



Desplegament de la Infraestructura i Arquitectura, l'aplicació mòbil, i la Base de Dades

Índex...

Descripció general Sprint	3
Desplegament de la Infraestructura i Arquitectura	4
Descripció general	4
Explicació software/hardware utilitzat	4
Manual d'usuari	4
Conclusions	4
L'aplicació mòbil	5
Descripció general	5
Explicació software/hardware utilitzat	5
Manual d'usuari	5
Conclusions	4
Base de Dades	6
Descripció general	6
Explicació software/hardware utilitzat	6
Manual d'usuari	6
Conclusions	6



Descripció general Sprint

//Descripció global de les tasques realitzades i l'organització del grup. Part del projecte realitzada, estructuració del treball...

En aquest Sprint s'ha prioritzat, el funcionament de la Infraestructura i Arquitectura i l'Aplicació mòbil. L'equip ha prioritzar resoldre aquells requisits mínims que encara no estàvem implementats. El resultat ha estat bastant satisfactori, perquè creiem que hem complert amb totes les expectatives i que s'ha presentat un molt bon Sprint i per tant, projecte. En els següents apartats podeu trobar la informació detallada de cada secció.

Desplegament de la Infraestructura i Arquitectura

Descripció general

//Descripció més detallada i concreta d'aquesta part del treball.

En aquest Sprint s'ha prioritzat, el funcionament de la Infraestructura i Arquitectura. Hem decidit resoldre els problemes no solucionats en anteriors Sprint per tal de poder tractar tots els missatges que ha de rebre la Terminal. En el següent apartat, s'explica amb detalls les eliminacions i addicions de material dins les màquines del CRAAX.

Explicació software/hardware utilitzat

Tant el software com hardware és el mateix que en Sprints anteriors, en aquest cas, hem tingut un gran augment de la utilització del servei Docker, sobre el Cloud, Fog i Edge.

CANVIS REALITZATS:

1. No sabiem resoldre els problemes de connexions entre diferents dockers.

El que hem fet ha estat crear un sistema distribuït, entre el FOG (manager), i l'Edge (worker). S'han obert diferents ports (3000, 3003, 3516, entre d'altres, per destinar-los a la comunicació amb l'API, l'aplicació mòbil, els cotxes, l'aplicació web i la base de dades MONGO (ubicada en el Cloud).

2. Actualització de l'API que estava en el Cloud. Hem ficat l'última versió, ara permet la comunicació amb els cotxes (port 3003). Aquesta API s'ha pujat al Dockerhub (compiladori d'imatges docker), i posteriorment, hem creat el sistema distribuït a la màquina FOG.
3. Eliminació de tota la informació anterior en màquines FOG.

S'ha utilitzat l'eina docker swamp init, que ens inicia un enjambre de sistema distribuït, i que ens permet la connectivitat.

A la màquina Edge s'ha fet un join a l'enjambre, amb un Token que s'ha generat en el moment d'iniciar-lo. En aquest punt, ja tenim dues màquines en un mateix sistema distribuït. Creació d'una xarxa de tipus; overlay, per permetre canviar la xarxa per defecte per una pròpia. Amb aquest hem obert els ports necessaris a utilitzar.

4. Construcció d'un servei API amb la imatge que hem comentat anteriorment del dockerhub. S'ha utilitzat la xarxa creada i els ports indicats, mapejant-los a les màquines virtuals a les que corresponien.
5. Hem demanat al client tècnic que obri diferents ports dins la màquina Edge, això ens ha permès la connexió amb el cotxe i de l'aplicació quan estiguin dins la nostra xarxa.

Manual d'usuari

//Interaccions que pot realitzar l'usuari amb el projecte actual.

Aquesta part, engloba tot el que s'ha comentat anteriorment. L'usuari pot realitzar proves amb el projecte de manera que:

- Si es sol·licita un cotxe, es podrà veure el funcionament de l'Infraestructura.
- La connexió amb la base de dades Mongo.
- La connexió amb l'aplicació mòbil.
- Les màquines virtuals que simulen tota la creació d'elements.

Conclusions

Creiem que hem realitzat un gran Sprint, hem deixat en conclusió, la Infraestructura i Arquitectura acabada, el més important és que hem sabut distingir els diferents elements Cloud, Edge i Fog, i hem ubicat cada servei/element en el lloc adequat.

L'equip ha pensat que aquesta Infraestructura és la més adequada, ja que ens beneficia a llarg plaç, tenim millores de seguretat en totes les comunicacions (ja que estan encriptades), i utilitzar Docker ens serveix per mantindre un sistema d'eleccions intern, on si cau una màquina, el docker s'encarregarà d'assignar una nova màquina líder.



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Politècnica Superior d'Enginyeria
de Vilanova i la Geltrú

L'aplicació mòbil

Descripció general

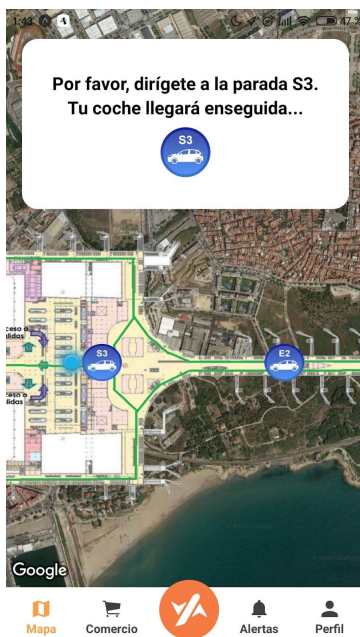
S'ha treballat conjuntament amb els responsables de la API per dur a terme les indicacions dels clients.

Explicació software/hardware utilitzat

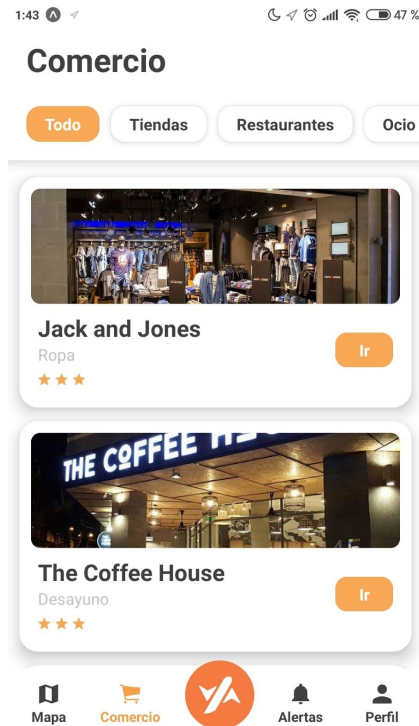
Hem utilitzat la API, els servidors del crax i la app

Manual d'usuari

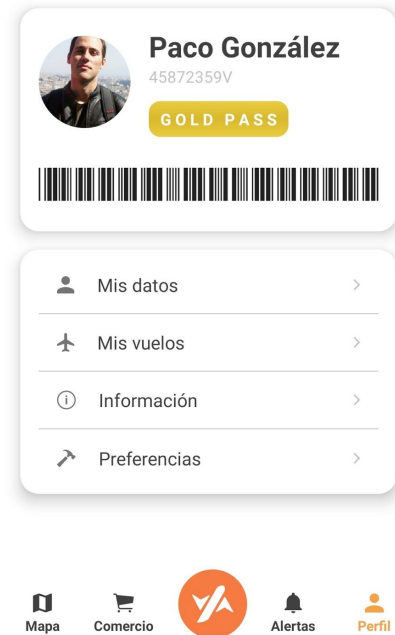
Ara l'usuari es mou per el mapa mitjançant la localització a través del punt d'accés wifi al que l'usuari està connectat. Aquest pot demanar un cotxe.



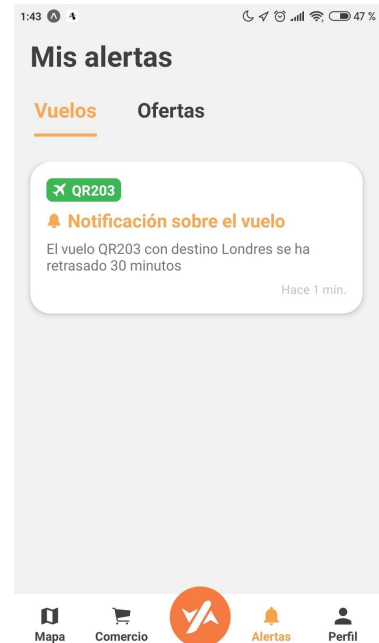
També les botigues y el perfil ja estàn connectats a la base de dades mitjançant la api



Mi perfil



i per últim ja reben notificacions en temps real, tant dels vols, com de ofertes d'interés per als usuaris de la app





Conclusions

Queda polir detalls per a l'últim sprint i la app ja estarà acabada.

Base de Dades

Descripció general

//Descripció més detallada i concreta d'aquesta part del treball.

En aquesta versió hem fet pocs canvis a la Base de Dades.

Boarding_passes:

El primer canvi és a la colecció `boarding_passes`, que s'ha eliminat i s'ha posat com una llista de `boarding_passes` en la colecció `flights`, es ha dir que totes dades de la col·lecció `boarding_passes` ara es una llista dins de la col·lecció `flights`.

Administrators:

El segon canvi es en la col·lecció d'administradors, la variable `job` indica qui tipus d'administrador es, s'han definit tres tipus:

- Jefe
- Controlador Vols
- Controlador Cotxes

```
{  
  id: "admin",  
  name: "Josep M. Roig",  
  email: "jmroig@gmail.com",  
  password: "admin",  
  job: "Jefe"  
}
```

Passengers:

Els canvis en els passatgers son 3 atributs nous:

url_image: Aquesta es la url de la foto d'un passatger si el vol.

type_user: Aquesta indica el tipus d'usuari que es, gold, premium, etc..

likes: Aquesta és una llista dels gustos que té un passatger.

```
{
  {
    {
      id: 56764564,
      name: "Antonio Delgado",
      url_image: "https://www.p.es",
      type_user: 0,
      email: "adelgado@gmail.com",
      birthdate: new Date("1993-01-26T16:00:00Z"),
      phone: "+34 9839565956",
      password: "lIMsd5NVgs",
      country: "Spain",
      City: "Barcelona",
      location_x:1,
      location_y:1,
      notices: [{
        id: "1",
        notification: "Tu vuelo A25 con direccion a sidney se ha retrasado 30 minutos",
        date: new Date("2020-01-05T16:00:00Z")
      }],
      likes: [{
        type: "Ropa"
      }]
    }
  }
}
```

Shops:

S'han afegit dos atributs nous.

url_image: Es la url de la imatge del tipus de tenda.

type: En aquesta es defineix quin tipus de tenda es:

- Restauració
- Roba
- DuttyFree
- Oci
- Altres

```
{
  {
    {
      id: "234537744289",
      name: "Jack and Jones",
      url_image: "https://marinedacity.com/store-photos/jack-jones.jpg",
      product_name: "Ropa",
      location_x: 102,
      location_y: 145,
      type: "Ropa",
      promotions: [ {
        offer: "10% en camisetes"
      },
      {
        offer: "2 x 1 en pantalones"
      }
    ]
  }
}
```

Explicació software/hardware utilitzat

Hem utilitzat mongoDB, pero ja hem explicat el seu funcionament en sprints anteriors.

Manual d'usuari

El manual d'usuari es el de mongoDB, ja explicat en sprints anteriors.

Conclusions

Hem aquests canvis per ajustarnos a les peticiones fetes per els clients, ara ja podem mostrar les targetes d'embarcament de cada passatger que tingui un vol, les tendes per diferents tipus, la seva imatge. Els avisos per als passatgers.



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Politècnica Superior d'Enginyeria
de Vilanova i la Geltrú