Desplegament del servei

Web i definició de protocols

Índex…

[**Descripció general Sprint**](#_8hnuvclbx2x5) **3**

[**Desplegament**](#_dj4y7a4oi0s0) **del servei web 4**

[Descripció general](#_464e7cwasjnm) 4

[Explicació software/hardware utilitzat](#_z6rvkcrqgag9) 4

[Manual d’usuari](#_gp1apcw159i) 4

[Conclusions](#_ld8qovzazto9) 4

**Definició de protocols 5**

[Descripció general](#_v7eno92lpqt1) 5

[Explicació/Justificació](#_by39bb8vw3bd) 5

[Conclusions](#_deqbv3qh8u5p) 5

# 

# 

# 

# 

# 

# 

# Descripció general Sprint

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

# En aquest sprint ens hem centrat principalment en la programació de la pàgina web a partir del disseny en Adobe XD que havíem presentat en el anterior sprint i en fer un prototip inicial de la API.

# En quant a la web es refereix, hem programat diferents parts tant de la web d’administrador com de la d’usuari. A la part d’usuari hem fet la pàgina principal, la secció de serveis y la página descriptiva de cada botiga o restaurant que hi ha al aeroport i a la part d’administrador el login principal i el menú on seleccionar el tipus de gestió que vols fer i on es podrà tancar la sessió actual.

Per altre banda la API programada en Node.js conté les funcions bàsiques per poder interconectar la base de dades que ens proporcionarà l’altre grup amb els serei web i la APP.

# 

# Desplegament del servei web

## Descripció general

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

El servei web està dividit en dos parts en la part d’usuari enfocada al consumidor on es poden realitzar consultes amb fins informatius com botigues, restaurants, mapes… I una altre part d’administrador per els operaris on es podran fer gestions de coches, passatgers i avions.

**Web Administrador:**

En iniciar la web ens encarem amb un login d'usuari i contrasenya per a poder accedir a les

següents funcions:

- **Alertes**: en aquest apartat informarem sobre les alertes de vols i dels cotxes autònoms dels

quals disposa en l'aeroport, així com de l'administració de tots dos. D'una banda estaran els

vols amb les seves respectives columnes, de les quals la més important seria l'estat de l'avió, ja que

en cas que estigui en mal estat hauríem de modificar-ho amb un petit comentari que

estaria amagat per un desplegament.

Més a baix trobarem l'administració dels cotxes autònoms, on disposarem dels

noms dels diferents cotxes, el seu estat i quina és l'acció que s'està realitzant en ell.

- **Mapa**: en entrar en Mapa trobarem el mapa de l'aeroport per a poder realitzar un filtrat dels

passatgers. Cal dir que només coneixerem la ubicació dels passatgers que tinguin descarregada

l'aplicació i que estiguin registrats en la base de dades. D'aquesta manera podem fer un filtrat

de la ubicació de diferents passatgers separats per aerolínies, per la mena de vol o si volem

una mica més genèric, per plantes.

- **Administració de cotxes**: en aquesta funció el que farem serà administrar tots els cotxes dels

quals disposem. És a dir, haurem de canviar l'estat de cada cotxe, informant així de si

està ocupat, lliure o avariat. En cas d'estar ocupat hem d'informar que passatgers estan

sent transportats.

**· Web Client:**

- **Pàgina principal:** simplement és la pàgina principal de la web, la qual ens informarà de tot el que

sigui necessari, així com també ens proporcionarà els diferents serveis que demandem.

A la cantonada dreta de la part de dalt tindrem: Serveis, Mapa, Accessos i contacte.

**a) Serveis**: aquest apartat conté els diferents serveis comercials dels quals disposa l'aeroport.

Estan classificats per la mena de comerç que són i una vegada entrant en un d'ells

ens proporciona el tipus de franquícies que hi ha així com la ubicació de cada botiga.

**b)** **Mapa**: no té una funció molt avançada, simplement ens mostrarà el mapa per plantes de l'aeroport.

**c) Accessos**: si cliquem en aquest enllaç ens portarà a una pestanya que contindrà un requadre. Aquest

requadre que ens ofereix els recursos de Google Maps, ens demanarà una ubicació inicial i una ubicació final, la qual haurem de triar en el desplegament. Una vegada tot col·locat

correctament ens mostrarà el mapa amb la ruta en color i la ruta escrita perquè ho entenguem millor.

Tornem a la pàgina principal. En aquest cas podem consultar els vols que vulguem. Podem

consultar un vol en concret o tots els vols que arribin o surtin del nostre aeroport. Si

continuem baixant, trobarem apartats informatius sobretot, com el mapa, els cotxes

autònoms, una altra pestanya de com arribar...

## 

## 

## 

## 

## 

## Explicació software/hardware utilitzat

Per programar el servei web de l’aeroport hem fet servir diferents entorns de programació, llenguatges de programació, frameworks i serveis de pàgines web.

Per picar codi hem utilitzat principalment dos editors de text bastant coneguts que són Sublime Text y Visual Studio Code.

En quant a llenguatges de programació i frameworks per fer l’estructura bàsica de la pàgina hem fet servir HTML, CSS, Javascript i Bootstrap 4 què és un framework CSS desenvolupat per Twitter que permet donar forma a un lloc web mitjançant llibreries CSS que inclouen tipografies, botons, quadres, menús i altres elements que poden ser utilitzats en qualsevol lloc web.

A més a més també hem fet ús de diversos widgets, icones i fotografies gratuites. Entre les quals es troben :

**-W3.CSS Icons:** Llibreria d’icones gratuïts de W3Schools.

**-Ionic icons:** Pàgina web d’icones.

**-Flaticons:**Pàgina que ofereix fotos en format png.

**-Pexels**:Servei que ofereix fotos de stock.

**-Weatherwidget:** Widget responsive sobre el clima en temps real de qualsevol ubicació.

## Manual d’usuari

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

## Conclusions

Creiem que com a primera aproximació ens ha servit per a familiaritzar-nos amb l'ús de les APIs al món real, a la espera de poder connectar-la amb la base de dades i APP reals del grup B2.

# 

# 

# Desplegament de la API

## Descripció general

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

## Explicació software/hardware utilitzat

Per realitzar la API hem utilitzat diferentes eines que són les següents:

- **Node.js**: Entorn que treballa en temps real d'execució, de codi obert, multiplataforma

que permet al programador crear tota classe d'eines del costat servidor (BackEnd) i

aplicacions en Javascript.

Això ens permetrà crear una API REST que representi les operacions que podem tenir en una aplicació Web i realitzar operacions sobre qualsevol repositori de dades (Base de dades, fitxers o qualsevol altre).

- **Fitxer package.json**: Es crea en iniciar el projecte (npm init --yes) i conté totes les

dependències que necessitarem, les quals instal·larem conforme anem necessitant i es guardaran aquí.

- **Servidor Express JS**: Framework de Node, crea l'estructura del servidor i ens permet

escriure codi de manera senzilla.

- **Body-Parser**: Mòdul que ens permet convertir les dades que ens arriben en les peticions al servidor, en objectes amb format JSON.

- **Mongoose**: Mòdul que ens proveeix mètodes i funcionalitats per a treballar millor amb MongoDB.

-**Morgan**: Mòdul que ens permet veure per consola les peticions que arriben.

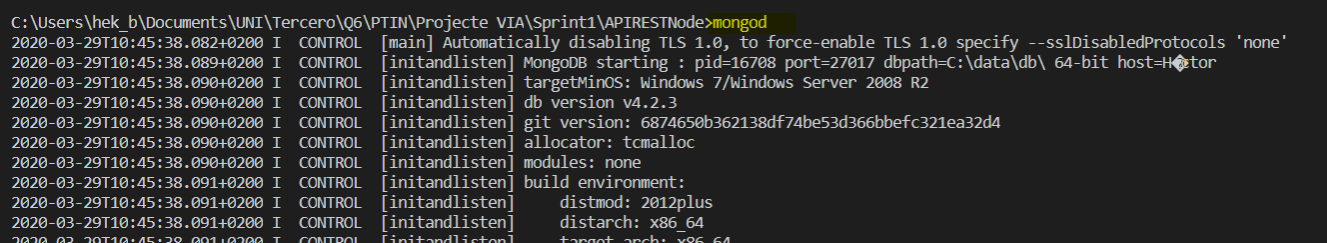
-**Nodemon**: Mòdul que farà que amb qualsevol canvi en el servidor, aquest es reiniciï (fa com una compilació), a més de que mostra per consola totes les peticions HTTP (GET, POST, PUT, DELETE…) que rep el servidor (200, 404, 500, etc.)

- **POSTMAN** és un programari que simula peticions a la API de rutes, des d'una aplicació web/mòbil. Ens cap afegir capçalera en Extensió per a Firefox: Content-Type: application/json

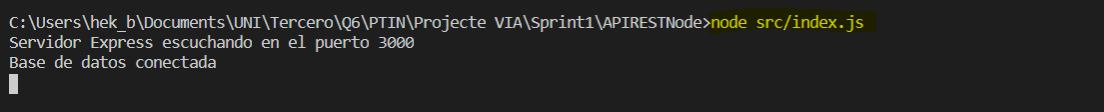
## Manual d’usuari

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

En aquest primer prototip, l’usuari pot llistar, afegir, actualitzar i eliminar usuaris, així com llistar, afegir i eliminar vols, tot això simplement tenint corrent el servidor i una base de dades MongoDB. En aquest cas, gràcies al mòdul Mongoose, em connectat el servidor amb una base de dades en local per a fer proves de peticions. L’usuari simplement haurà d’obrir un terminal i inserir la comanda *mongod:*



I des d’un altre terminal, arrancar el servidor Express:



Si l’usuari havia arrencat primer la base de dades, apareixerà el missatge de Base de dades connectada, si l’arrenca després d’arrencar el servidor Express, llavors el missatge apareixerà en el moment en què aquesta es connecti.

Un cop està tot arrencat, l’usuari ja pot fer proves amb les dades disponibles. En aquesta primera aproximació del prototip, hem creat dos models de dades, un per dades d’usuaris (gestors) i un altre per dades dels vols. Els paràmetres que es llisten per cadascun dels grups es reduït per el moment, però es podrà ampliar de cara a següents Sprints.

D’un usuari es tindran les següents dades:

* Nom
* Cognom
* Rol [Usuari / Administrador]

D’un vol, per altre costat, es tindran les següents dades:

* Número de vol
* Origen
* Destí
* Aerolinea (Només disponibles les aerolínies que actualment fan ús de l’aeroport) [Ryanair / QATAR Airways / Vilanova Airlines / Vueling / Iberia]
* Estat de vol [En hora / Retrassat / Cancel·lat]

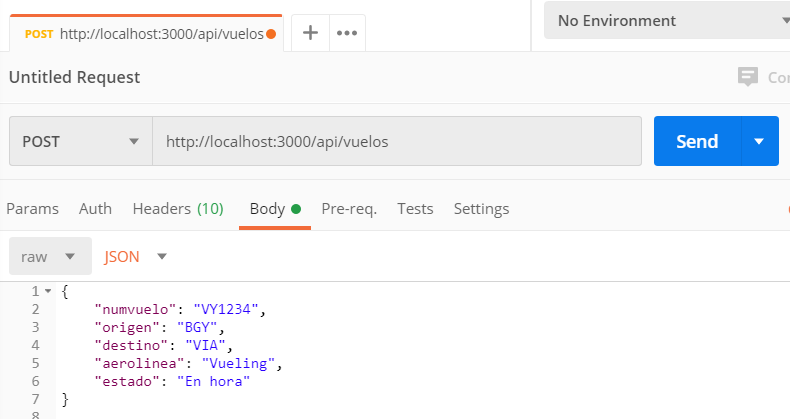
En els dos casos, pels paràmetres que tenim opcions definides, només veurem alguna d’aquestes opcions, ja que a l’hora afegir noves dades, si en aquests camps s’intenta afegir una opció diferent, donarà error.

A més, cada cop que s’afegeix una nova dada (usuari o vol) es genera un ID únic, i amb aquest ID es amb al que podrem modificar o esborrar aquest camp.

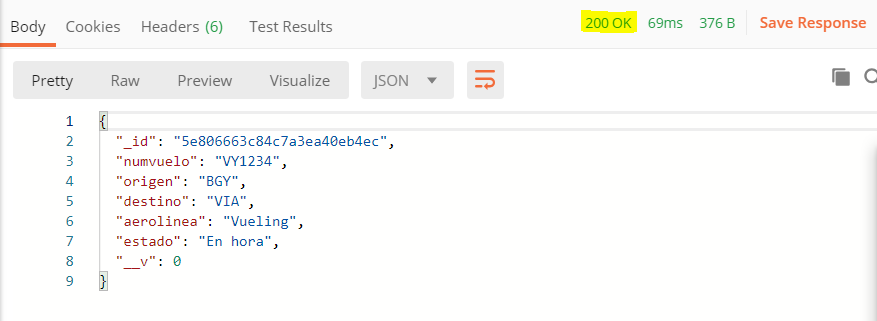
El servidor Express utilitzarà el port 3000, a no ser que hi hagi un port definit, i pel moment, l’usuari podrà accedir-hi a les dades, simplement entrant a un navegador i escrivint l’enllaç [http://localhost:3000/api/**users**](http://localhost:3000/users)ó [http://localhost:3000/api/**vuelos**](http://localhost:3000/vuelos) respectivament, pel moment no tenim definida la ruta arrel (<http://localhost:3000>) que permet-hi accedir des d’aquí als dos conjunts de dades (i als que puguin haver-hi pròximament), però es farà en una propera implementació. Als dos enllaços també es pot accedir-hi sense posar la paraula api a l’enllaç i les dades es mostraràn igualment, però per conveni s’acostuma a utilitzar i per això hem afegit aquesta opció.

Des del navegador l’usuari podrà llistar (corresponent a una petició HTTP GET), però no podrà fer res més, qualsevol inserció, modificació o eliminació (corresponents a peticions HTTP POST, PUT i DELETE), l’haurà de fer des del software POSTMAN, un cop afegida la capçalera indicada i el tipus de dades.

Per exemple, l’usuari administrador vol afegir un nou vol:



un cop envia la dada:

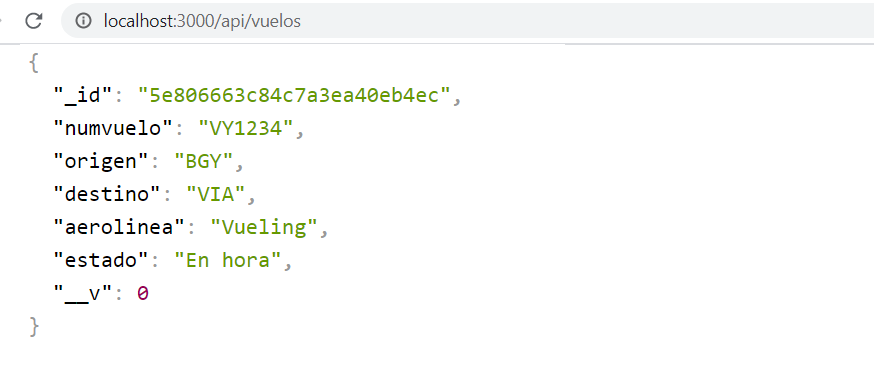


Al peu de POSTMAN veiem que la dada s’ha inserit bé (codi 200 HTTP).

També ens ho mostrarà el nostre Servidor (gràcies a tenir el mòdul Nodemon):

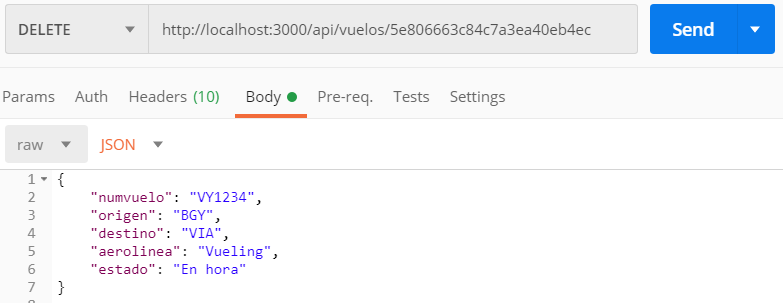


I ara podem llistar la nova dada des del navegador, o fent una petició GET des del mateix POSTMAN:



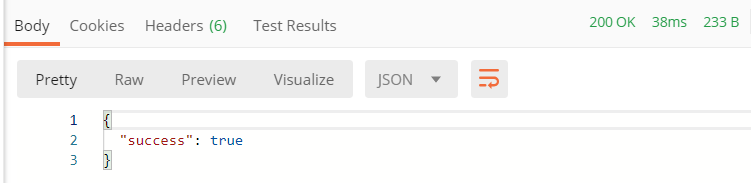
En aquest cas, podem veure des del navegador la dada inserida amb els seus paràmetres i la ID generada.

Ara que tenim la ID, des de POSTMAN es pot fer una modificació o eliminació d’aquesta dada (només capturarem una de les dues ja que el procés es identic):

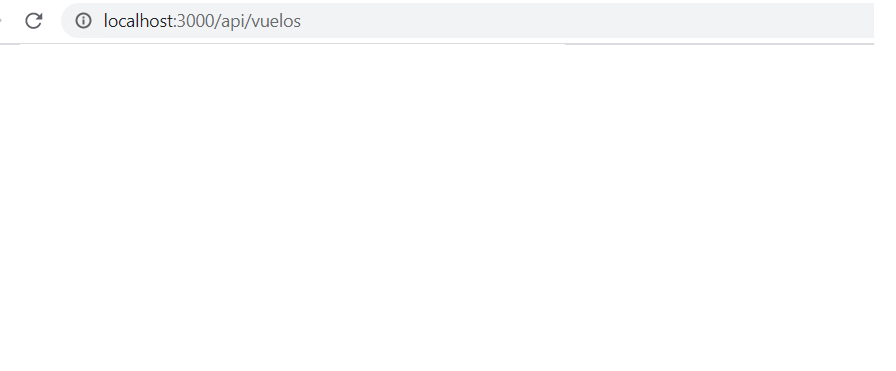


En aquest cas es una eliminació (petició DELETE), hem de fixar-nos que a l’enllaç, s’afegit la ID, per a que el servidor pugui saber quina dada s’eliminarà.

Un cop enviada veiem el següent:

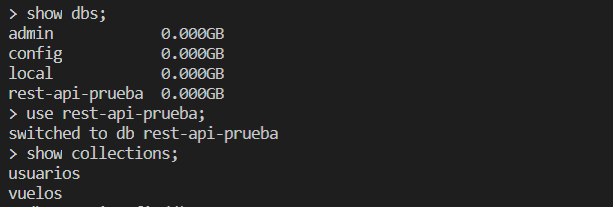


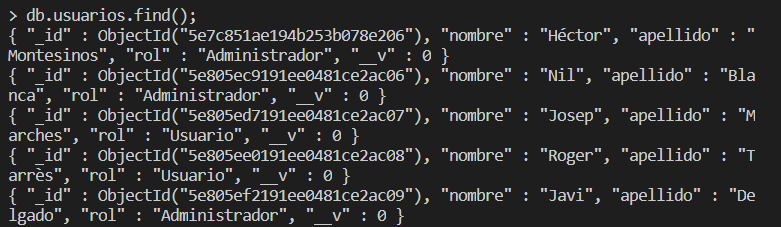
La dada s’ha eliminat correctament, i ara al fer un visionat dels vols, veiem que ja no apareix aquest:

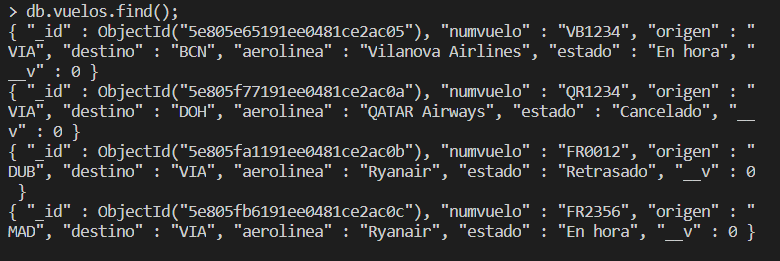


A part d’això, com a administradors podem fer un seguiment de les dades desde la propia base de dades. Hem afegit uns quants usuaris i uns quants vols com s’ha explicat a dalt i hem obert una altra terminal (tot això amb el servidor Express i la base de dades arrencats, es clar).

En aquest cas ens connectem a la base de dades mitjançant la comanda **mongo** i amb algunes comandes típiques del SGBD MongoDB, podem llistar les bases de dades que tenim actualment, els tipus de dades i les pròpies dades:







Conclusions

Creiem que com a primera aproximació ens ha servit per a familiaritzar-nos amb l'ús de les APIs al món real, a la espera de poder connectar-la amb la base de dades i APP reals del grup B2.

# Definició de protocols

## Descripció general

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

Tant el login, com les peticions a les bases de dades per a la ubicació dels clients i dels cotxes, es fan mitjançant protocols HTTPS al servidor, ja que es tracten de peticions com a POST, GET, PUT... De tot això s'encarregarà la API que ens connecta la web amb el que he dit abans, amb la base de dades, així com també ho farà amb l'aplicació de mòbil de cada client.

Ressaltar també que en la web de client, l'apartat de com arribar serà també una petició al servidor de Google, per tant tornarem a tenir el protocol HTTPS.

En resum, el protocol que s'usa sobretot és l'HTTPS, el qual usa un servei orientat a connexió

(TCP).

## Explicació/Justificació

//Explicació de la raó per la qual s’utilitzaran els protocols definits

## Conclusions

//\*\*Cal afegir un manual o be un link amb documentació dels protocols utilitzats.