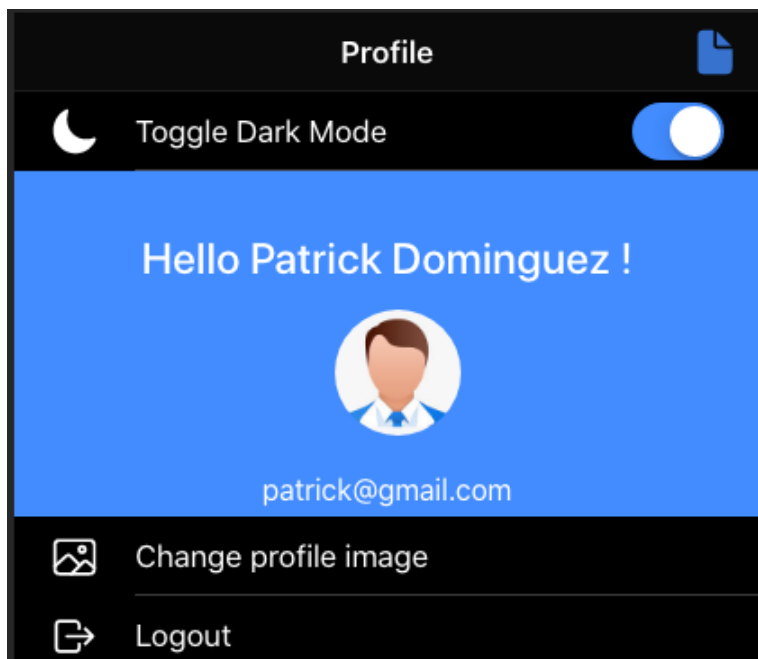
[All](#) [Top 100](#)

EMAIL	DESCRIPTION	DEFAULT SOURCE	CREATED
<input type="checkbox"/> patrick@gmail.com	—	—	1 Jun, 11:28
<input type="checkbox"/> patrick@gmail.com	—	—	1 Jun, 11:22
<input type="checkbox"/> dummy@dummy.com	—	—	31 May, 19:59
<input type="checkbox"/> dummy@dummy.com	—	—	31 May, 19:00
<input type="checkbox"/> testing@test.com	—	—	29 May, 12:11

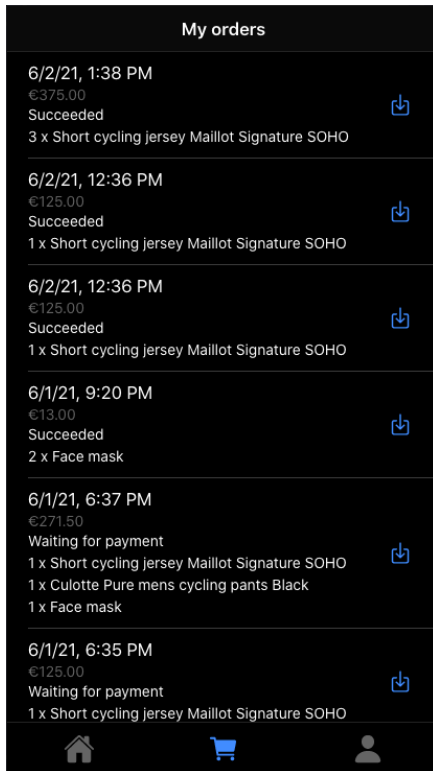
5 results Previous Next

Per verificar la compra, es pot anar al tab de *PROFILE* per veure les comandes.

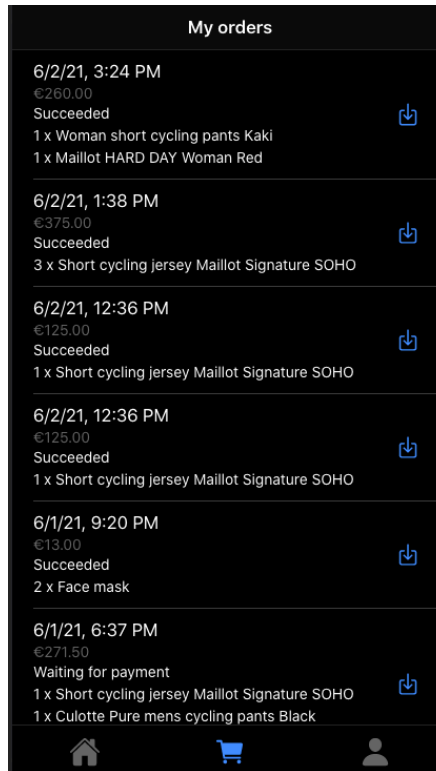
En la pàgina del perfil, a dalt a la dreta tenim una icona amb forma de document a través de la qual es pot anar a les comandes de l'usuari connectat.



Clicant a la icona, l'usuari és guiat a la pàgina de «My orders», a on es veu un llistat de totes les ordres realitzades des de que s'ha donat d'alta.



Abans de la compra de 260€

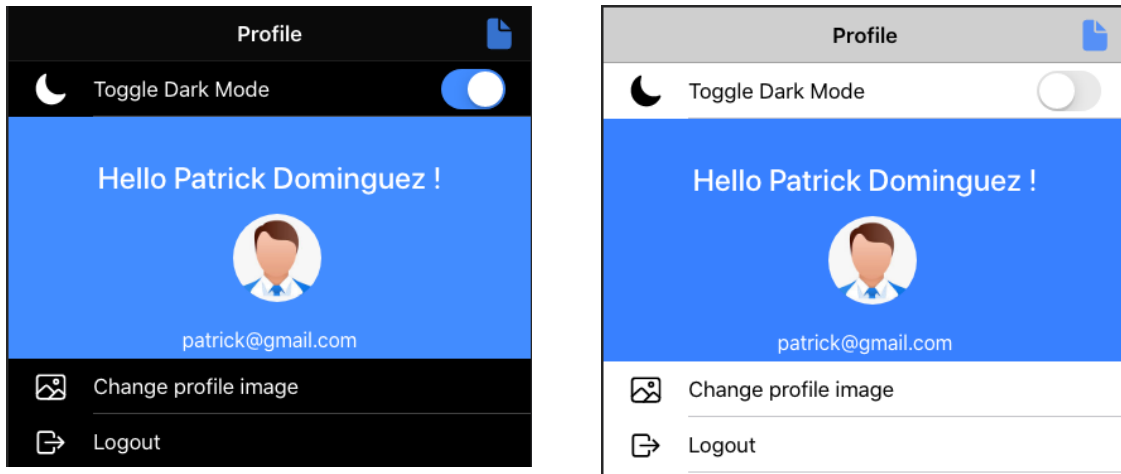


Després de la compra de 260€

Per veure el «Invoice», l'usuari ha de clicar a la icona de descàrrega que està situada a la dreta de cada ordre. Si el status es *SUCCEDED*, s'obrirà la copia de la compra. Si el status es diferent a *SUCCEDED*, no es genera cap *invoice* i la compra queda pendent de ser verificada.



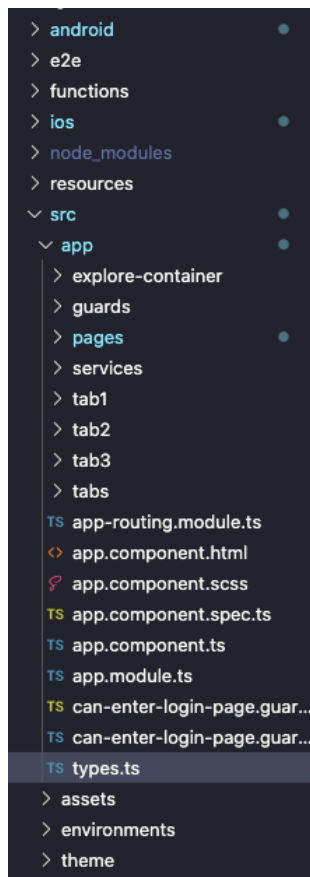
A la pantalla del perfil també es té la possibilitat de canviar entre DARK / LIGHT mode.



També té l'opció de canviar la foto del perfil a «Change profile image».

4.2 – Explicació del codi

El codi de l'aplicació consta de varies parts.



Tindríem la part pròpiament de l'aplicació a la carpeta de «src» i la part de backend o que connecta la part del Eshop amb Firebase i Stripe a la carpeta de «functions».

Com es veu, l'aplicació es va generar fent servir la plantilla de Ionic per a aplicacions amb TABS (tab1, tab2, tab3, tabs).

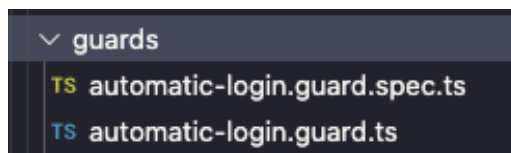
Tab1: tindriem el codi per a la part de les Activitats, com serien el Bike Tour, el Tourism i el Event.

Tab2: el eshop

Tab3: el profile

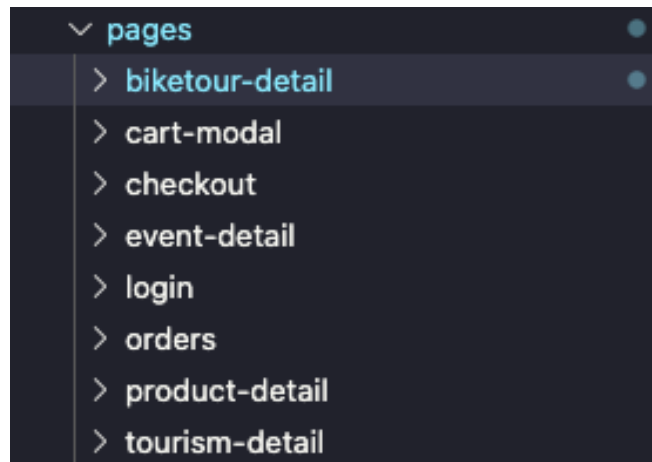
Tabs: aquí pràcticament no n'hi ha codi, només el «routing» que és la part important de navegació entre les diferents pantalles.

A la carpeta de «guards» esta la implementació del AUTOLOGIN, que s'encarrega de deixar entrar a l'usuari de nou a l'aplicació sense necessitat de fer un login, sempre que tingui el caché en la memòria (que no s'hagi esborrat).

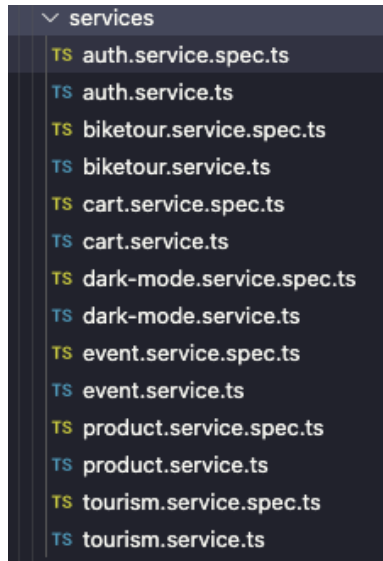


A la carpeta de «pages» hi tenim:

- la part que gestiona el LOGIN (i REGISTER)
- la part que gestiona les pagines de detalls de les diferents activitats (bike tour, tourism, event)
- la part que gestiona el *CART MODAL*
- la part que gestiona les ordres i el checkout



A la carpeta «services» es gestionen els serveis necessaris per fer funcionar les activitats (bike tour, toruism, event), l'autenticació (login / register), la cistella de la compra, i l'opció de canviar a Dark / Light mode.



Després també hi ha un arxiu anomenat «types.ts» a dins el qual hi tenim les diferents interfícies que es fan servir en l'aplicació.

```
//Biketours interface
export interface Biketours {
  id: number;
  title: string;
  description: string;
  imageURL: string;
  price: number;
  guide: string;
  coach: string;
  bicycleRental: string;
  availability: string;
  tags: string[];
  type: string;
  duration: string;
  routeMap: string;
}
```

```
//Tourism interface
export interface Tourism {
  id: number;
  title: string;
  description: string;
  imageURL: string;
  price: number;
  category: string[];
  guide: string;
  coach: string;
  stages: number;
  distance: number;
  unevenness: number;
  bicycleRental: string;
  availability: string;
  tags: string[];
  type: string;
  duration: string;
  routeMap: string;
  details: string;
  journey: string[];
}
```

```
//Event interface
export interface Event {
  id: number;
  title: string;
  description: string;
  imageURL: string;
  price: number;
  category: string[];
  difficulty: string;
  guide: string;
  coach: string;
  stages: number;
  distance: number;
  unevenness: number;
  bicycleRental: string;
  availability: string;
  tags: string[];
  type: string;
  duration: string;
  routeMap: string;
  details: string[];
  descriptionRoute1: string[];
  linkRoute1: string;
  imageRoute1: string;
  descriptionRoute2: string[];
  linkRoute2: string;
  imageRoute2: string;
  journey: string[];
}

export interface LoginCredential {
  email: string;
  password: string;
}
```

A l'arxiu «environment.ts» hi tenim la part que connecta al backend de Firebase per a poder realitzar les operacions de Autenticació per exemple.

Capítol 5 - Proves aplicació

Les proves realitzades han sigut entre la versió web, una versió en simulador iOS i un dispositiu d'Apple iPhone 6 Plus.

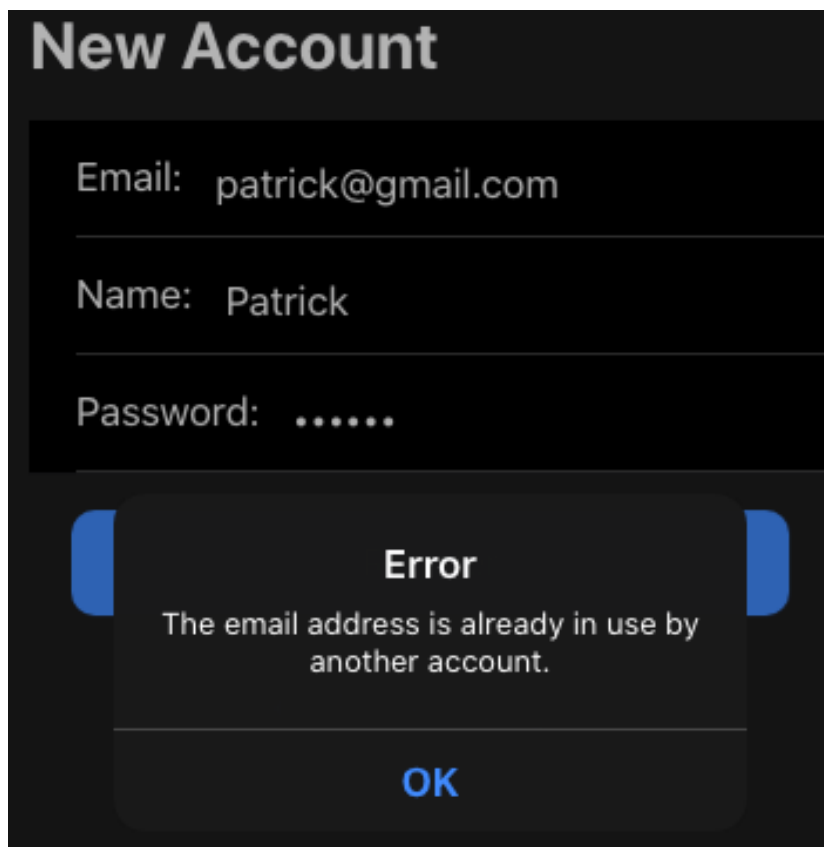
Tests realitzats:

Crear un nou usuari:

Identificador	CU-001
Nom	Register
Prioritat	Alta
Descripció	Usuari es registra al sistema
Actors	Usuari
Pre-condicions	Usuari no esta registrat
Iniciat per	Usuari
Flux	
Post-condicions	Usuari es registra amb èxit

Quan un nou usuari és creat al sistema, es connecta al backend (Firebase) per veure si ja existeixen aquestes credencials.

En cas que existeixin, torna un missatge d'error.



Identificador	CU-002
Nom	Login
Prioritat	Alta
Descripció	Usuari es logueja al sistema
Actors	Usuari
Pre-condicions	Usuari no esta loguejat
Iniciat per	Usuari
Flux	
Post-condicions	Usuari es logueja amb èxit

Usuari ja existent es connecta al sistema i se li permet accés a l'aplicació.

Identificador	CU-003
Nom	Purchase order
Prioritat	Alta
Descripció	Usuari efectua una compra
Actors	Usuari
Pre-condicions	Usuari estigui logiejat
Iniciat per	Usuari
Flux	
Post-condicions	Usuari realitza una compra

Usuari connectat és capaç de realitzar una compra.

Identificador	CU-004
Nom	Search product
Prioritat	Alta
Descripció	Usuari cerca un producte
Actors	Usuari
Pre-condicions	Usuari estigui logiejat
Iniciat per	Usuari
Flux	
Post-condicions	Usuari pot efectuar una cerca

Usuari registrat cerca un producte d'un producte al E-Shop.
 Si troba un resultat mostra el producte, si no existeixen no mostra el llistat de productes.
 Si esborra la cerca realitzada, es mostren els productes sense el llista de cerca.

Identificador	CU-005
Nom	Realitzar una consulta
Prioritat	Alta
Descripció	Usuari efectua una consulta
Actors	Usuari
Pre-condicions	Usuari estigui logiejat
Iniciat per	Usuari
Flux	
Post-condicions	Usuari realitza una consulta a Gironaps

A les activitats que l'usuari estigui interessat, l'usuari té l'opció de demanar més informació per correu electrònic.

Degut a que Capacitor no té encara un plugin propi per a EMAILS, s'ha hagut de provar amb plugins de Cordova. Degut a problemes d'incompatibilitat que feien que molts altres mòduls deixessin de funcionar (app no compilava després), s'ha optat per no instal·lar aquest «feature» i tirar d'una opció simple amb un AllertControler.

Les proves s'han realitzat en la versió web, en els simuladors de iOS i d'Android i també en un dispositiu iPhone 6 Plus. En tots ha funcionat el login, register (generant-se els entries al backend ja sigui Firebase o Stripe), moure entre diferents tabs, realitzar compres i pagaments, poder canviar la foto del perfil i canviar el Darkmode, veure les comandes realitzades i desconnectar-se de l'aplicació.

Coses que no s'han pogut testejar degut a falta d'implementació són el fet de poder enviar correus directament clicant un botó a l'aplicació i implementar una cerca més completa.

Esperem que això es pugui acabar d'implementar en una futura versió de l'aplicació.

Capítol 6 - Problemes

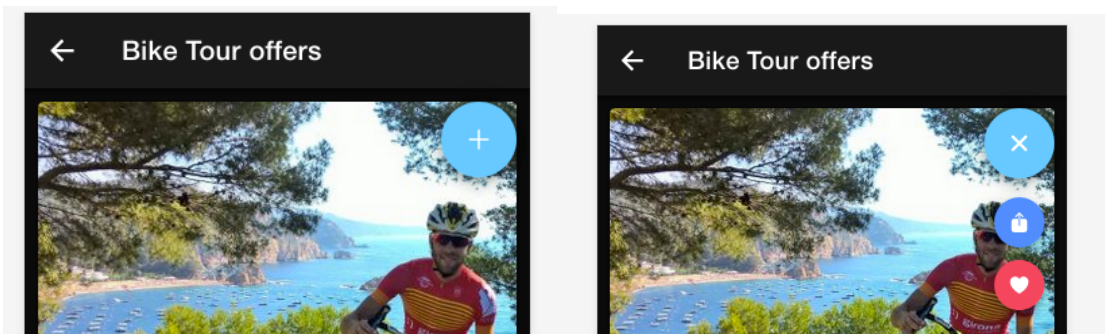
Un dels primer problemes que s'han tingut és la quantitat de temps que es perdia en fer canvis, compilar i poder veure els resultats.

En iniciar el projecte es va optar per fer servir Cordova. Apache Cordova és un *framework* de codi obert que permet als desenvolupadors web utilitzar el seu contingut HTML, CSS i JavaScript per crear una aplicació nativa per a diverses plataformes mòbils. S'han tingut problemes per veure els canvis realitzats en iOS simulator i un iPhone 6 Plus.

Es va optar de provar amb Capacitor i ben sovint es va veure que treballar en temps real, o sigui realitzar un canvi, compilar i veure el canvi reflectit en la versió web o el simulador va ser molt més efectiu que amb anterioritat amb Cordova.

Es pot saber més sobre Capacitor a la següent [web](#).

Una de les implementacions que es volia afegir en l'aplicació era l'opció de fer *SocialSharing*, o sigui, re-enviar a través de l'aplicació a través de Whatsapp un enllaç a altres contactes. Inicialment es va aconseguir de fer amb Cordova, però degut a les dificultats que es varen tenir per poder fer anar altres funcionalitats més necessàries, s'ha optat finalment de no implementar aquesta opció.



La idea era tenir un *FABUTTON* a la pàgina de detall de l'activitat a on poder enviar aquesta activitat concreta a un contacte via Whatsapp. També es va contemplar l'opció de poder afegir a un llista de favorits una activitat en concreta. Encara que es va aconseguir de fer anar el tema FAVORITES, s'ha descartat per dedicar més temps a la part de la creació del E-SHOP i tot el seu BACKEND. No es descarta afegir aquestes funcionalitat en una futura versió quan Capacitor tingui un plugin propi per el SOCIALSHARING. Respecte al favorits, està pendent d'incloure aquest opció en el perfil de l'usuari i veure un llistat amb els favorits.

Un altre problema que s'ha tingut es la no possibilitat de poder fer anar el EMAILCOMPOSER, que hauríem fet servir per a que un usuari contactés amb Girona Pro Sport a través de correu electrònic. En voler provar la funcionalitat es rebia un missatge tipus: `this._emailComposer.isAvailable() is not a function.`

```

✖ ▶ ERROR TypeError: this._emailComposer.isAvailable is not a function
    at BiketourDetailPage.sendInquiry (biketour-detail.page.ts:54)
    at BiketourDetailPage_Template_ion_button_click_61_listener (template.html:86)
    at executeListenerWithErrorHandling (core.js:15279)
    at wrapListenerIn_markDirtyAndPreventDefault (core.js:15317)
    at HTMLElement.<anonymous> (platform-browser.js:592)
    at ZoneDelegate.invokeTask (zone-evergreen.js:399)
    at Object.onInvokeTask (core.js:28578)
    at ZoneDelegate.invokeTask (zone-evergreen.js:398)
    at Zone.runTask (zone-evergreen.js:167)
    at ZoneTask.invokeTask [as invoke] (zone-evergreen.js:480)
    
```

Una alternativa que s'ha implementat és el fer servir *AllertControler*. Pot ser no ha sigut la solució més elegant, però creiem que farà la seva funció mentre es troba la manera de fer anar aquest funcionalitat.

En lo referent al BACKEND, els problemes que s'han tingut ha sigut a la hora de crear el PAYMENT INTENT a Stripe.

Per a realitzar aquestes accions, s'ha hagut d'implementar FUNCTIONS a Firebase.

Per crear això, dintre del projecte s'executa la comanda «`firebase init -project PROJECTNAME`» i es selecciona «FUNCTIONS».

```

? Which Firebase CLI features do you want to set up for this
, then Enter to confirm your choices. (Press <space> to sele
lection)
  ○ Database: Deploy Firebase Realtime Database Rules
  ○ Firestore: Deploy rules and create indexes for Firestore
>○ Functions: Configure and deploy Cloud Functions
  ○ Hosting: Configure and deploy Firebase Hosting sites
  ○ Storage: Deploy Cloud Storage security rules
  ○ Emulators: Set up local emulators for Firebase features
    
```

Un cop finalitzat tot, en el nostre projecte es veu una nova carpeta anomenada *Functions*, en la qual s'han d'implementar les funcions necessàries per a la creació d'usuaris i gestió de pagaments a Stripe.

Quan es volia realitzar el DEPLOYMENT van aparèixer molts errors a l'arxiu «index.ts» que està ubicat a Functions → src → index.ts

```

danieldominguez@MacBookPro functions % sudo firebase deploy --only functions
Password:
    
```

```

/Users/danieldominguez/Desktop/gironaps/functions/src/index.ts
11:19 error   Missing trailing comma                comma-dangle
14:1  error   This line has a length of 93. Maximum allowed is 80  max-len
14:81 warning  'context' is defined but never used  @typescript-eslint/no-unused-vars
15:1  error   Expected indentation of 2 spaces but found 4        indent
15:17 error   Strings must use doublequote           quotes
16:1  error   Expected indentation of 2 spaces but found 4        indent
17:1  error   Expected indentation of 4 spaces but found 6        indent
17:24 error   Missing trailing comma                comma-dangle
18:1  error   Expected indentation of 2 spaces but found 4        indent
19:1  error   Trailing spaces not allowed           no-trailing-spaces
20:1  error   Expected indentation of 2 spaces but found 4        indent
21:1  error   Trailing spaces not allowed           no-trailing-spaces
22:1  error   Expected indentation of 2 spaces but found 4        indent
22:50 error   Strings must use doublequote           quotes
23:1  error   Expected indentation of 2 spaces but found 4        indent
23:49 error   Strings must use doublequote           quotes
24:1  error   Trailing spaces not allowed           no-trailing-spaces
25:1  error   Expected indentation of 2 spaces but found 4        indent

```

Per solvatar això s'havia de realitzar un FIX abans del *deployment* amb la comanda «`sudo npx eslint src/index.ts --fix`».

Un cop fet això es varen arreglar els errors quedant només algun que altre *warning*.

```

danieldominguez@MacBookPro functions % npx eslint index.ts --fix
Oops! Something went wrong! :(

ESLint: 7.27.0

danieldominguez@MacBookPro functions % sudo npx eslint src/index.ts --fix
/Users/danieldominguez/Desktop/gironaps/functions/src/index.ts
 15:28 warning  'context' is defined but never used  @typescript-eslint/no-unused-vars

* 1 problem (0 errors, 1 warning)

```

0 errors
1 warning

Ara si el *deployment* funcionava.

```

danieldominguez@MacBookPro functions % sudo firebase deploy --only functions
Password:

=== Deploying to 'gironaprosport-45a9e'...

i deploying functions
Running command: npm --prefix "$RESOURCE_DIR" run lint
> lint
> eslint --ext .js,.ts .

/Users/danieldominguez/Desktop/gironaps/functions/src/index.ts
 15:28 warning  'context' is defined but never used  @typescript-eslint/no-unused-vars

* 1 problem (0 errors, 1 warning)

Running command: npm --prefix "$RESOURCE_DIR" run build
> build
> tsc

```

```

> build
> tsc

✓ functions: Finished running predeploy script.
i functions: ensuring required API cloudfunctions.googleapis.com is enabled...
i functions: ensuring required API cloudbuild.googleapis.com is enabled...
△ functions: missing required API cloudbuild.googleapis.com. Enabling now...
✓ functions: required API cloudfunctions.googleapis.com is enabled
✓ functions: required API cloudbuild.googleapis.com is enabled
i functions: preparing functions directory for uploading...
i functions: packaged functions (88.77 KB) for uploading
✓ functions: functions folder uploaded successfully
i functions: creating Node.js 14 function createStripeCustomer(us-central1)...
✓ functions[createStripeCustomer(us-central1)]: Successful create operation.

✓ Deploy complete!

Project Console: https://console.firebase.google.com/project/gironaprospor-45a9e/overview
danieldominguez@MacBookPro functions %
    
```

Amb això vam crear les funcions (functions) necessàries a Firebase.

Funció	Activador
createStripeCustomer	user.create
getCustomerOrders	HTTP Solicitud https://us-central1-gironaprospor-45a9e.cloudfunctions.net/getCustomerOrders
startPaymentIntent	HTTP Solicitud https://us-central1-gironaprospor-45a9e.cloudfunctions.net/startPaymentIntent
webhook	HTTP Solicitud https://us-central1-gironaprospor-45a9e.cloudfunctions.net/webhook

En quant al tema de pagaments amb Stripe, els problemes que s'han tingut han sigut amb CORS.

```

▶ You may test your Stripe.js integration over HTTP. However, live Stripe.js integrations must use HTTPS. (index):1
▶ POST https://us-central1-gironaprospor-45a9e.cloudfunctions.net/startPaymentIntent 500 zone-evergreen.js:1068
✖ ERROR Error: INTERNAL core.js:6210
    at new HttpsErrorImpl (index.esm.js:54)
    at _errorForResponse (index.esm.js:149)
    at Service.<anonymous> (index.esm.js:594)
    at step (tslib.es6.js:102)
    at Object.next (tslib.es6.js:83)
    at fulfilled (tslib.es6.js:73)
    at ZoneDelegate.invoke (zone-evergreen.js:364)
    at Object.onInvoke (core.js:28591)
    at ZoneDelegate.invoke (zone-evergreen.js:363)
    at Zone.run (zone-evergreen.js:123)
    
```

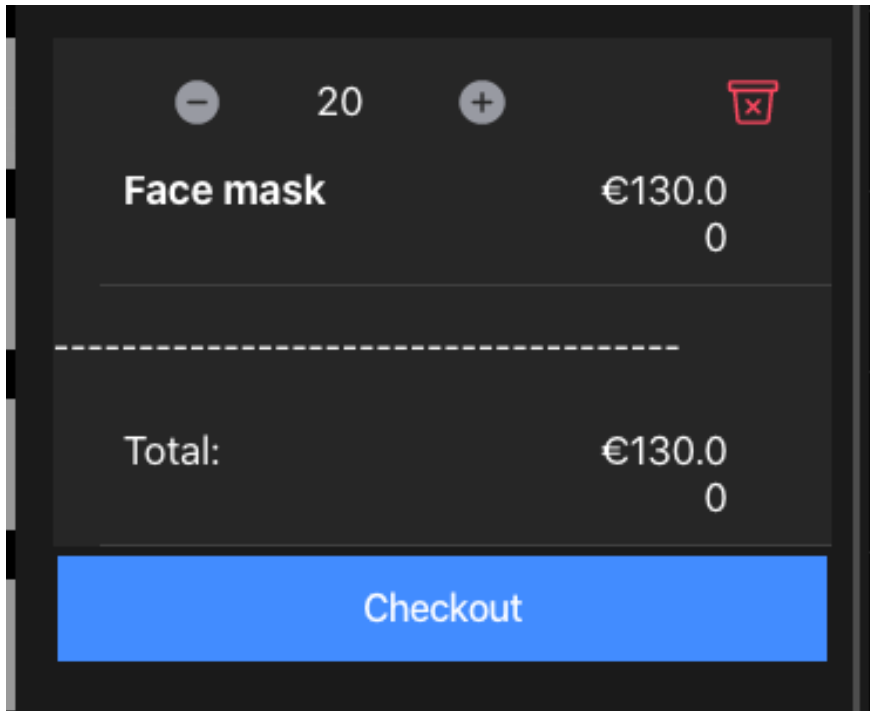
Instal·lant una extensió de Google Chrome, s'ha pogut evitar aquest error en forma de «bypass».

Després s'ha pogut observar que aquest problema apareixia només amb usuaris creats prèviament abans de la implementació de les FUNCTIONS a Firebase.

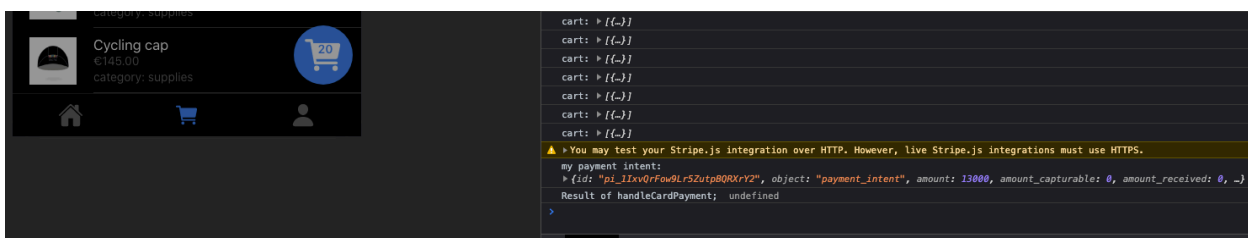
El que hem fet és eliminar un usuari existent creat abans de implementar las funcionalitats del Eshop al backend i crear-lo de nou en l'aplicació: dani@dani.com

		creacion	acceso	
patrick@gmail.com	✉	1 jun. 2021	2 jun. 2021	253g6mJV7KS8MmuQ5pbBWNHy...
dummy@dummy.com	✉	31 may. 2021	31 may. 2021	CCE3fLIHULW5X4MINSgp6YHQA...
dani@dani.com	✉	2 jun. 2021	2 jun. 2021	urERw6egk6NHKJRbYF6TS7GRnT...

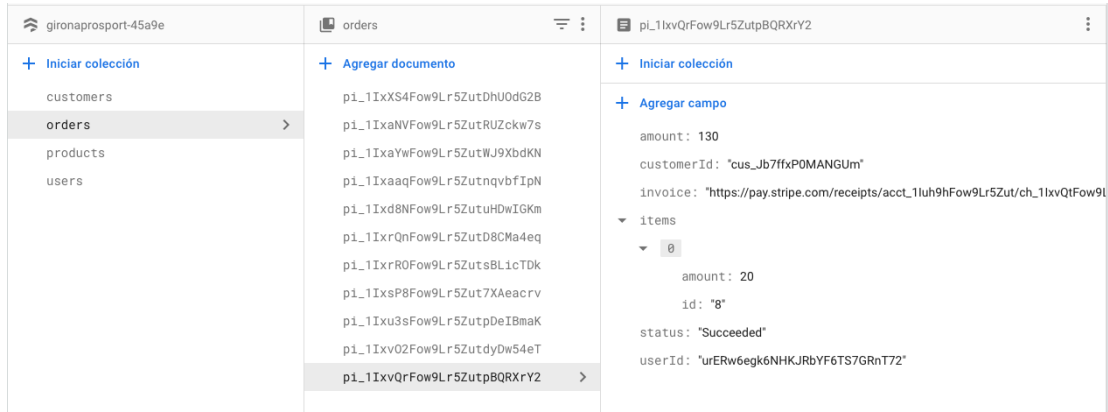
Realitzar una compra i veure si donava error o no.



No s'ha vist cap missatge d'error.



S'ha verificat al backend si la compra s'ha realitzat o no. Apareix una comanda amb 20 mascaretes, que es el test que hem realitzat.



Finalment mirant a Stripe a veure si apareix la comanda.

Payments Filter Export Create payment

All Succeeded Refunded Uncaptured

AMOUNT	DESCRIPTION	CUSTOMER	DATE
€130.00	Succeeded ✓ pi_1IxvQrFow9Lr5ZutpBQRXrY2	dani@dani.com	2 Jun, 16:52

PAYMENT TEST DATA pi_1

€130.00 EUR Succeeded ✓

Date	Customer	Payment method	Risk evaluation
2 Jun, 16:52	dani@dani.com	visa **** 4242	38 Normal

Timeline

- Payment succeeded 2 Jun 2021, 16:52
- Payment started 2 Jun 2021, 16:52

Payment details

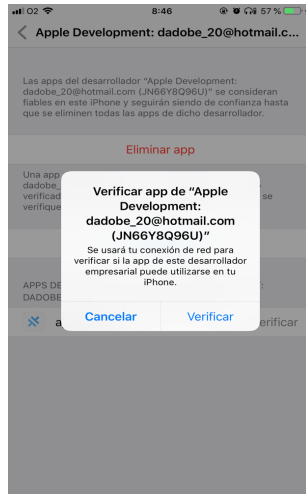
Statement descriptor	GIRONA PRO SPORT
Amount	€130.00
Fee	€4.02
Net	€125.98
Status	Succeeded
Description	No description Edit

Payment method

ID	pm_1Ixv0sFow9Lr5ZutgqyXKfCH	Owner	Daniel Domínguez
Number	**** 4242	Owner email	dani@dani.com
Fingerprint	KUDCowNm4GLXZ8SB	Address	Avenida Girona
Expires	02 / 2022		Girona, 33333, ES
Type	Visa credit card	Origin	United States 🇺🇸
Issuer	Stripe Payments UK Limited	CVC check	Passed ✓

Problem solved!!

Un problema que vam tenir en llençar l'aplicació per primera vegada amb Xcode va ser el no haver «verificat» l'app.



Després de realitzar la verificació es va poder veure com era la *app* en un *smartphone*.

Un nou problema va sorgir en voler fer anar l'aplicació en un REAL DEVICE com el iPhone 6 Plus després d'implementar les funcions de pagament amb Stripe. Després de fer el BUILD & RUN l'aplicació no carrega al dispositiu donant un TIME OUT.

```
2021-06-02 17:12:36.066045+0200 App[36248:4555150] DiskCookieStorage changing policy from
2 to 0, cookie file:
file:///private/var/mobile/Containers/Data/Application/88A55590-9A59-4745-ACAC-A4398ED8
1789/Library/Cookies/Cookies.binarycookies
Loading network plugin
2021-06-02 17:12:37.387524+0200 App[36248:4555150] CAPKeyboard: resize mode - native
⚡ Loading app at http://192.168.1.48:8101...
Reachable via Cellular
APP ACTIVE
⚡ WebView failed provisional navigation
⚡ Error: The request timed out.
2021-06-02 17:15:00.005716+0200 App[36248:4555150] Could not load IOSurface for time
string. Rendering locally instead.
```

El que s'ha pogut fer per fer anar l'app al dispositiu és:

- Reiniciar el dispositiu
- Mirar que estigui connectat a la mateixa xarxa que l'ordinador
- Tornar a compilar a Visual Studio Code: `ionic cap run ios -l --external`

```
danieldominguez@MacBookPro gironaps % ionic cap run ios -l --external
```

Una última cosa amb la qual hem tingut problemes ha sigut la possibilitat de testejar en un entorn Android, ja que després del «build» a Android Studio l'aplicació no s'executava donat el següent error.


```
06/02 11:00:39: Launching 'app' on Nexus 5 API 30.
Install successfully finished in 23 s 313 ms.
$ adb shell am start -n "com.uoc.gironaps/com.uoc.gironaps.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER
Error while executing: am start -n "com.uoc.gironaps/com.uoc.gironaps.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.uoc.gironaps/.MainActivity }
Error type 3
Error: Activity class {com.uoc.gironaps/com.uoc.gironaps.MainActivity} does not exist.

Error while Launching activity
```

Passes que s'han realitzat per arreglar aquest error han sigut:

- Un clean build
- Tancar Android Studio
- Reiniciar el portàtil
- Carregar el projecte al IDE i tornar a obrir-lo amb Android Studio
- Tornar a crear un nou Simulador, però aquest cop instal·lant també una versió d'Android anterior
- Tornar a muntar-ho tot: «build & run»

Amb això hem pogut veure com corria l'aplicació en una versió simulada d'un Nexus 5.

Capítol 7 - Conclusions

En aquest apartat voldríem afegir una aportació personal, m'és ben dit, com de diferent enfocaríem el projecte des del principi amb tot el que hem après.

En primer lloc realitzaríem proves a l'aplicació un cop tinguéssim el login implementat, ens referim a testejar l'aplicació amb els simuladors de iOS i Android.

Preparar l'entorn és una cosa, una altre de diferent es voler compilar i carregar el codi en Xcode / Android Studio i veure tot de missatges d'error. Dedicaríem un temps en anar corregint això des d'un principi per no anar tan estressats a la recta final de l'entrega.

La memòria també l'actualitzaríem més sovint, de manera que evitem perdre dies valuosos al final del projecte que podríem dedicar a millorar una mica el codi.

També dedicariem més temps al principi en fer les proves unitàries, per que hem vist lo útil que pot arribar a ser realitzar una bona quantitat d'aquestes proves.

Capítol 8 - Agraïments

Per fi la meua aventura en aquest Grau d'Enginyeria Informàtica arriba a la seva destinació final.

M'envaeix una mescla de satisfacció i de tristesa haver arribat a aquest punt, però quant miro tot el que he recorregut per arribar fins aquí no puc més que dir paraules d'agraïments a totes les persones que m'han acompanyat durant el trajecte.

Voldria agrair:

- Al meu Pare i a la seva dona Ana haver-me criat i donat una bases per a la meua vida.
- A la meua mare per haver-me donat suport a través de les converses telefòniques.
- Als companys que he tingut al llarg dels estudis, però voldria fer una especial menció a l'Ismael Fernández Molina, per haver sacrificat moltes tardes i cap de setmana per ajudar-me a entendre la matèria.
- Als meus germana Sandra i al meu germà Patrick per haver passat tants bons moments al pis compartit.
- Als meus tiets Nuria i Jürgen, per haver-me ajudat amb orientacions i bons consells en la vida laboral.
- Als professors de la UOC i en especial al meu tutor assignat al Treball Final de Grau, en Carles Sánchez Rosa per l'ajuda donada aquest últims mesos del Grau.
- I un agraïment molt especial a la meua parella Jana per haver-me motivat sempre que ho he necessitat i al meu fill Oskar per haver tingut la paciència suficient per totes les vegades que he estat ocupat amb projectes i/o assignatures de la UOC. Tampoc em voldria oblidar de Lajka, la nostra gossa, que m'ha fet companyia durant aquest mesos de HomeOffice a casa durant la pandèmia. Us estimo molt.

Capítol 9 - Bibliografia

Per a la realització d'aquest projecte hem fet servir varies fonts a internet, com:

- [1] OOP Coders. https://www.youtube.com/watch?v=Q1839NnaYMY&ab_channel=OOPCoders
- [2] Academind. https://www.youtube.com/watch?v=r2ga-iXS5i4&ab_channel=Academind
- [3] S. Grimm. Devdactic. How to build ionic shopping cart with Ionic 4. https://www.youtube.com/watch?v=ZFFVMBhJzVU&ab_channel=SimonGrimm
- [4] A. Petro. https://www.youtube.com/watch?v=vYBtWDiHsCU&ab_channel=AlexPetro
- [5] A. Kubów. Code with Ania Kubów. Make your own mock API (super simple). https://www.youtube.com/watch?v=FLnxgSZ0DG4&t=639s&ab_channel=CodewithAniaKub%C3%B3wCodewithAniaKub%C3%B3w
- [6] Fireship. Angular Test quick start. https://www.youtube.com/watch?v=BumgayeUC08&ab_channel=Fireship
- [7] Ionic Framework. <https://ionicframework.com/>
- [8] S. Grimm. Devdactic Code to success. <https://devdactic.com/>
- [9] . Developers LearnIT. Building an Ionic 3 Shopping Cart App with Firebase Realtime Database - Part 4 <http://www.developerslearnit.com/2018/10/building-ionic-3-shopping-cart-app-with-firebase-realtime-database-part-4.html>
- [10] S. Grimm. Ionic Academy. Firebase Basics [v5]. <https://ionicacademy.com/courses/firebase-basics/>
- [11] S. Grimm. Ionic Academy. Stripe Payments with Firebase Cloud Functions [v4]. <https://ionicacademy.com/courses/stripe-payments-with-firebase-cloud-functions/>
- [12] S. Grimm, Practical Ionic, en versió PDF.
- [13] A. Rathore. "Beginner's Guide to Ionic Angular Unit Testing (Part 1)". NAPPD. <https://enappd.com/blog/beginners-guide-to-ionic-angular-unit-testing-part-1/151/>
- [14] J. Morony. "How to Unit Test an Ionic/Angular Application". JOSH MORONY/. <https://www.joshmorony.com/how-to-unit-test-an-ionic-2-application/>
- [15] A. Volpe. "Test your components in Ionic 4". aVolpe Blog. <https://www.volpe.com.py/ionic/2018/12/07/test-ionic-4.html>
- [16] "Testing". <https://ionicframework.com/docs/angular/testing>