

# ***Mobilitat sostenible: Electrike***

## ***Sprint 3***

[GitHub](#) - [Taiga](#) - [Project Record](#) - [Trello](#)



Cognoms	Nom	Roles	Correu electrònic		
			Google Drive	Taiga	GitHub
Asenjo Carvajal	Victor	Public relations	victor.asenjo@estudiantat.upc.edu	victorasenj@gmail.com	
Balaer Morales	Eloi	Scrum Master	eloi.balaer@estudiantat.upc.edu	eloix2@gmail.com	
Ni	Peilin	Dev. team Member	peilin.ni@estudiantat.upc.edu		
De La Varga Antoja	Ferran	Dev. team Member	ferran.de.la.varga@estudiantat.upc.edu		ferrandelavargaantoja@gmail.com
Dumitru Maroz	Alexandru	GitHub Master	alexandru.dumitru@estudiantat.upc.edu		alexandru666@outlook.es
Coll Ribas	Xavier	Dev. team Member	xavier.coll.ribas@estudiantat.upc.edu		xaviercollr@gmail.com
Rodriguez Rubio	Alvaro	Dev. team Member	alvaro.rodriguez.rubio@estudiantat.upc.edu		alvaro.alumne.tcn@gmail.com

Grau en Enginyeria Informàtica

Projecte d'enginyeria del software - Grup 21

Curs 2021-22 Quadrimestre de Primavera

"Primero resuelve el problema.  
Entonces, escribe el código"  
-- *John Johnson*

# Índex

<b>Introducció</b>	<b>1</b>
Què s'ha fet?	1
Sprint master report	3
Descripció individual de treball	5
Alexandru Dumitru Maroz	5
Álvaro Rodríguez Rubio	5
Eloi Balaer Morales	5
Ferran de la Varga Antoja	6
Peilin Ni	7
Víctor Asenjo Carvajal	7
Xavier Coll Ribas	8
<b>Requisits</b>	<b>10</b>
“Not” List - Actualitzada	10
Product Backlog	12
Requisits no funcionals - Plantilla Volere	15
10a. Requisits d'Aparença	15
11a. Requisits de Facilitat d'Utilització	15
11b. Requisits de Personalització i Internacionalització	16
11e. Requisits d'Accessibilitat	16
12a. Requisits de Velocitat i Latència	16
12c. Requisits de Precisió o Exactitud	17
12d. Requisits de Confiabilitat i Disponibilitat	17
12e. Requisits de Fortalesa o Tolerància a Fallades	17
13b. Requisits d'Interfície amb Sistemes Adjacents	18
14c. Requisits d'Adaptabilitat	18
15a. Requisits d'Accés	18
15c. Requisits de Privacitat	19
Gestió Taiga	20
Funcionalitats transversals	21
Serveis de tercers	25
Comunicació entre equips	25
Funcionalitats pactades	26
<b>Cerimònia Agile</b>	<b>28</b>
Report on the sprint planning, review and retrospective meetings	28
Sprint Planning	28
Sprint Review	30
Sprint 1 review	30
Sprint 2 review	31
Sprint Retrospective	32
Sprint 1 Retrospective	32

Sprint 2 Retrospective	33
Sprint 3 Retrospective	34
Updated release and iteration burndown charts and velocity chart	35
<b>Metodologia</b>	<b>38</b>
Visió global	38
Gestió de l'equip	39
Gestió projecte	40
Gestió repositori	41
Comunicació dins l'equip	42
Gestió de la qualitat	43
Estratègia de proves	45
Gestió de configuracions	46
Interacció amb companys	47
Gestió dels bugs	48
Tractament RNFs	49
<b>Descripció tecnològica</b>	<b>50</b>
Concepció global de l'arquitectura	50
Altres diagrames d'arquitectura	51
Patrons de disseny aplicats	52
MVC	52
Servicelocator	52
Singleton Pattern	54
Models de dades (UML)	55
Altres aspectes tecnològics	58
Instrumentació	58
Nombre de servidors	58
Configuració servidors	58
Nombre de BDs	59
Versionat de BDs	59
Nombre de llenguatges	60
APIs	60
API oferida	60
API interna	60
Nombre d'APIs externes	61
Consum servei	61
Subministrament servei	61
Consum de dades obertes	61
Modificació de llibreries lliures	62
Utilització d'eines de desenvolupament	62
Ús de Frameworks	62
Integració Continua	63
Desplegament (deployment)	63

Configuració del servidor	63
Firewall	65
Emmagatzematge	65
MongoDB	66
Express .js	69
Proxy CORS	69
<b>Referències</b>	<b>70</b>
<b>ANNEX</b>	<b>71</b>
<b>USUARIS</b>	<b>76</b>
Inserir un usuari	76
Eliminar un usuari	76
Consultar la informació d'un usuari	77
Consultar l'idioma de l'aplicació d'un usuari	78
Canviar l'idioma de l'aplicació d'un usuari	78
Canviar el CO2 produït per un usuari	79
Canviar els KM conduïts per un usuari	79
Canviar el contador de cotxes d'un usuari	80
Canviar el contador de rutes d'un usuari	80
Comprovar si existeix un usuari	80
Obtenir el número de notifikacions d'un usuari	81
Canviar el número de notifikacions d'un usuari	81
<b>COTXES</b>	<b>81</b>
Obtenir tots els cotxes per defecte del sistema	82
Obtenir les marques de cotxes del nostre sistema.	82
Obtenir models de cotxes d'una marca.	83
Obtenir informació d'un vehicle	83
<b>CARREGADORS ELÈCTRICS</b>	<b>83</b>
Obtenir carregadors elèctrics de Catalunya.	84
Obtenir informació d'un carregador.	85
Obtenir informació d'endolls.	85
Obtenir carregadors elèctrics de Barcelona.	87
<b>BICINGS</b>	<b>87</b>
Obtenir tots els punts Bicing per defecte del sistema	88
Obtenir informació d'un punt bicing.	88
Obtenir informació d'un punt bicing donat un carrer.	90
Obtenir informació de l'estat d'un punt Bicing.	90
<b>TROFEUS</b>	<b>90</b>
Obtenir tots els trofeus d'un usuari.	91
Posar com a obtingut un trofeu d'un usuari.	92
Posar com a no obtingut un trofeu d'un usuari.	92
<b>NOTIFICACIONS</b>	<b>92</b>

Obtenir totes les notificacions d'un usuari.	93
Afegir una notificació a un usuari.	93
Esborrar una notificació d'un usuari.	94
Desactivar una notificació d'un usuari.	94
Activar una notificació d'un usuari.	95
OCUPACIÓ	95
Obtenir l'ocupació d'un punt de recàrrega.	96
PUNTS BICING PREFERITS	97
Afegir un punt Bicing preferit a un usuari.	97
Eliminar un punt Bicing preferit d'un usuari.	97
Obtenir tots els punts Bicing preferits d'un usuari	97
CARREGADORS DE VEHICLES ELÈCTRICS PREFERITS	98
Afegir un carregador preferit a un usuari.	99
Eliminar un carregador preferit d'un usuari.	99
Obtenir tots els carregadors preferits d'un usuari	100
VEHICLES DELS USUARIS	100
Afegir un vehicle propi a un usuari.	101
Eliminar un vehicle elèctric d'un usuari.	101
Obtenir tots els carregadors preferits d'un usuari	102
DOCUMENTACIÓ SERVEI A BUILD GREEN	103
Documentació API Happy Lungs	106

# 1. Introducció

## 1.1. Què s'ha fet?

Aquest és el tercer i últim sprint en la qual l'equip desenvolupa parts addicionals pels sistemes que ja hi havien sigut desenvolupats en els anteriors sprints, a més de millores per l'usuari i incentivar a utilitzar l'aplicació.

Primer, com als anteriors sprints, hem escollit les tasques a fer per aquest sprint, les quals eren les que queden dels anteriors, a més s'han repartit per tal d'equilibrar la quantitat de treball de cadascun dels integrants del grup. Hem utilitzat el mateix mètode que en l'sprint anterior pel tracking, ja que ha sigut el que ens ha donat millors resultats.

Després vam posar-nos a desenvolupar les tasques. Per a aquesta entrega prevista ens hem compromès a fer una sèrie d'històries d'usuari i tasques, llistat que exposarem en l'apartat de Cerimònia Àgil i de les quals hem aconseguit enllestir les següents:

- Rutes (optimitzacions i millores per l'usuari)
- Ruta amb menys contaminació
- Guanyar trofeu
- Consultar trofeu
- Compartir ubicació
- Actualitzar text amb Multi idiomes
- Ocupació (millores per l'usabilitat)
- Notificacions programades (desglosada de activar notificacions)
- Desactivar notificacions programades (desglosada de desactivar activar notificacions)

- Ràpid aprenentatge (tutorial d'ús)
- Actualitzacions (errors que trobem d'històries anteriors o petites tasques pendents que no afecten en gran mesura a l'aplicació com migrar la base de dades a AWS)

També donarem per fetes les que són requisits no funcionals, ja que es finalitza l'aplicació:

- Càlcul ràpid de rutes
- Facilitat d'ús
- Aplicació fluida
- Ràpid aprenentatge



## 1.2. Sprint master report

Scrum Master: Eloi Balaer Morales

Hem arribat al tercer i últim Sprint d'aquest projecte, i ja tenim implementades totes les funcionalitats i característiques que havíem previst per a la nostra aplicació. Al final del segon Sprint vam deixar pràcticament totes les històries d'usuari tancades, a excepció d'unes poques, ja que havíem planificat que aquest últim Sprint hi hagués el mínim de modificacions possibles a causa del poc temps i exàmens i entregues finals. Respecte a la distribució de l'equip, ha estat la mateixa, 2 persones a la capa de presentació, 3 al domini, i 1 al back-end, i l'Alexandru que ha estat a tots els equips com a fullstack.

Al començament, vam fer l'Sprint planning, on entre tots els membres vam escollir com es repartirien les històries d'usuari restants a la vegada que fèiem les seves estimacions de temps. El nostre objectiu de l'Sprint era, no només acabar l'aplicació, sinó que acabar-la una setmana i mitja abans per poder corregir bugs que sortissin amb el temps. Aquest objectiu, l'hem complert, exceptuant unes poques històries d'usuari que estaven lligades a l'API de l'altre grup, que les vam haver d'acabar més tard per uns problemes de comunicació.

Pel que fa a la metodologia de treball i el treball individual, no ha canviat respecte a l'Sprint anterior, cadascú ha desenvolupat les seves tasques, i s'ha comunicat amb els altres membres per Discord en cas que hagués de connectar codi o solucionar algun dubte.

Pel que fa a les sessions presencials, a mitjans de classe, posàvem les tasques que havíem fet, en comú al Taiga. Miràvem com anaven les històries d'usuari, les tasques i en què estàvem treballant en aquell moment. Això feia que sabéssim en tot moment com anàvem. També, aprofitàvem aquella hora per fer les coses més "col·laboratives", com ara fer els merges, acabar de definir algunes tasques, preguntar als professors diversos dubtes...

Per finalitzar aquest últim Sprint, dir que hi ha hagut petits problemes de comunicació, però ja han estat solucionats. Ara tot el grup està molt satisfet amb el treball realitzat, com va quedar la presentació tècnica i, el resultat final. Més endavant ens quedarà considerar si en un futur pròxim publiquem l'aplicació amb ajuda d'alguna entitat.

### 1.3. Descripció individual de treball

Alexandru Dumitru Maroz

Durant aquest últim Sprint vaig tenir relativament poques tasques, en comparació a les altres iteracions. No obstant això, la dedicació més gran que he tingut al llarg d'aquest Sprint va anar resolent bugs o codi que no arribava a funcionar del tot bé. A més, vaig poder ajudar en unes certes tasques a diversos integrants de l'equip.

Respecte a les tasques més importants que tenia assignada va ser la migració del servidor, encara que al principi pensem que ens portaria diversos problemes, el canvi va resultar relativament senzill i ràpid.

Álvaro Rodríguez Rubio

Durant aquest Sprint m'he dedicat a fer els càlculs necessaris pels diversos trofeus de l'aplicació i la lògica necessària per guardar-los i mostrar-los a l'usuari. A més com la meva tasca va ser més senzilla del que esperàvem i, per tant, tènia més temps, així que em vaig dedicar a l'apartat de trofeus de la interfície a partir del que ja havien fet els meus companys d'aquest grup. Vaig editar les imatges necessàries i crear una alerta per quan es guanya un trofeu.

Després d'acabar aquest apartat vaig revisar les traduccions i vaig afegir algunes que faltaven pels nous apartats que creaven, ja que Peilin em va deixar encarregada aquesta tasca quan ja estàvem acabant l'Sprint, ja que la majoria d'aquesta feina estava feta i ella necessitava fer les rutes ecològiques, ja que no havia pogut perquè no teníem l'api dels companys tal com ens feia falta.

Finalment, em vaig dedicar a solucionar errors i bugs juntament amb l'Eloi.

Eloi Balaer Morales

En aquest tercer i últim Sprint m'he dedicat a la realització d'aspectes relacionats amb la capa de presentació, inicialment vaig canviar el funcionament dels gràfics d'ocupació perquè mostressin les seves dades en funció del dia de la setmana que escollís l'usuari, i vaig canviar l'estètica de la vista, ja que no ens agradava gaire.

Pràcticament a la vegada, vaig començar a editar la vista d'info ruta, que havia fet a l'Sprint anterior, per afegir les variables on es mostraria la distància i el temps que trigaria a fer-se una ruta cada vegada que l'usuari escollís una ruta. Aquesta tasca va tenir la seva dificultat, perquè vaig haver de fer refactor de moltes parts de la classe per aconseguir que s'actualitzés la informació en els moments que volíem. A més per a aquesta part vaig treballar amb Pei Lin quan necessitava alguna de les funcions que oferia de domain.

Mentre duia a terme aquesta tasca, també vaig crear la base de la pàgina de trofeus, on només es veien caps buides on anirien els trofeus que més tard va crear l'Álvaro. Juntament amb l'Álvaro vaig ajudar a mostrar els pop-ups de trofeus i també vam corregir bastants bugs relacionats principalment amb presentació, amb l'ajuda d'Alexandru.

Cap al final de l'Sprint m'he dedicat a arreglar bugs, coses que no es veien prou bé, he afegit millores per a la fàcil realització d'altres tasques, com podria ser afegir unes fletxes per a canviar de vehicle. Finalment, vaig connectar la capa de presentació amb l'última de les rutes, l'ecològica, que havien fet Pei Lin i Alexandru.

## Ferran de la Varga Antoja

Aquest Sprint ha sigut, en diferència, l'Sprint que he dedicat més hores. M'he encarregat de les notificacions programades, és a dir, que l'usuari pugui programar que li surtin notificacions certs dies de la setmana informant sobre l'estat de certs punts de càrrega (si estan lliures o ocupats).

Al principi vaig estar desenvolupant totes les funcions de la classe de notificacions que s'encarregaven d'organitzar-les i programar-les en el mòbil perquè sortissin en un futur. Seguidament, vaig fer les funcions de domini que cridaven a aquestes i retocaven algunes coses si era necessari. Després vaig estar treballant amb en Víctor, que va dissenyar les vistes i cridava a aquestes funcions de domini.

Finalment, un cop testejat el sistema de notificacions locals, vaig treballar amb en Xavi per pujar les notificacions a la base de dades, de tal manera que aquestes notificacions programades ara es guarden conjuntament amb les dades de l'usuari i si l'usuari fa un inici de sessió a l'aplicació, es descarreguen i es configuren al mòbil.

Aquí vaig tenir molts problemes amb els Futurs i asyncs de Flutter, de tal manera que de vegades no es pujaven bé a la base de dades. Finalment, vaig estar treballant amb l'Alex i ho vam poder solucionar.

I per acabar l'Sprint, l'Alex i jo vam estar preparant la presentació tècnica.

## Peilin Ni

Durant aquest últim Sprint, m'he dedicat a la continuació de les tasques assignades a l'Sprint anterior, que són principalment la US de multi idiomes i l'algorisme de rutes. Aquesta darrera tasca només es podia començar després d'obtenir les dades demanades a l'API de HappyLungs de forma correcta.

Primer em vaig dedicar a traduir els nous camps que s'han introduït per aquesta última fase. Els companys, majoritàriament de la capa de presentació, m'han indicat els textos a traduir per tal d'identificar-los de forma més ràpida. Aquesta tasca la va acabar de testejar l'Álvaro, ja que vaig haver de començar el càlcul de rutes ECO.

Un cop HappyLungs va poder oferir el servei segons la nostra especificació, vaig iniciar ràpidament la tasca de desenvolupar l'algorisme del càlcul de rutes, pel fet que era l'última setmana abans de la *Demo*. Donat que al Sprint anterior em vaig dedicar a una tasca similar per al càlcul de rutes amb càrrega, aquest no em va suposar cap problema a l'hora de desenvolupar-lo. Per últim, juntament amb l'Alexandru, vam estar testejant-les per poder finalitzar la tasca i completar el NFR de càlcul ràpid de rutes.

## Víctor Asenjo Carvajal

En aquest darrer Sprint he seguit amb la dinàmica dels dos darrers blocs de desenvolupament. A la capa de presentació he acabat de treballar amb les eines que oferim a l'usuari final per l'apartat de Ràpid aprenentatge: tutorial la primera vegada que entra a l'app, ajudes per les diverses pantalles, tags als botons per saber què fan. En resum tot el necessari per poder emprar l'aplicació de manera intuïtiva i saber-la

fer funcionar ràpidament. També hem seguit treballant en la facilitat d'ús buscada als [requisits no funcionals 11a](#).

D'altra banda, m'he encarregat de fer el display de tot el que es refereix a notificacions i com l'usuari pot interactuar dins l'aplicació en el procés de gestió d'aquestes. Aquesta part l'he desenvolupat conjuntament amb el Ferran De La Varga que era l'encarregat de la part de domini d'aquesta història d'usuari.

## Xavier Coll Ribas

En aquest darrer sprint he seguit fent gairebé el mateix que els dos últims. M'he centrat en la base de dades i en la API. He hagut d'afegir diverses taules a la BBDD per poder completar els nostres objectius i dur a terme les històries d'usuari. Sobretot, he hagut de treballar amb les tasques relacionades amb les notificacions (que s'ha treballat juntament amb en Víctor i en Ferran) i amb els trofeus (treballat amb l'Alvaro). Per cada cosa mencionada, he hagut de crear una entrada nova a la base de dades i fer les funcions necessàries per tal que els membres de domini i presentació poguessin treballar de manera adequada.

Un cop acabades les meves tasques, ha vingut una setmana crítica. He hagut de marxar a Alemanya degut a que tenia una competició acadèmica. Això ha fet que hagués de deixar llestes moltes coses abans que marxés i que no pogués fer la presentació. Durant el viatge, m'he dedicat a fer la documentació de la API interna que hem fet servir per tal de desenvolupar la nostra aplicació.

## 2. Requisits

### 2.1. “Not” List - Actualitzada

Només ha hagut un únic canvi: els gràfics de concurrència també es mostraran tant en els punt de càrrega com en les estacions Bicing, punt que inicialment vam considerar no necessària per al segon cas.

SÍ ÉS	NO ÉS
Tenir perfils d'usuari	Xarxa social que permet sistema de followers
Enregistrar els vehicles elèctrics que l'usuari dona d'alta	Registrar vehicles no elèctrics per part de l'usuari
Introduir les característiques del cotxe o moto elèctric d'un usuari	No té xat de cap mena
Geolocalització	Verificació del perfil d'usuari mitjançant dades privades
Generar rutes en el mapa pels usuaris entre la localització d'origen de l'usuari i el destí desitjat	Valorar perfils
Generar rutes en el mapa pels usuaris amb punts de recàrrega entre la localització d'origen de l'usuari i el destí desitjat	No efectua transaccions econòmiques entre l'usuari i punts de càrrega
Generar rutes ecològiques en el mapa pels usuaris amb punts de recàrrega entre la localització d'origen de l'usuari i el destí desitjat	Valorar punts del mapa
Comparativa entre el consum del cotxe o moto de l'usuari i un model semblant que utilitzi combustibles fòssils	No es permeten reserves de bicicletes a les estacions Bicing
Sistema de gamificació que dona premis als usuaris segons $n$ emissions estalviades, quilòmetres recorreguts, etc.	
Visualització de punts de recàrrega ocupats en el moment	
Mostrar els tipus d'endoll per carregar el vehicle en el punt de càrrega seleccionat	

Calcular el temps de recàrrega del cotxe, arribat ja al punt de càrrega indicat	
Fitxa d'informació de cada punt de càrrega: potència, endoll, preu, etc.	
Fitxa d'informació dels punts d'estació Bicing	
Mostrar en el mapa els punts amb accés a bicicletes Bicing	
Oferta de diferents tipus de rutes com l'estàndard, amb càrrega, l'ecològica i l'ecològica amb càrrega.	
Compartir ubicació una vegada l'usuari arriba a un punt de càrrega mitjançant un enllaç	
Notificacions i avisos d'un punt de càrrega als usuaris que tinguin marcats aquell punt	
Mostra informació concreta d'establiments o llocs concrets	
Mostrar gràfics de concurrència de les estacions de càrrega al mapa	
POT SER	
Instal·lació de punts de recàrrega propis pels usuaris de la nostra aplicació	
Opció de reserva de punts de recàrrega limitat per una certa categoria d'usuaris.	
No mostra gràfics de concurrència a les estacions bicing	
Ampliació d'una nova categoria d'usuaris Premium per tenir funcionalitats extres a l'aplicació	
Aplicació col·laborativa on els usuaris poden afegir nous punts de càrrega i les seves característiques no existents en l'aplicació	
Recomanador de rutes genèric per diferents tipus de transport sostenible: bus, bicicletes, patinets elèctrics, etc	
Reportar estacions de càrrega del mapa: anomalies de funcionament	
Mostrar 3 estacions més properes a la ubicació de l'usuari en el cas de quedar-se sense connexió	
Filtrar les estacions de recàrrega per tipus d'empresa	



## 2.2. Product Backlog

A la taula següent hi trobem les històries d'usuari d'Electrike.

Tal com indica, la columna “Estat” indica l'estat actual de la US: en el cas que ja estigui finalitzada, es marca l'Sprint en el que s'ha tancat; en el cas d'haver-se començat, però, no acabat, el trobem *In progress*; per últim, si la US encara no s'ha obert per aquest Sprint, el marquem com a N/A.

[Nota:

- Gran part de les US assignades a l'Sprint 1 han estat finalitzades. Aquelles que s'han començat són, en gran part, per altres Sprints.
- Les caselles marcades en **verd** són US que hem afegit; **groc**, les que hem modificat el nom.]

Èpics	Històries d'usuari	Estat
Gestió Usuari	Login	Sprint 1
	Log out	Sprint 1
	Sign up	Sprint 1
	Eliminar perfil	Sprint 1
	Imatge perfil	Sprint 1
	Afegir punt a preferits	Sprint 2
	Eliminar punt de preferits	Sprint 1
	Activar Notificacions	Sprint 3
	Desactivar Notificacions	Sprint 3
	Multi idiomes	Sprint 2
	Compartir ubicació	Sprint 3

Gestió Vehicles	Afegir un vehicle	Sprint 2
	Eliminar un vehicle	Sprint 2
	Modificar dades vehicle	Sprint 2
	Introduir bateria inicial	Sprint 2
	Escollir vehicle de viatge	Sprint 2
Gamificació	Consultar trofeu	Sprint 3
	Guanyar trofeu	Sprint 3
Mapes i rutes	Ruta més ràpida	Sprint 2
	Ruta amb menys contaminació	Sprint 3
	Ruta amb punts de recàrrega	Sprint 2
	Bicing propers	Sprint 2
	Recerca de direccions	Sprint 2
	Vista alternada	Sprint 1
	Consultar Estació de Bicing	Sprint 1
	Consultar punt de càrrega	Sprint 1
	Gràfiques d'ocupació dels punts	Sprint 2
	Visualitzar punts favorits	Sprint 2
	Visualitzar només punts favorits en el mapa	Sprint 2

Requisits no funcionals	Multiplataforma	Sprint 1
	Mostrar ràpidament les rutes	Sprint 3
	Ràpid aprenentatge	Sprint 3
	Aplicació fluida	Sprint 3
	Facilitat d'ús	Sprint 3
	Navegació per l'aplicació	Sprint 1

## 2.3. Requisits no funcionals - Plantilla Volere

### 10a. Requisits d'Aparença

- El producte ha de ser atractiu per a tota mena de públic que tingui l'edat per poder conduir un vehicle.
- El producte ha de complir amb uns estàndards decidits pels participants del grup, per tal que l'estètica sigui simple i minimalista alhora que uniforme a tot el sistema.

#### Criteri de satisfacció

Una mostra representativa del públic major de divuit anys amb un vehicle elèctric, sense incentius ni entrenament o formació prèvia, ha de començar a utilitzar les funcionalitats principals del producte en 5 minuts després de la seva primera trobada amb aquest.

L'equip de desenvolupament ha de certificar que compleix els estàndards i patrons estètics acordats per tal que es vegi com a una marca unificada.

### 11a. Requisits de Facilitat d'Utilització

- El producte ajudarà al fet que l'usuari no cometi errors a l'introduir dades.
- El producte serà fàcil d'emprar per a qualsevol adult.
- El producte farà que els usuaris vulguin utilitzar-lo.
- El producte ha de ser emprat sense entrenament i amb un coneixement mínim sobre el vehicle elèctric que fa servir l'usuari.

#### Criteri de satisfacció

El 75% del panel de prova d'adults podrà navegar pel mapa, visualitzar i afegir a favorits punts de càrrega i Bicing, afegir un vehicle i cercar una ruta amb èxit després d'un dia fent servir l'aplicació.

El 60% d'usuaris farà ús regularment de l'aplicació després d'un mes d'ús i familiarització amb les seves funcionalitats.

## 11b. Requisits de Personalització i Internacionalització

- L'usuari podrà canviar el llenguatge de l'aplicació en qualsevol de les opcions que oferim.
- Convencions decimals i sistema internacional d'unitats.

### Criteris de satisfacció

L'usuari podrà canviar entre els 3 idiomes disponibles en tot moment anant a l'opció, aquest són el castellà, l'anglès i el català. L'idioma seleccionat serà guardat per la propera sessió.

Les unitats de mesura seran marcades a cada camp.

## 11e. Requisits d'Accessibilitat

- El producte pot ser utilitzat per daltònics
- El producte serà utilitzable per persones amb sordesa.

### Criteris de satisfacció

Les persones daltòniques no tenen cap impediment per utilitzar les funcionalitats principals de l'aplicació, ja que cap informació rellevant es mostra amb només diferenciació de colors.

Les persones amb sordesa no tenen cap dificultat al fer servir l'aplicació, perquè cap informació es notifica per sorolls.

## 12a. Requisits de Velocitat i Latència

- L'aplicació ha d'iniciar-se en menys de 7 segons
- El producte ha d'oferir una ruta en menys de 3 segons
- L'usuari ha de poder canviar de vista i aquesta s'ha de carregar en menys de 2 segons, en el cas del mapa en menys de 5 segons.
- La base de dades s'actualitzarà cada 5 minuts les dades que no són estàtiques.

### Criteris de satisfacció

Els criteris anteriors es consideraran satisfets quan la connexió a internet de l'usuari és òptima i el dispositiu suporta amb normalitat la capacitat de l'aplicació.

### 12c. Requisits de Precisió o Exactitud

- Totes les quantitats s'arrodoneixen a 2 decimals.
- Les quantitats de temps s'expressen en minuts, hores o dies.

#### Criteris de satisfacció

Les mesures que no siguin de temps han de complir el primer criteri mencionat, ja que la resta de decimals no són importants per l'usuari.

Les unitats de temps l'actitud en segons no és necessària, perquè és massa exacta en una aplicació per crear rutes i aquest segons poden variar bastant per la velocitat de l'usuari, inclòs els minuts poden.

### 12d. Requisits de Confiabilitat i Disponibilitat

- El producte estarà disponible 24 hores al dia, 365 dies per any.
- El producte treballa el 97 % del temps sense fallades.

#### Criteris de satisfacció

El producte ha d'estar disponible i en funcionament el temps mencionat i no ha de fallar tret que la connexió de l'usuari no sigui bona.

### 12e. Requisits de Fortalesa o Tolerància a Fallades

- El producte ha de guardar l'última ruta consultada per l'usuari si es queda sense connexió
- El producte mostrarà les dades de l'usuari, els seus vehicles i marcarà els punts de càrrega i Bicing si l'usuari ha consultat aquestes dades mentre està enregistrat.

#### Criteris de satisfacció

El producte ha de poder mostrar la informació mencionada anteriorment sense connexió una vegada s'ha iniciat un comte i s'ha consultat aquesta informació mínim un cop quan aquest tenia connexió a internet.

### 13b. Requisits d'Interfície amb Sistemes Adjacents

- L'aplicació es podrà utilitzar en els navegadors webs més populars (Chrome, Edge...)

#### Criteris de satisfacció

L'aplicació ha de poder ser compilada i executada per a web. A més de ser testejada per l'equip i que les funcionalitats principals funcionin en els navegadors més populars.

### 14c. Requisits d'Adaptabilitat

- Android Jelly Bean, v16, 4.1.x o posterior, i iOS 8 o més nous.
- Dispositius iOS (iPhone 4S o posterior) i dispositius Android ARM.
- Window, MAC i Linux

#### Criteris de satisfacció

L'aplicació ha de poder ser compilada i executada per a web, Android i IOS a través de l'opció que dona el *framework* Flutter. A més de poder descarregar-la i utilitzar-la correctament indistintament del tipus de dispositiu.

### 15a. Requisits d'Accés

- L'usuari haurà d'iniciar sessió per tal de disposar de totes les funcionalitats importants del producte.

#### Criteris de satisfacció

Els usuaris enregistrats disposaran de les funcionalitats al complet del producte, en canvi, un usuari que no ha iniciat sessió només podrà visualitzar el mapa, els punts de càrrega i Bicing.

### 15c. Requisits de Privacitat

- El producte requerirà el consentiment per part de l'usuari per usar la seva ubicació.
- Es notificarà a l'usuari per tal que sàpiga quines dades seran emmagatzemades en la base de dades
- Es notificarà a l'usuari per tal que sigui conscient de quines dades seran utilitzades d'altres aplicacions.

#### *Criteris de satisfacció*

Els usuaris hauran d'acceptar aquest requisit per fer servir per complet l'aplicació i ha de ser conscient de l'ús que li donem i el que comporta.

Mantenim encriptades aquestes dades dintre de la base de dades del producte, a les quals només podrà accedir ell mateix i l'equip de desenvolupament, per tal de mantenir la privacitat de l'usuari.



## 2.4. Gestió Taiga

La informació relacionada al progrés de les tasques d'aquest Sprint i les US es poden consultar al nostre taulell de [Taiga](#).

Dins de cada història d'usuari, hi ha el conjunt de tasques necessàries per tal de poder-la completar. Aquestes tasques estan ben indicades, és a dir, han seguit el procés de “Not done” → “In progress” → “Test” → “Closed”. A més, està apuntat el temps previst per cada una i el temps real que hem tardat, que prèviament havíem apuntat en el PRT, per tal de fer un millor seguiment.

A més, a petició de Silverio, hem definit també els respectius “Criteris d'acceptació” de cada US d'aquest Sprint. Cal remarcar que aquests criteris poden anar actualitzant-se en futurs Sprints en el cas de que la US no s'hagi finalitzat. També pot ser que la US encara no assoleix tots aquests criteris, pel mateix motiu de que encara no està tancada.

Un punt a ressaltar és la gestió de les NFRs: no tots aquests requisits han passat a ser US, n'hi ha que formen part d'altres històries d'usuari i, per tant, es distingeixen com a tasques.

Aquelles que han passat a ser US directament i aquesta entrega són les següents:

- Requisits d'Aparença i facilitat d'utilització → Navegació aplicació: les funcionalitats de l'aplicació es troben fàcilment al navegar per aquesta. Les opcions ofertades es distingeixen molt bé en els apartats del menú lateral.
- Requisits d'Adaptabilitat → Multiplataforma: aplicació desenvolupada tant per web com aplicació mòbil.

Altres que han estat assignades dins d'US:

- Requisits d'accessibilitat → Consultar punt de càrrega: els punts de càrrega mostren la informació amb icones diferenciadores que ajuden a l'enteniment dels estats d'aquests punts, és a dir, no es distingeixen únicament per color.
- Requisits d'accés → *log in* i *sign up*: aquestes funcionalitats es duen a terme amb els respectius comptes de Google de l'usuari.

## 2.5. Funcionalitats transversals

Aspecte transversal	Descripció	Estat
Geolocalització	L'aplicació consisteix en un mapa on es mostra diferents marcadors i la localització de l'usuari, a més dels marcadors, també proporciona a l'usuari diferents camins per arribar a localitzacions (rutes)	Acabat
Xarxes socials	L'usuari es dona d'alta al sistema mitjançant les seves credencials de Google. La seva sessió dins l'aplicació es manté oberta.	Acabat
	Pot compartir una ubicació generant una URL.	Acabat
Gamificació	L'aplicació assigna premis als usuaris segons els requisits predefinitos dins del sistema.	Acabat
Notificacions	L'usuari rep notificacions d'aquells punts que ell ha activat. Aquests avisos informen de l'ocupació dels punts.	Acabat
Calendari	L'usuari pot activar notificacions personalitzades, defineix un dia i una hora. Aquesta segueix l'horari definit al calendari estàndard de la zona.	Acabat
Multi idioma	Proporciona a l'usuari diferents idiomes per als textos de l'aplicació, en concret: català, castellà i anglès	Acabat
Multiplataforma	L'aplicació és compatible amb Android i web.	Acabat
Gràfics d'ocupació	Es podran veure gràfiques de les ocupacions dels punts Bicing i dels carregadors de Barcelona.	Acabat

## - Geolocalització

Aquest aspecte s'ha desenvolupat amb la API de Google Maps, el qual ens ofereix tota la informació que hem necessitat fins ara.

La nostra integració del concepte transversal ha estat la implementació d'un mapa com a finestra principal, en la que localitzem a l'usuari, diferents punts de càrrega i Bicing, que aconseguim de la base de dades i de les APIs obertes. També hem afegit rutes per trobar el mètode més ràpid per arribar a un punt, a més de la ruta alternativa en el cas que el vehicle elèctric necessiti un punt de càrrega intermig, i la ruta ecològica que marca el camí amb menys contaminació. Per obtenir les localitzacions al mapa de manera senzilla, hem implementat una barra de cerca de direccions, que serveix per afegir una ruta al mapa.

## - Xarxes socials

Per poder guardar la informació relacionada a un usuari dins del nostre sistema, hem fet servir la llibreria i l'API de Google Sign In, i hem utilitzat la web de Google console, on s'ha configurat aquesta gestió d'usuaris. Inicialment, tenim una pàgina per fer el login, però ara ha canviat a un apartat del menú, necessari per utilitzar diferents funcionalitats ja esmentades. Aquest login demana a l'usuari el correu electrònic de Google i fins ara només podem registrar-nos amb els nostres comptes, i amb els dels professors que hem afegit.

Ara a l'últim Sprint, hem afegit que l'usuari pugui compartir la seva localització, o qualsevol altre, mitjançant un URL que es genera quan ell decideix fer ús d'aquesta funcionalitat.

## - Gamificació

A Electrike hem dissenyat un sistema de gamificació amb trofeus per tal de premiar als usuaris per la utilització de la nostra aplicació. El conjunt de trofeus actualment existent està desenvolupat a partir de codi, és a dir, els requisits per aconseguir-los estan programats a mà i, per tant, hi ha poca varietat. Els requisits, com es pot arribar a intuir, estan relacionades amb l'ús de l'aplicació i, en conseqüència, poden comptabilitzar característiques del perfil d'usuari com els km totals recorreguts, la quantitat de Co2 estalviada, etc.

En un futur, en cas de publicar l'aplicació i aconseguir un volum de descàrregues i usuaris considerables, seria recomanable que aquest sistema es gestioni mitjançant un admin-web. D'aquesta forma la creació de nous premis seria més fàcil i es podria crear una gran varietat d'aquests.

#### - **Notificacions**

L'usuari pot rebre notificacions sobre un punt de càrrega de la seva llista de favorits quan així ho desitgi. Les notificacions actualment només onden la informació del punt de quan l'usuari s'ha loggejat, ja que per falta de temps i desconeixement de com fer que aquesta informació s'actualitzés, no hem pogut fer que la informació del punt de càrrega sigui la del mateix instant en el qual l'usuari rep la notificació. Per poder desenvolupar aquesta funcionalitat hem utilitzat dues llibreries de Flutter que és *local\_notifications* i *provider*, que es guarda les notificacions i s'encarrega d'enviar-les un cop arriba el temps de fer-ho.

#### - **Calendari**

Aquest aspecte està directament relacionat a l'anterior, és a dir, un usuari pot definir una notificació donat un dia concret o un dia de la setmana, una hora, etc. Aquestes característiques segueixen el fus horari de Catalunya, que és el territori sobre on està desenvolupat l'aplicació.

#### - **Multi idioma**

Per poder oferir aquest aspecte, nosaltres hem considerat les 3 llengües més parlades dins del territori en què està pensada l'aplicació: català, castellà i anglès. Fent ús d'un plugin que ens ofereix el mateix entorn, nosaltres només hem hagut de fer les traduccions dels camps que canviarien d'idioma i fer crida de la funció pertinent dins del codi de la UI per carregar el text que pertoca.

#### - **Multiplataforma**

Gràcies al framework amb el que treballem, la implementació de la pàgina web va enllaçada al desenvolupament de l'aplicació mòbil. D'aquesta forma, no necessitem implementar un nou codi per fer el desplegament en web. Al compilar el codi que tenim actualment, podem escollir l'opció de desplegar-lo en format web, cosa que mostrara la versió desenvolupada, fins al moment.

### **- Gràfics d'ocupació**

Donat un punt de càrrega de la nostra base de dades, l'usuari té l'opció d'obtenir la concurrència d'aquests en format de gràfic lineal. Aquesta informació pot decidir obtenir-la tan visitant la informació detallada d'un punt de càrrega com configurant una notificació per avisar-li de l'ocupació del punt.

Per desenvolupar aquesta funcionalitat, han hagut de cooperar les 3 capes de l'arquitectura per tal d'obtenir les dades necessàries per dibuixar el gràfic que es mostra a l'usuari.

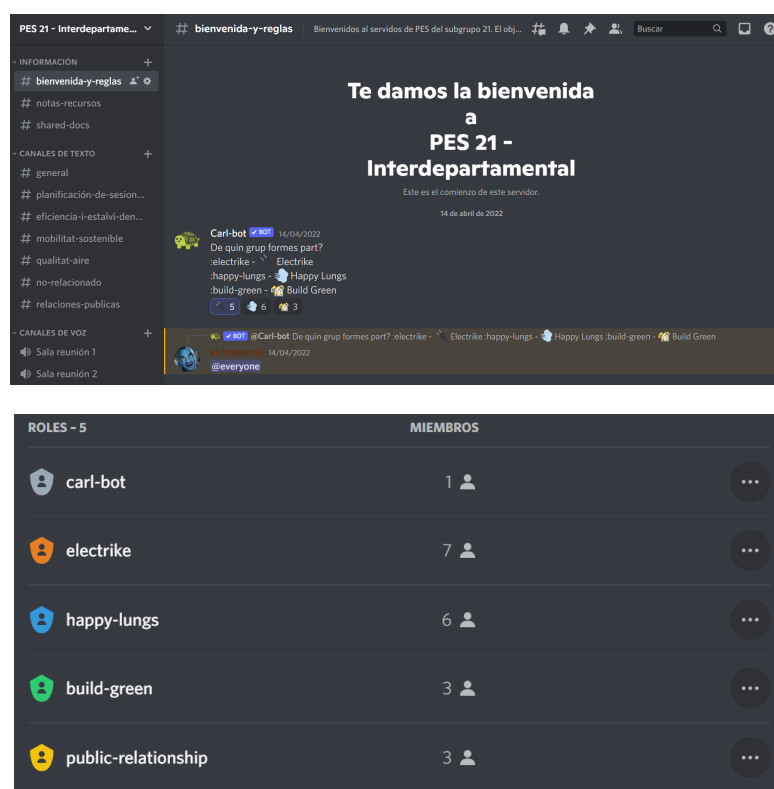
## 2.6. Serveis de tercers

### Comunicació entre equips

Tenim 1 via principal de comunicació i una en reserva per si falla la primera.

La principal és Discord per a reunions, consultes i discussions que engloben tot tipus de detall. S'ha eliminat el grup de WhatsApp perquè no s'utilitzava i s'ha incorporat a Discord un nou canal de comunicació de text que substitueix el de WhatsApp. A Discord estem tots els integrants del grup 21 de PES 2021-22 Primavera. S'han assignat rols a cada integrant del grup de Discord per diferenciar els equips de desenvolupament. Per dur la gestió d'assignació de rols s'ha emprat el bot Carl-bot: amb aquest bot es crea un missatge al canal que es vulgui ("Benvinguda i regles" en el nostre cas) i els usuaris de la sala poden interaccionar amb les reaccions; les quals tenen un rol vinculat que assigna automàticament. D'aquesta manera també facilitem la menció als diferents grups per comunicats entre equips.

Com a mètode de comunicació en cas de fallada de serveis de Discord és Gmail: el correu oficial de la Universitat.

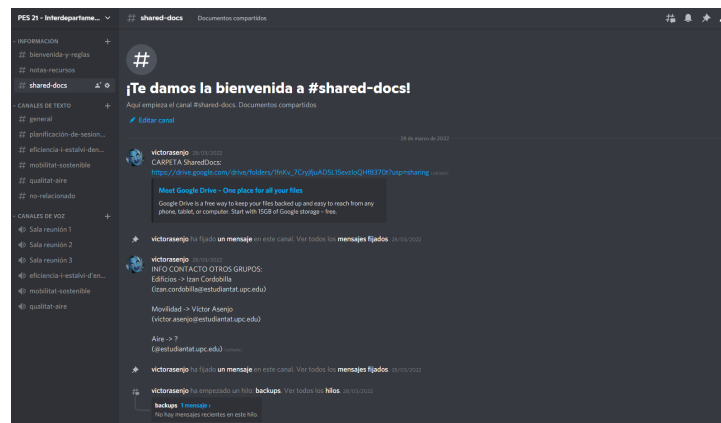


## Funcionalitats pactades

La repartició de serveis acordada seria la següent:

<b>Grup</b>	<b>Oferim a</b>	<b>Rebem de</b>
Qualitat de l'Aire	<i>Electrike</i>	<i>BuildGreen</i>
<i>Happy Lungs</i>	els punts de contaminació i els seus nivells segons una determinada coordenada i un radi per escollir la zona respecte a la coordenada	geolocalització d'aquells edificis que pertanyen a un determinat rang d'eficiència
Cases Sostenibles	<i>Happy Lungs</i>	<i>Electrike</i>
<i>BuildGreen</i>	geolocalització d'aquells edificis que pertanyen a un determinat rang d'eficiència	els punts de càrrega (i si volem, parades de bici) més propers donada una ubicació concreta (coordenades).
Mobilitat sostenible	<i>BuildGreen</i>	<i>Happy Lungs</i>
<i>Electrike</i>	els punts de càrrega (i si volen, parades de bici) més propers donada una ubicació concreta (coordenades).	els punts de contaminació i els seus nivells segons una determinada coordenada i un radi per escollir la zona respecte a la coordenada

Per treballar conjuntament sobre un mateix fitxer fem servir [Drive](#). El link a la carpeta d'aquest es troba com a missatge fixat al xat "Shared-Docs" de Discord i hi ha un fil, en aquest mateix canal, anomenat "backups" on es van pujant còpies de seguretat d'aquests documents periòdicament.





### 3. Cerimònia Agile

#### 3.1. Report on the sprint planning, review and retrospective meetings

##### Sprint Planning

Aquest Sprint el vam pensar per tenir la menor càrrega de treball possible així que la majoria d'històries d'usuari són per tal de millorar l'aplicació, encara així hi ha diverses funcionalitats que hem desenvolupat en aquests Sprint. Per tant, aquestes històries són millores i coses que queden per fer dels anteriors Sprint. Els costos de les històries d'usuaris definits anteriorment i de les actualitzacions hem posat costos baixos, ja que no tenen massa complexitat. Així que com a resultat d'aquest Sprint volem tenir al 100% l'aplicació acabada i amb millores en algunes tasques dels anteriors Sprints.

Història d'Usuari	Puntuació
Rutes	7
Ruta amb menys contaminació	21
Guanyar trofeu	8
Consultar trofeu	1
Compartir ubicació	5
Notificacions programades	13
Desactivar notificacions programades	3
Càlcul ràpid de routes	8
Facilitat d'ús	8
Aplicació fluida	8

Ràpid aprenentatge	8
Actualitzacions	N/E

Primer hem canviat de scrum master com és normal a l'inici d'aquest sprint en el projecte, sent l'Eloi el nou scrum master.

A continuació vam pensar que hauríem de millorar de l'aplicació a més de les taques que queden per fer, ja que coneixem bugs o coses que es poden fer d'una altra manera i perquè funcionés tot millor, perquè després de dos mesos familiaritzant-nos amb dart i flutter tenim més recursos per millorar treball que s'havia fet a l'inici.

Hem continuat amb la mateixa estructura d'equips que a l'anterior, l'Alexandru Dumitru com a full-stack pel seu domini en els diversos camps, el Xavi encarregat de la base de dades i de l'api que consumeixen l'altre equip, el Ferran, la Peilin i l'Álvaro dedicats a la capa de domini, i el Víctor i l'Eloi dedicats a la interfície, amb ajuda de l'Álvaro, ja que com a tingut bastant contacte amb presentació durant els anteriors sprints, en aquest s'ha dedicat a fer algunes tasques sense complexitat per tal que la resta de l'equip de presentació pugui dedicar-se a tasques complexes sense perdre temps.

A més, l'Alexandru ha migrat la base de dades a AWS pel fet que abans ho teníem al servidor propi i, com que no necessitàvem desenvolupar molt més codi pel back-end, vam veure oportú canviar-lo al final d'aquest Sprint.

## Sprint Review

### Sprint 1 review

Sprint review feta amb Maria José a la sessió de classe següent a la de l'entrega.

La gran majoria de US s'ha completat de forma satisfactòria, a excepció de la “Visualització de Mapa”.

En aquesta, la tasca de Geolocalitzar l'usuari a través del dispositiu amb el qual està consultant l'aplicació per prendre la coordenada d'origen, ha estat completada amb èxit, al contrari del que creiem. La consideràvem més complicada que introduir manualment la direcció d'origen. Per aquest motiu, s'ha deixat pel Sprint 2 la tasca de poder introduir un punt d'origen manualment, a més de prendre per defecte la localització de l'usuari.

Una altra US que no s'ha tancat ha sigut la de “Rutes ràpides” degut a que l'equip ha considerat més òptim deixar la tasca que falta, que és mostrar gràficament la distància i duració del recorregut, per quan tinguem els tres tipus de rutes que acabarem oferint. A part d'aquestes US, no hi ha hagut cap més inconvenient relacionat al progrés de les funcionalitats.

Com a punts a millorar: la Product Owner ha considerat convenient afegir a l'inici de l'aplicació una petita llegenda per facilitar la interpretació de les icones que fem servir per indicar els estats i compatibilitats dels endolls de les estacions de càrrega.

També recomana mostrar les localitzacions, tals com ciutats, províncies, barris, etc, en el mapa per facilitar i situar l'usuari al mapa. A més, coincideix amb l'equip amb la idea de clusteritzar els punts de càrrega i Bicing per tenir un mapa més “net” i millorar la UX.

Els punts forts han estat que un gran percentatge de les US assignades com a objectiu en aquesta entrega s'han completat amb satisfacció i bona qualitat de resultats. És més, s'han obert altres US que no estaven previstes per aquest Sprint, cosa que ens fa pensar que podem ampliar l'abast de cara als pròxims lliuraments. A la M<sup>a</sup> José li ha agradat la idea que gravem vídeos amb petites demostracions entre nosaltres per tal de tenir un back-up en el cas que falli el servidor, o qualsevol altre element, al dia de

la demo oficial.

## Sprint 2 review

Sprint Review de la release 2 feta amb Silverio a classe.

La gran majoria de US s'ha completat satisfactòriament a excepció de “Notificacions programables”, una de les US importants de l'aplicació.

Aquesta només s'ha pogut desenvolupar fins al punt que l'usuari pot rebre una notificació a l'instant immediat de pulsar un botó. Originalment, s'esperava que per aquest Sprint l'usuari pogués rebre un missatge de l'aplicació a un dia i hores concretes que l'usuari hagi decidit programar l'alarma. Per motius d'organització de l'equip i la falta de coneixement de la tecnologia a aplicar per aquesta funcionalitat, no s'ha arribat a completar a temps, per tant, aquesta passa com a tasca pendent per a la següent release. Segons el Product Owner, per tal de considerar aquesta US com a transversal satisfactòria, hem de contemplar la idea de programar més d'una notificació per punt de càrrega.

De les recomanacions fetes al sprint review 2 per la M<sup>a</sup> José, no s'ha pogut començar encara el desenvolupament de les llegendes per la interpretació de gràfics i icones de l'aplicació per la facilitat d'ús. Això és degut al fet que l'equip ha decidit fer tota la NFR de facilitat d'ús per al següent Sprint, on tindrem la versió definitiva de l'aplicació i, per tant, la UI final.

Com a punts a millorar: el Silverio ha recomanat millorar les gràfiques d'ocupació. Seria convenient que l'usuari pogués interaccionar amb les gràfiques d'alguna forma, sigui seleccionant o filtrant els dies i hores a la setmana en les quals vol consultar aquests gràfics. També ha suggerit que l'usuari pogués seleccionar, mitjançant un calendari, un dia en concret per fer la consulta de la gràfica del punt de càrrega. En resum, queda com a decisió de l'equip quin mètode afegir a les gràfiques d'ocupació per fer-les més dinàmiques i personalitzables per l'usuari.

Per acabar, com a punts forts d'aquest Sprint, l'equip s'ha organitzat millor i, per tant, hem pogut avançar força en el product backlog. El Product Owner considera que portem un bon percentatge i ritme de treball per ser l' Sprint 2, el qual recomana mantenir per acabar bé l'aplicació.

## Sprint Retrospective

### Sprint 1 Retrospective

Tenint en compte que aquest és el primer Sprint de desenvolupament dins d'un entorn completament nou per a tots els membres de l'equip, el caos que ens hem trobat a l'inici és bastant comprensible.

Primer de tot, les eines de gestió de projecte no s'han utilitzat de la forma més eficient: el Taiga no s'ha anat actualitzant com cal i les tasques no s'han anat assignant a l'inici, que és quan s'ha de fer per mantenir una càrrega de treball compensada entre tots els membres; quant a PRT, no hem indicat clarament sobre quina tasca i US anava treballant cadascú i, per tant, a l'hora de calcular les hores totals reals dedicades a aquelles tasques, només hem pogut obtenir valors estimats. De cara al pròxim Sprint, es planteja tenir una bona definició de tasques tant al PRT com al Taiga, d'aquesta forma tothom tindrà el seu propi objectiu al Sprint.

Seguidament, relacionat al punt anterior, podem observar com en el PRT hi han membres que doblen les hores dedicades a uns altres. Per tal de disminuir aquesta desigualtat, a més de millorar les eines de gestió, hem vist convenient disminuir la quantitat de tasques d'aquests membres per la següent iteració i, augmentar les tasques en aquells que tenen menys hores dedicades. D'aquesta forma, de cara a l'últim Sprint, tothom tindrà unes hores similars per repartir les tasques equitativament.

Com a punts positius, hem de mencionar que l'equip té una molt bona comunicació entre sí i hi ha un molt bon ambient de treball. Això ajuda a l'hora de demanar canvis o informació entre les diferents capes de l'arquitectura.

Com a últim, la qualitat del codi és força bona, ja que tothom segueix un estàndard definit en el nostre entorn de Github i, per tal de poder fer commit d'una tasca, aquesta ha de superar els testos del CD.

## Sprint 2 Retrospective

Per començar, analitzarem els punts febles per poder millorar de cara el pròxim Sprint.

Tot i haver fet una segmentació i assignació de tasques més detallada que al Sprint 1, s'ha notat que moltes d'aquestes generen dependències amb altres assignades a diferents membres.

És a dir, algunes tasques depenien d'altres, ja que es necessitava informació d'una altra capa de l'arquitectura i, per tant, no es podia avançar fins que l'anterior no finalitzi. Per aquest problema, l'equip veu convenient dues solucions: assignem les tasques de forma més agrupada de manera que una persona s'encarregui de les tasques que formen una mateixa US o, establir deadlines a la persona encarregada de la tasca de la qual es depèn, de forma que evitem que es deixin tasques vitals per a l'última setmana del Sprint o s'assignin tasques noves a l'últim moment. Per aquest mateix motiu, hi ha hagut dues US que no s'han pogut tancar per aquesta entrega les quals són totes les de la família de "Notificacions", ja que no vam calcular molt bé el temps i no es va començar amb la suficient antelació.

A part d'això, no hi ha res més dolent a destacar, ja que, gràcies a la bona assignació i definició de tasques al principi d'aquest Sprint, ens hem evitat problemes que ens hem trobat en el Sprint 1 tals com: càrrega de treball molt descompensada on algunes persones doblen les hores de treball d'altres, "trepitjar-se" la feina entre membres per la mala assignació de tasques, etc. L'adient organització de l'entorn de treball com el Taiga i el PRT, ha millorat les dinàmiques de treball i, tanmateix, a compensar aquell desbalanç d'hores dedicades al Sprint anterior.

Per últim, en quant a qualitat de codi, al seguir el mateix estàndard establert a l'inici del procés de desenvolupament, no hi ha cap problema en quant a diferència de format del codi. Cal fer especial èmfasi que l'equip ha considerat suprimir la

necessitat dels tests unitaris, ja que no n'hem fet ús d'aquests ni en la creació de rutes amb càrrega, que és considerada com una de les tasques més complexes del sistema.

### Sprint 3 Retrospective

Iniciarem analitzant els punts forts. No hem canviat la metodologia d'assignació del treball i interacció respecte a l'Sprint anterior, pel fet que ja ho veiem eficient. Després d'haver compensat la càrrega de treball en el Sprint anterior, aquest cop hem assignat les tasques de forma més equitativa possible. També hem tingut en compte la continuïtat de les tasques assignades als membres en les entregues anteriors, intentant així que tothom sàpiga què està desenvolupant de cara a aquesta última entrega. Per exemple, si en Ferran s'encarregava de les notificacions, ha seguit amb tasques d'aquesta mateixa US.

Respecte les observacions de l'Sprint 2, hem intentat disminuir les dependències entre les tasques a l'hora d'assignar-les. Tot i això, hi ha tasques sobre les quals no s'ha pogut aplicar, tals com les traduccions, ja que tenia dependències de moltes US com gràfiques d'ocupació, notificacions, gamificació, etc. Però, en la gran majoria de casos sí que s'ha pogut aplicar.

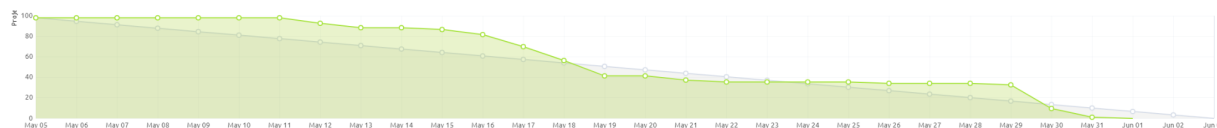
Per últim, hi ha un parell de punts febles a destacar.

En primer lloc, hi ha membres que han trobat a faltar als Daily Scrums, més específicament, els moments en els quals l'Scrum Master ordena que els membres deixessin de treballar i tothom posava en comú l'estat de les seves tasques i els problemes que es trobaven, ja que no sempre es deixaven de fer altres activitats mentre es feia el Daily Scrum.

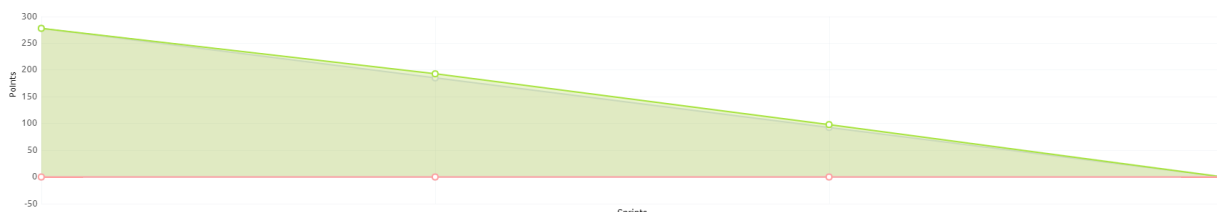
En segon lloc, hi ha hagut discussions entre els membres de l'equip, principalment de comunicació. També pot ser degut a l'alçada del curs actual i l'estrès acumulat en l'equip. Tot i això, ja s'ha parlat amb els membres aplicats.

Pel que fa a visió global, l'equip se sent molt satisfet amb el treball fet i el producte resultat obtingut pels mesos d'esforç.

### 3.2. Updated release and iteration burndown charts and velocity chart



Quant al Burndown chart podem observar com la majoria d'històries d'usuari s'han anat completant a la meitat del sprint, i s'han completat finalment totes les històries uns dies abans de l'entrega final per evitar fer crunch. D'altra banda, això va ser possible perquè vam assignar una data límit pròpia, concretament una setmana abans de la presentació final per evitar deixar tasques importants per l'últim moment.

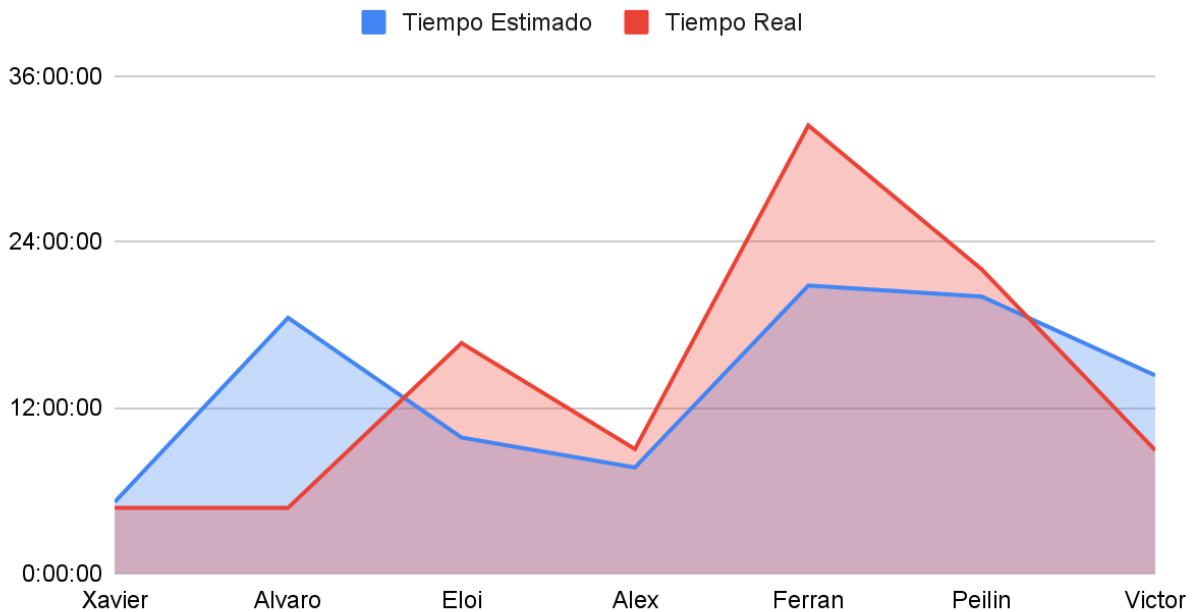


D'altra banda, amb el Burndown chart del backlog, no s'aprecia gaire, però podem veure com hi ha una petita desviació de puntuació, sobretot en el Sprint #1, que és conseqüència de traslladar certes tasques a altres Sprints per problemes de temps.

No obstant això, vam poder acabar totes les tasques definides per aquest projecte.

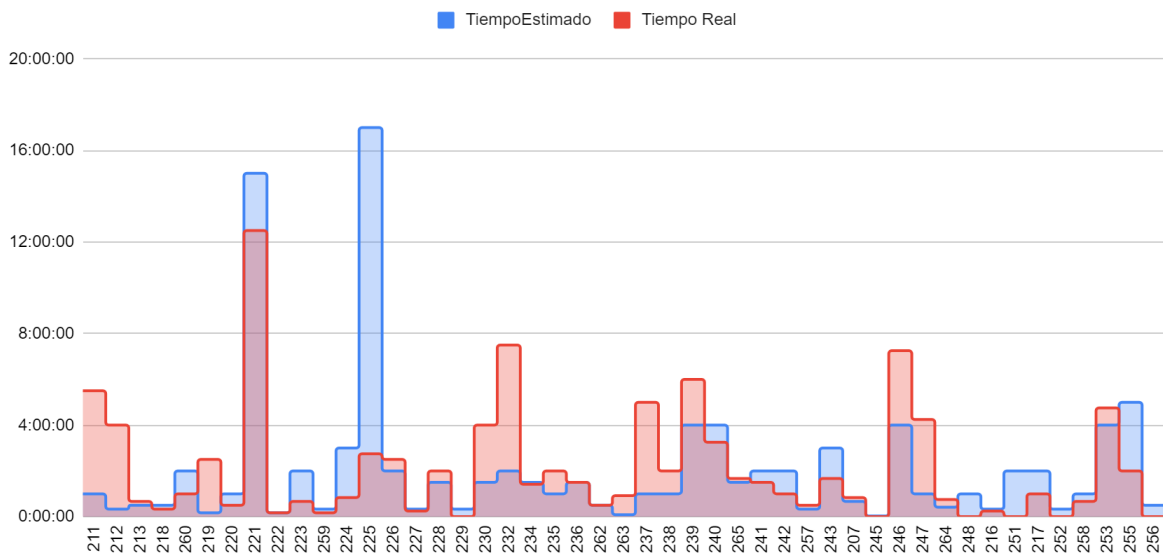


## Temps Estimat i Temps Real



En aquest altre gràfic podem veure, en el cas d'una tasca de l'Álvaro, la qual és del sistema de trofeus, esperàvem que la tasca portés molt més temps del que realment va portar. Per contra, en el cas del Ferran i l'Eloi, les seves tasques van portar més temps de l'esperat per certs bugs o errors que trobem a l'implementar-les. Per últim, les estimacions referents a les tasques del Xavi, de l'Alex, de la Peilin i del Victor van ser més adequades.

## Desglossament de tasques



En aquest últim gràfic podem veure el temps individual per cada tasca. Aquí es pot observar com la majoria de tasques les vam poder aproximar correctament a excepció d'unes certes tasques que van requerir més temps del que vam preveure, la majoria de notificacions i d'alguna altra que vam sobreestimar el temps com el sistema de trofeus.

Així i tot cal destacar que aquest tipus de tasques les vam anar fent esporàdicament entre més persones, per la qual cosa el temps real no arriba a ser de tan sols una persona.

## 4. Metodologia

### 4.1. Visió global

Nosaltres hem adoptat una metodologia de treball SCRUM. D'aquesta metodologia podem destacar 5 events: la del Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective i Refinement.

Pel que fa al Sprint Planning a les fases d'Inception: primer decidim el Scrum Master d'aquell sprint, mirem quines són les tasques a elaborar i ens distribuïm la feina entre els integrants.

En el Daily Scrum, cada integrant de l'equip, o els subgrups en què s'ha creat per fer les tasques, posen en comú el treball que ha avançat i si necessita ajuda o implicació d'algun altre integrant més. També es posa en comú informació interessant trobada de cara al procés de desenvolupament. A destacar, a cada daily scrum, miràvem com anava el desenvolupament de les històries d'usuari i posàvem al dia el Taiga. Llavors, de decidir en quines tasques ens posàvem a continuació

Per el primer Sprint, si podrem fer una sprint review juntament amb la M<sup>a</sup> José. Tot i no estar per aquesta mateixa entrega, la idea és fer una presentació, amb els stakeholders (els professors) dels resultats obtinguts durant aquest Sprint fent una presentació del producte actual. D'aquesta forma veurem si s'han completat els objectius definits en el Sprint Planning.

Per últim, la fase de refinament ens l'hem trobat quan algú de nosaltres descobria alguna informació o aportava quelcom interessant i la compartia amb el grup. Llavors, es feien les modificacions pertinents. Tot i seguir aquesta planificació, és molt habitual el contacte entre els membres de l'equip, ja que molts treballem de manera conjunta i necessitem avançar al mateix ritme.

## Gestió de l'equip

Per treballar en els Sprints hem segmentat l'equip en tres sub equips principals segons la capa en la qual s'havia de treballar: capa de presentació, capa de domini i capa de dades.

Segons les preferències de cada membre, s'han assignat 2 persones per capa de Presentació: Víctor Asenjo i Eloi Balaer; 3 persones per Domini: Álvaro Rodríguez, Peilin Ni i Ferran de la Varga; i 1 per la BD: Xavier Coll. L'Alexandru Dumitru, que abans formava part del grup de BD, ara segueix sent el github master com a l'anterior sprint i participa en tasques de tot el projecte (fullStack).

Un cop les tasques assignades a cada membre han anat finalitzant, els equips s'han anat redistribuint segons la demanda d'altres capes o tasques encara pendents. És a dir, si en Domini necessitaven més ajuda, el que havia acabat la seva feina en BD els ajudava.

## 4.2. Gestió projecte

La gestió del projecte la fem mitjançant diversos softwares.

Per mantenir en ordre les nostres tasques, fem servir el Trello, on apuntem què hem de fer, per a quan ho hem de fer i qui ho ha de fer. Tenim columnes per marcar aquelles tasques a fer, fetes, en procés o en revisió. També tenim definit les hores estimades per cada tasca i el temps real que hem tardat.

Al Taiga tenim les històries d'usuari, amb els seus propis Criteris d'acceptació i amb els seus respectius costos (calculats prèviament amb una sessió de planning poker que vam fer online). Allà, tenim el backbone, on ens organitzem per sprints. Dins de cada sprint, tenim diverses històries d'usuari dividides per tasques, on cada cop que treballem, ens assignem una i la intentem realitzar. En cas que l'acabem, la movem a la casella de "Ready for Test" i, un cop integrat, es classifica com Closed.

Per tenir constància del nostre temps de treball, tenim un Google SpreadSheet on anotem les hores que dediquem individualment per cada tasca relacionada amb el projecte. Per finalitzar, les tasques les repartim entre membres del grup, ja que hem vist que no és eficient treballar tots junts en una cosa concreta i, a més, ens és difícil coincidir a tots.

### 4.3. Gestió repositori

De cara a la gestió de repositoris fem servir principalment GitHub i tenim un encarregat, Alexandru Dumitru, que és el Github Master, qui fa les branches i els merges a petició dels membres del grup. Quan una persona creu que el que ha desenvolupat, té els estàndards establerts, no dóna error i passa els corresponents tests, aquest ho puja a la seva branch. Un cop allà, es demana a l'encarregat del Github que faci merge cap a master o cap a una altra branca que necessita el codi pertinent, per exemple, un company del mateix subgrup.

Respecte al workflow del mateix repositori, hem dividit aquest en dos: en Backend i Frontend, per tal de facilitar la seva implementació i tenir-lo diferenciat. Així doncs, el Backend es troba a un repositori i, d'altra banda, el Frontend. Pel que fa a l'estructura interna de cada repositori, seguim un model GitFlow on hi haurà una branca Màster on estarà el producte final i múltiples branques secundàries de desenvolupament per cada membre que estigui treballant al repositori. Podem veure tota aquesta estructura als següents enllaços: [Backend](#); [Frontend](#).

[Notes:

- *Com podem observar, actualment el nostre repositori ja té gràfics de workflow segons els commits que cada membre ha anat fent. Tot i així, aquesta informació no és precisa: hi ha membres que han treballat en un altre entorn pel desenvolupament de la BD, on ara tenim el repositori HADES, o en la dedicació de la descripció del document.]*

#### 4.4. Comunicació dins l'equip

Pel que fa a la comunicació entre membres del grup, tenim un grup de Whatsapp on decidim horaris de reunions i per on enviem petites demos de l'evolució de l'aplicació amb vídeos, imatges... i parlem sobre assumptes poc rellevants i volàtils.

D'altra banda, tenim un Discord on tenim un petit "achieve" de tecnologies o documents d'interès per al projecte i per a on ens comuniquem de manera més habitual i per on fem les reunions online.

A part de les reunions que es duen a terme durant les dues sessions presencials de classe a la setmana, també s'ha organitzat reunions online quasi diàries per acabar de polir el treball fet o per definir les tasques a fer.

Per a aquest Sprint, ens hem arribat a connectar diàriament per treballar en sub equips i, especialment, en la integració del projecte i resolucions de problemes, ja que la comunicació és més immediata.

## 4.5. Gestió de la qualitat

Després d'haver estat desenvolupant amb flutter hem pogut observar que implementa nativament un analitzador de codi estàtic, d'aquesta manera hem evitat utilitzar eines com SonarQube.

A més, aquesta anàlisi es realitza automàticament tant en les màquines individuals de cada integrant com per al CI del nostre repositori Github. Tenir-ho localment aporta gran avantatge sobre el desenvolupament, ja que no teníem la necessitat d'eines de tercers i sobretot ho fa en directe estalviant-nos temps. D'aquesta manera ens assegurem de pujar un codi net.

Tenir aquest sistema ens ha ajudat a tenir un codi més net i organitzat perquè, sense dependre de qui estigués programant, es pot llegir i interpretar sense problemes.

En el següent exemple podem veure com el CI de Github no va acabar correctament a causa de problemes amb l'anàlisi. L'explicació dels errors és bastant intuïtiva, fins i tot arribant a dirigir-te al punt exacte emprant el IDE localment. A més, si algun d'aquests no és comprensible, flutter té una documentació on explica cadascun d'ells, donant fins i tot exemples de com evitar-ho.

El canvi més notori i que ens és molt útil va ser tenir el codi net de prints, llevar variables i importància en desús i mantenir uns noms homogenis.



```
> ✓ Set up job 4s
> ✓ Run actions/checkout@v2 1s
> ✓ Run subosito/flutter-action@v2 0s
> ✓ Run flutter pub get 4s
▼ ✗ Run flutter analyze 9s
  1 ▶ Run flutter analyze
  7 Analyzing pes_electrike...
  8
  9 info • Don't import implementation files from another package • lib/interficie/ctrl_presentation.dart:14:8 • implementation_imports
 10 info • The value of the local variable 'ctrlPresentation' isn't used • lib/interficie/main.dart:19:20 • unused_local_variable
 11 info • The value of the local variable 'usuari' isn't used • lib/interficie/main.dart:20:10 • unused_local_variable
 12 warning • The include file 'package:pedantic/analysis_options.1.8.0.yaml' in '/home/electrike/actions-runner/_work/pes_electrike/pes_electrike/lib/libraries/flutter_google_maps/analysis_options.yaml' can't be found when analyzing
 13 '/home/electrike/actions-runner/_work/pes_electrike/pes_electrike/lib/libraries/flutter_google_maps' •
 14 lib/libraries/flutter_google_maps/analysis_options.yaml:1:10 • include_file_not_found
 15
 14 4 issues found. (ran in 7.3s)
 15 Error: Process completed with exit code 1.
  ✓ Run flutter test --coverage 0s
  ✓ Upload coverage to Codecov 0s
> ✓ Post Run subosito/flutter-action@v2 0s
> ✓ Post Run actions/checkout@v2 0s
> ✓ Complete job 1s
```

En la següent imatge es pot observar com s'ha realitzat el workflow degudament sense errors.

```
> ✓ Set up job 5s
> ✓ Run actions/checkout@v2 1s
> ✓ Run subosito/flutter-action@v2 0s
> ✓ Run flutter pub get 4s
▼ ✓ Run flutter analyze 10s
  1 ▶ Run flutter analyze
  7 Analyzing pes_electrike...
  8 No issues found! (ran in 7.2s)
  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
```

## 4.6. Estratègia de proves

Definirem un conjunt de joc de proves limitat per tal d'anar fent testing automatitzat cada cop que hi ha un commit al GitHub, aquestes proves encara no estan implementades a GitHub.

Actualment, segons les tasques, fem mains per tal de testejar la nova funcionalitat. Si permet provar-la printant valors per consola, la persona que la ha implementat la funció pot veure el resultat i comprovar si funciona de manera correcta, aquestes són principalment les referents a domini i les de l'API. En el cas de que aquesta tasca es pugui provar directament en l'aplicació, com són principalment les de la interfície, s'implementen i es proven per diversos companys per tal de trobar errors i bugs o casos no contemplats que puguin dur al crash de l'aplicació. Ara per ara no hem trobat necessari implementar tests unitaris i no els preveiem desenvolupar en el futur.

L'últim mètode que utilitzem per provar tasques es el Postman. En aquest workspace comprovem que les funcions de l'API funcionen correctament i retornen la informació o valors esperats. Emprem sobretot aquest mètode a l'hora de fer els tests d'integració entre les capes per confirmar que la comunicació entre aquestes és correcta.

## 4.7. Gestió de configuracions

Per al moment actual de desenvolupament tenim diverses configuracions implementades.

La primera i més important seria el Continuous Integration, el qual executa proves de codi estàtic. Encara que tots els canvis es pugen correctament al github per a evitar perdre codi, al moment de realitzar merge entre branches, Github avisa que les corresponents branches poden tenir futurs problemes a causa d'aquests errors.

A continuació tindríem el Continuous Deployment, el qual ens permet fer un build automàtic tant de l'aplicació web com android perquè siguin fàcilment accessible des de Github.

Finalment, hem configurat dos ecosistemes de servidor.

El primer seria de testing que ho tenim allotjat en un servidor privat propi, encara que accedit a través de DDNS. Aquest mateix ho utilitzem per a desenvolupar les API, fer les primeres proves de noves implementacions, etc.

El segon seria AWS, el qual utilitzem més per a producció, és a dir, el que tindrà accés un usuari final.

D'altra banda, a causa d'un bug, vam haver d'implementar un servidor proxy per fer les peticions a una api de Google. Aquesta solució fa servir CORS-Anywhere conjuntament amb heroku perquè necessitem un domini amb certificat SSL.

## 4.8. Interacció amb companys

Aquesta secció fa referència a les APIs que hem d'usar d'un altre grup i la que hem de proporcionar nosaltres.

Pel que fa a les relacions d'oferta i consum de serveis a tercers ja les hem vist a l'apartat [Funcionalitats pactades](#).

Pel nostre projecte usarem l'API de la qualitat de l'aire, ja que creiem que pot encaixar amb la nostra app. D'altra banda, nosaltres proporcionarem els punts de recàrrega (i possiblement punts d'estacionament Bicing) més propers donada una localització concreta. Aquesta API s'ofereix al grup de cases sostenibles.

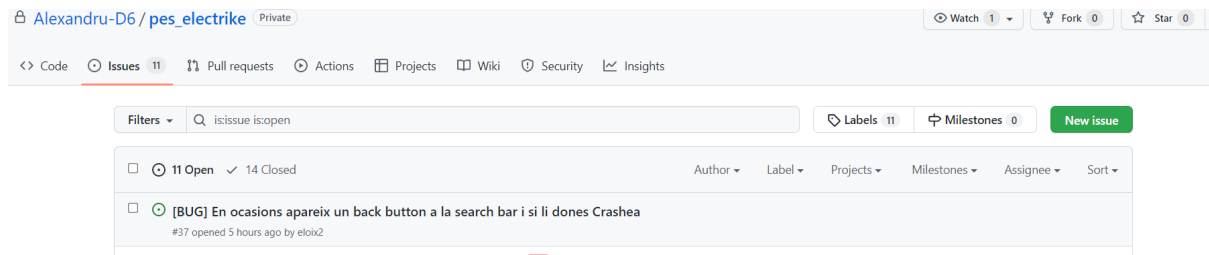
En aquest sprint ja hem completat aquestes relacions i hem integrat el servei ofertat per Happy Lungs dins el nostre sistema. Tot i haver obtingut l' API oferta per l'altre equip quasi al final de l'Sprint, finalment hem aconseguit integrar-lo a l'aplicació per generar les esmentades rutes ECO.

Aquest procés ha estat relativament senzill, ja que, amb les especificacions que van rebre per part nostra, ens retornen les dades esperades donada una coordenada i un radi. El resultat obtingut és un conjunt de coordenades amb el nivell de contaminació d'entre 1-5. Gràcies a aquesta informació, la nostra aplicació pot executar un algorisme per generar les rutes ecològiques calculant aquestes perquè passin pels punts menys contaminats. La documentació d'aquesta API la podem trobar a [l'annex](#) i visualitzar adientment a través de [Swagger](#).

Aquestes negociacions les fem a través d'un portaveu que és l'encarregat de comunicar-se amb els altres equips i comunicar les necessitats de la nostra aplicació, el Victor Asenjo.

## 4.9. Gestió dels bugs

La gestió dels bugs es fa mitjançant GitHub, on hi ha un log amb tots els bugs a la finestra d'issues i els anem solucionant de mica en mica, on a més fem una descripció d'aquests i què hem fet per solucionar-los a part d'assignar-los. D'aquesta manera tenim tots els problemes actuals agrupats en una única finestra i podrem fer un millor seguiment dels mateixos.



Aquests també es comenten en el grup per tal que l'equip sigui conscient dels bugs. A més que quan un integrant te temps i està implicat en la part que hi ha un bug ja sigui perquè ell mateix o ha programat o perquè es dedica a programar coses semblants i relacionades, ho intenta arreglar.

#### 4.10. Tractament RNFs

Els requisits no funcionals estan posats al Taiga, on allà anem modificant-los i adequant-nos a les necessitats. Alguns d'aquests RNFs les considerem com a User Stories senceres i, d'altres, els tenim apuntats en el llistat de requisits no funcionals i els tenim en consideració constantment com podrien ser:

- Requisits d'Aparença: on l'equip de desenvolupament ha de certificar que compleix els estàndards i patrons estètics acordats per tal que es vegi com a una marca unificada.
- Requisits de Personalització i Internacionalització: ens assegurem que l'usuari pugui canviar l'idioma de l'aplicació.
- Requisits d'Accessibilitat: les persones daltòniques no tenen cap impediment per utilitzar les funcionalitats principals de l'aplicació, ja que cap informació rellevant es mostra amb només diferenciació de colors. Les persones amb sordesa no tenen cap dificultat al fer servir l'aplicació, perquè cap informació es notifica per sorolls.
- Requisits d'Accés: els usuaris enregistrats disposaran de les funcionalitats al complet del producte, en canvi, un usuari que no ha iniciat sessió només podrà visualitzar el mapa, els punts de càrrega i Bicing.

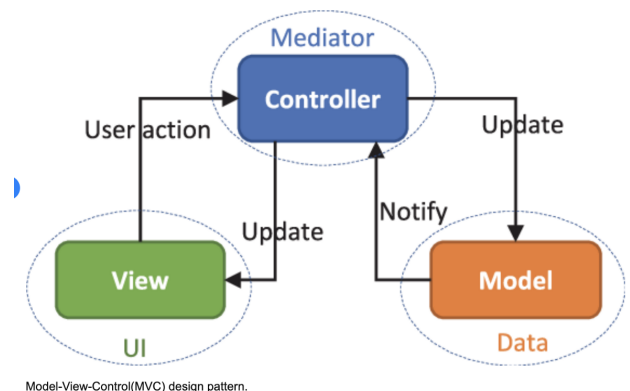
## 5. Descripció tecnològica

### 5.1. Concepció global de l'arquitectura

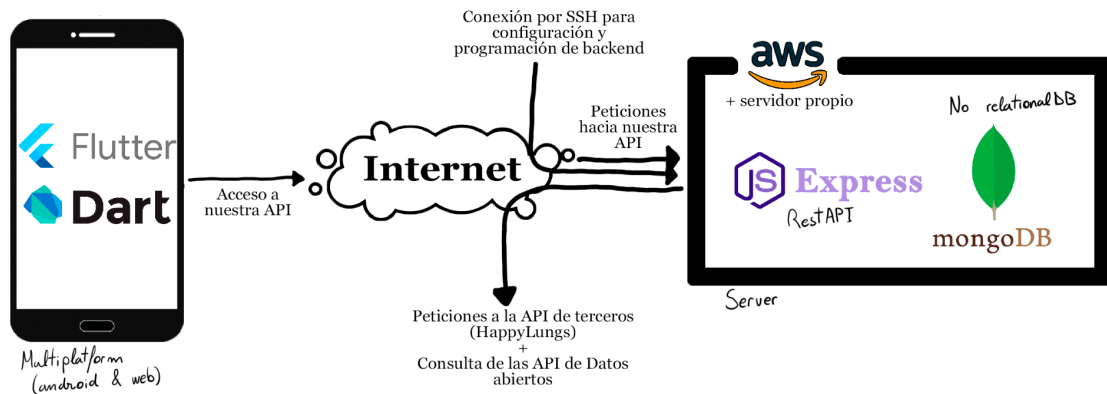
Quant a l'arquitectura del nostre programari podem destacar dos: MVC i disseny en 3 capes, on tots dos es complementen entre ells.

Per a començar explicaré el disseny en 3 capes: el nostre objectiu amb aquesta arquitectura era dividir el problema principal per a tenir de més petits, on podem diferenciar dràsticament el funcionament. Habitualment, i com l'hem utilitzat nosaltres, s'acostuma a dividir en interfície, domini i base de dades. D'aquesta manera, les persones que estan desenvolupant la lògica de tota l'aplicació no s'han de preocupar de com es mostrarà la informació o com s'emmagatzema la informació. El que ens ha permès aquesta tècnica és poder dividir el treball de manera eficient i evitar solapaments de companys durant el desenvolupament.

Pel que fa al MVC, ens permet interconnectar totes les capes, ja que sense ell, hi hauria molt d'acoblament entre les capes i, en el cas que s'hagin de fer canvis, seria molt complicat localitzar els canvis a fer. A més, fa que el codi quedi més net i, per consegüent, més mantenible.



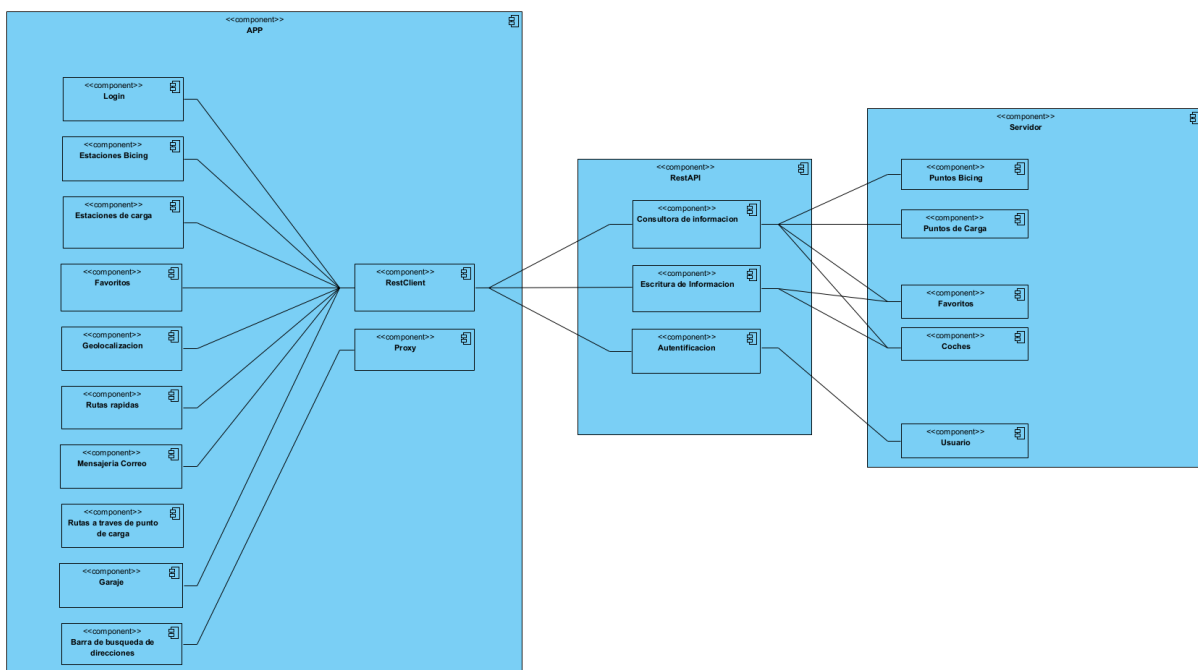
## 5.2. Altres diagrames d'arquitectura



En aquesta imatge es pot observar la nostra arquitectura física i tecnològica de l'aplicació. Començant amb la pròpia aplicació, la qual és multiplataforma tant en Android com en web.

Per a tota la persistència de dades, l'aplicació consulta el servidor a través d'una API pròpia. Actualment, el servidor està allotjat en un ordinador personal per a l'etapa de desenvolupament, encara que per al producte final es trasllada a AWS.

Quant al propi servidor, aquest implementa una API REST a través de ExpressJS connectada amb una base de dades no relacional MongoDB. A més d'oferir una API tant per a nosaltres com per a Build Green, també consulta tota informació varia d'altres API, entre elles la API de Happy Lungs.





En aquest diagrama de component es poden veure les principals funcionalitats de la nostra aplicació, API REST i de la base de dades.

### 5.3. Patrons de disseny aplicats

#### MVC

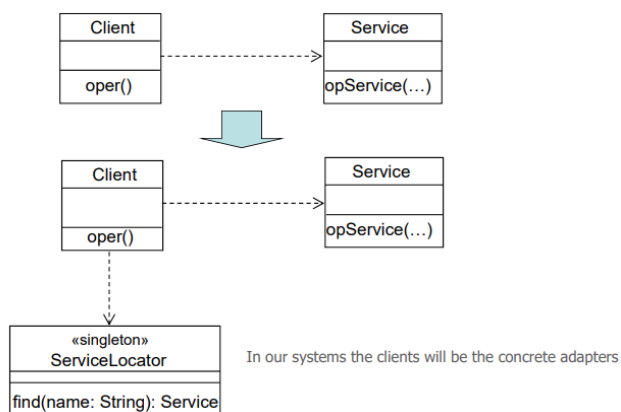
Com hem comentat anteriorment, hem dividit el sistema en 3 principals capes: Presentació, Domini i BD.

La idea és que la capa de presentació no “guardi” cap dada, sinó que s’encarregui només de demanar aquelles que les capes inferiors contenen i, mostrar aquesta a l’usuari.

Això s’aconsegueix implementant un controlador de Domini qui, després de rebre l’ordre de la capa de Presentació, obté les dades de la BD, les processa en el Domini i les envia a la capa de dades.

D’aquesta forma disminuïm l’acoblament entre capes, fent el codi més mantenible i reutilitzable.

#### ServiceLocator



La necessitat d'emprar aquest patró és semblant a l'anterior però, per disminuir l'acoblament amb els serveis. El nostre sistema hauria de comunicar-se amb un servei extern, el qual nosaltres no controlem i és fàcilment mutable.

En el cas que aquest servei decideixi canviar el tipus de dada que retorna o el nom de les seves funcions, seria molt complicat detectar els canvis que hauríem de fer en el nostre propi sistema.

Per tal d'evitar això, concentrem les crides al servei en classes adaptadores i, implementant un Service Locator, tenim aquí la col·lecció de serveis que utilitza el sistema.

```

1  import 'package:flutter_project/domini/services/google_login_adpt.dart';
2  import 'package:flutter_project/domini/services/google_maps_adpt.dart';
3  import 'package:flutter_project/domini/services/google_places_adpt.dart';
4  import 'package:get_it/get_it.dart';
5
6  final serviceLocator = GetIt.instance;
7
8  void setUpLocator() {
9    serviceLocator.registerLazySingleton<GoogleMapsAdpt>(() => GoogleMapsAdpt());
10   serviceLocator.registerLazySingleton<GoogleLoginAdpt>(() => GoogleLoginAdpt());
11   serviceLocator.registerLazySingleton<GooglePlaceAdpt>(() => GooglePlaceAdpt());
12 }
13
14
15 GoogleMapsAdpt get getMapsService {
16   return serviceLocator<GoogleMapsAdpt>();
17 }
18
19 GoogleLoginAdpt get getLoginService {
20   return serviceLocator<GoogleLoginAdpt>();
21 }
22
23 GooglePlaceAdpt get getPlaceService {
24   return serviceLocator<GooglePlaceAdpt>();
25 }
26

```

*Figura: Classe service\_locator.dart del nostre projecte*

## Singleton Pattern

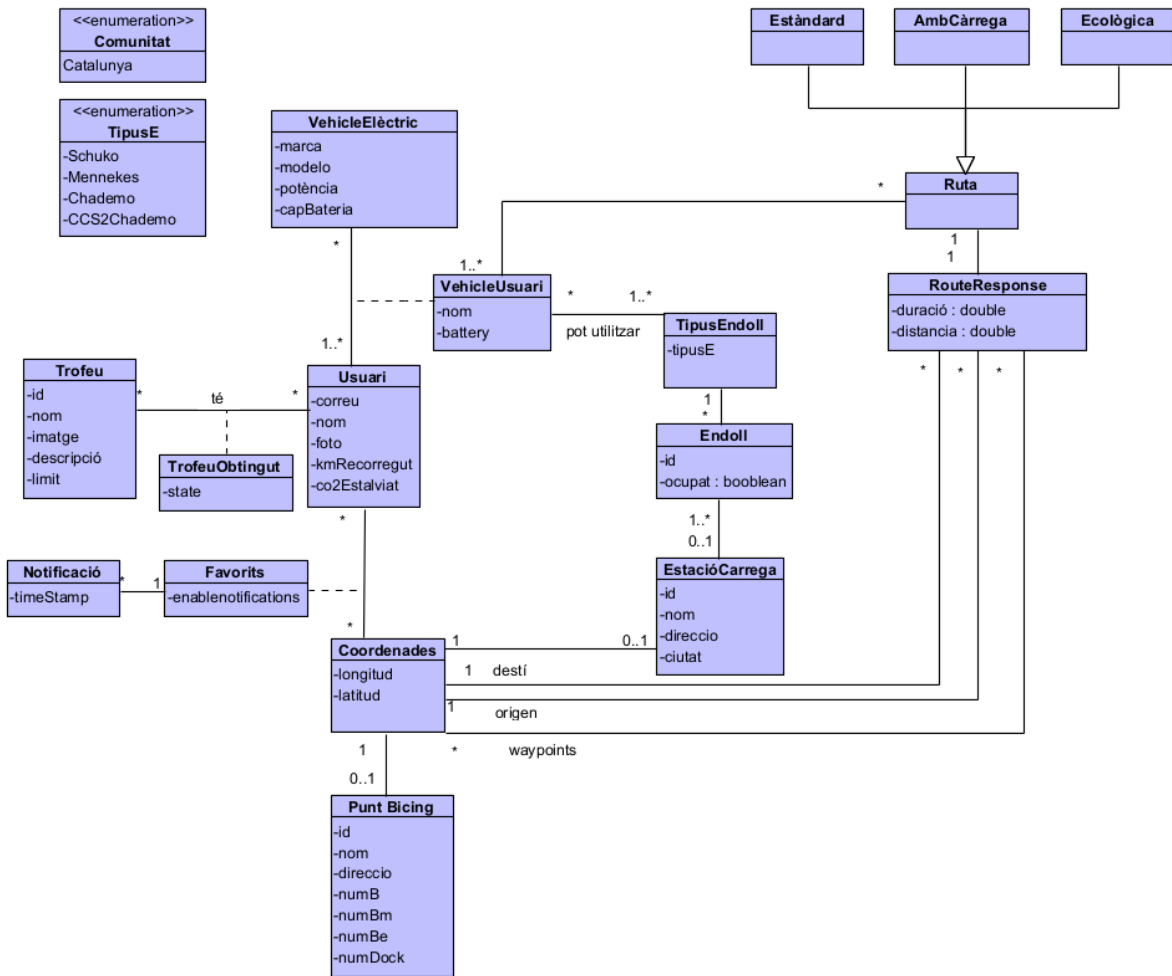
Un altre problema seria iniciar la sessió i assegurar-nos d'obtenir informació d'una mateixa instància de, per exemple, un usuari. En el cas de que volguessim fer actualitzacions en el conjunt sencer d'usuaris del sistema, voldriem només una sola copia d'aquesta. D'altra forma, podrien haver conflictes de sincronització.

Per aquesta raó, el Singleton Pattern s'utilitza per descriure que una classe pot tenir com a màxim una instància o cap (si no està inicialitzada). Tant el Servicelocator com els Controladors que hem vist anteriorment també compleix el patró Singleton perquè només tenen una sola instància.

```
class CtrlDomain {  
    static final CtrlDomain _singleton = CtrlDomain._internal();  
    factory CtrlDomain() {  
        return _singleton;  
    }  
    CtrlDomain._internal();  
}
```

## 5.4. Models de dades (UML)

El diagrama de classes UML de domini el podem trobar actualitzat a continuació. El que hem canviat al Sprint 3 són: les dades de trofeus i trofeu obtingut, hem deixat tres tipus de rutes només, hem eliminat els vehicles de combustible i, per tant, hem ajuntat la superclasse Vehicle i la subclasse Elèctric.

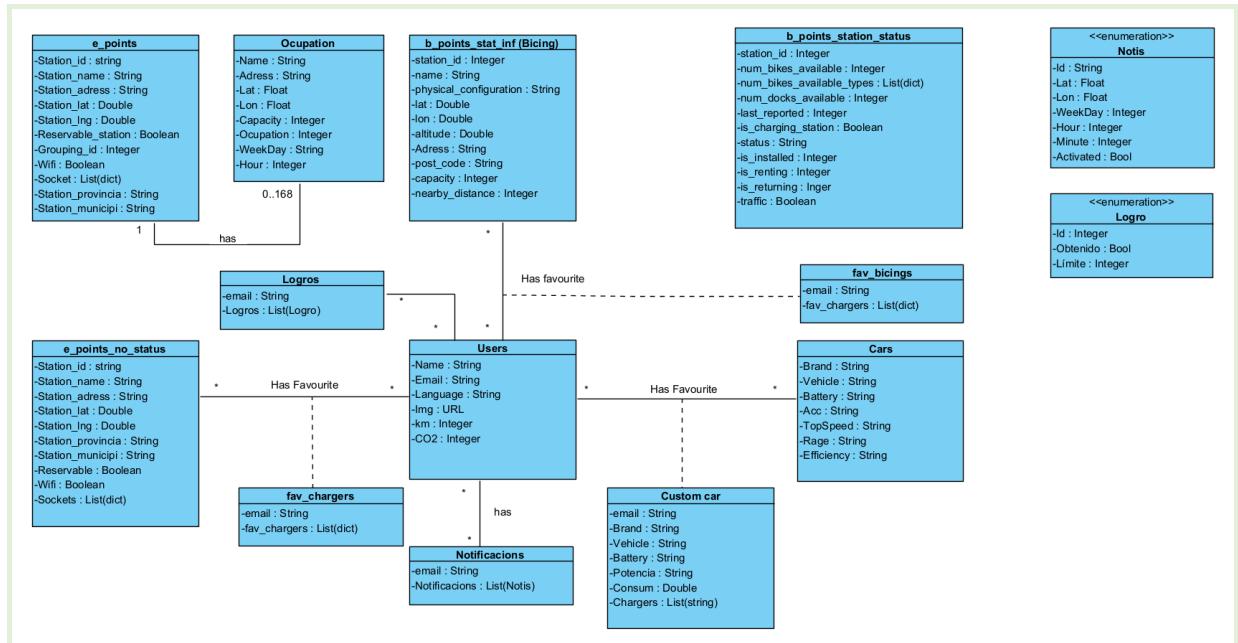


- Claus primàries: (VehicleElèctric, marca + model), (Usuari, correu), (Coordenades, longitud+latitud), (PuntBicing, id), (EstacióCàrrega, id), (Trofeu, nom), (Endoll, id), (TipusEndoll, tipus)
- Una ruta no pot tenir el mateix punt d'origen que de destí.
- Un usuari no pot utilitzar un endoll a una estació de càrrega en el que el seu vehicle elèctric no hi sigui compatible.

- L'usuari no pot tenir com a destinació una estació de càrrega amb el que el seu vehicle no tingui endoll compatible.
- Donada una coordenada que és un punt de Càrrega no pot ser d'un punt de Bicing ni viceversa.
- L'usuari només pot marcar com a favorit estacions de càrrega o punts Bicing.
- Una coordenada pot estar associada com a waypoint a una Ruta si la Coordenada és un punt de càrrega i aquest és compatible amb el vehicle de l'usuari, és a dir, que té els endolls que necessita el vehicle quan la Ruta és tipus amb Càrrega o de tipus Ecològica, o bé si és un punt ecològic de l'api d'Happy Lungs i la Ruta és de tipus Ecològica.

En el nostre cas, ens hem trobat un gran repte al plantejar l'esquema de la BBDD, ja que estem fent servir una BBDD no relacional, on s'organitza per documents, és a dir, no hi ha claus primàries ni relacions amb altres taules. No obstant, hem decidit representar-ho amb les relacions que fem servir nosaltres (tot i no ser de PK).

El diagrama de classes UML que descriu la base de dades és la següent (actualitzat segons sprint 3):



Com hem comentat anteriorment, no hi ha claus primàries com a tal, però si que nosaltres usem diversos atributs per tal d'identificar les dades, aquest són:

- (e\_points, Station\_lat + Station\_lng)
- (e\_points\_no\_status, Station\_lat + Station\_lng)
- (b\_points\_stat\_inf, lat +lon)
- (b\_points\_station\_status, lat +lon)
- (Users, email)
- (Cars, Brand+Vehicle)
- (Fav\_chargers, email)
- (Fav\_Bicings, email)
- (Custom\_car, email)
- (Ocupation, Lat+Lon)
- (Notificacions, email)
- (Logros, email)

## 5.5. Altres aspectes tecnològics

### Instrumentació

#### Nombre de servidors

Després d'estar fent servir durant les dues primeres setmanes un servidor AWS EC2 per gestionar les dades de la nostra aplicació, vam decidir canviar, ja que no ens estava donant els resultats esperats pel que fa a la possibilitat de desenvolupar codi remotament a través de SSH. L'Alexandru va decidir muntar ell un servidor local a casa seva, fent servir un portàtil amb una distribució Ubuntu Server 20.04 LTS. Des que utilitzem aquest nou servidor, no ens ha donat cap error i les peticions a ell són molt ràpides. No obstant això, per al producte final (producció) tindrem tots els nostres serveis d'API i DB en AWS EC2.

D'altra banda, a causa d'un bug, vam descobrir que necessitarem un servei de proxy per fer petició a API des de web. Per això, vam desplegar un petit script a heroku que fa de servidor proxy personal. Això no és possible fer-lo ni en els servidors Amazon ni el personal perquè no tindrem un domini web amb certificat SSL.

#### Configuració servidors

Dins el servidor hi haurà una base de dades MongoDB la qual ofereix un sistema NoSQL per emmagatzemar dades. Per l'altra part, també tenim implementat una Rest API amb el framework Express JS per tal d'accedir a les nostres dades de forma molt fàcil sense cap protocol ssh.

En quant a les dades obertes que usarem, el servidor serà l'encarregat de fer peticions per actualitzar l'estat de tots els punts de càrrega i de Bicing. D'aquesta manera, el dispositiu mòbil o la web poden consultar les dades post processades directament al servidor.

Adicionalment, per evitar fer ús dels minuts d'Actions que ofereix Github (workflows per realitzar CI/CD), ja que són bastant limitats. Vam traspasar tota la càrrega de treball al nostre servidor fent ús de Github com a intermediari. Això va ser

possible perquè Github ofereix aquesta opció, que es pot consultar a través d'aquest enllaç: [GitHub self hosted runners](#)

#### Nombre de BDs

Tenim una sola base de dades amb diverses taules, aquestes són:

- Usuaris
- Cotxes d'usuaris
- Carregadors elèctrics (estàtics)
- Carregadors elèctrics (dinàmics)
- Punts Bicing (estàtics)
- Punts Bicing (dinàmics)
- Cotxes elèctrics
- Punts preferits de Bicing
- Punts preferits de carregadors
- Ocupació
- Trofeus
- Notificacions

#### Versionat de BDs

Pel motiu d'estar utilitzant una base de dades no relacional com és MongoDB ens permet poder actualitzar les taules sense tenir la necessitat de migrar les dades a una nova taula. Per tant, això ens dona bastant llibertat a l'hora de decidir d'afegir un nou paràmetre. A més, fem servir una BD no relacional, ja que ens centrarem més en el rendiment que ens pot oferir per fer consultes i no tant en l'emmagatzematge de dades.



## Nombre de llenguatges

Principalment utilitzarem tres llenguatges diferents:

- Dart: per al desenvolupament general de l'aplicació Android, per a dispositiu mòbil, i per la pàgina web.
- NoSQL: per administrar les peticions que li arriben a la base de dades. - JavaScript: per a crear l'API de la nostra aplicació.
- Python: Per fer el script de recopilació de dades dels punts de recàrrega de Barcelona.

## APIs

### API oferida

La nostra API serà una integració per tota la base de dades, és a dir, tots els accessos al servidor, per part de l'aplicació, es farà a través d'aquesta amb l'objectiu d'evitar problemes de seguretat. D'aquesta manera evitem connectar-nos directament al servidor.

El nostre servei ofereix dues funcions. Aquestes funcions han estat acordades amb els membres de l'equip de *Build Green*. Oferim la possibilitat de consultar quines són les estacions de recàrrega de vehicles elèctrics i punts bicings propers. És a dir, donada una ubicació i un radi, dones aquest punts que estarien dins el cercle demanat.

Per obtenir més informació, dirigiu-vos a la [documentació d'aquesta API](#). Aquesta documentació està en castellà per petició de l'equip *Build Green*.

### API interna

Actualment, a nivell de gestió interna l'app recull i fa la petició de dades per mitjà d'una API desenvolupada pel Xavier Coll. La documentació està en **acabada i es pot trobar a l'annex d'aquest document.**

## Nombre d'APIs externes

- [Google Sign in](#)
- Google Maps [API](#)
- Directions API
- Places API

## Consum servei

El que consumim del servei que ens ofereixen és els punts de contaminació de Catalunya, dins d'un radi indicat i respecte a una coordenada, per poder calcular la ruta més ecològica donat un punt. La ruta més ecològica serà aquella que passi pels punts menys contaminats.

L'únic problema que hi ha amb aquest servei és el nombre limitat de coordenades que tenen, ja que no ens permet fer rutes bastant diverses i, per tant, no podem fer una ruta més exacta.

## Subministrament servei

Aquest servei es va oferir a finals de l'anterior sprint. Ha estat provat i funciona correctament. Quan l'altre equip va trobar un error, es va contactar amb nosaltres i en Xavier es va encarregar de solucionar-lo.

## Consum de dades obertes

Pel que fa a les dades obertes fem servir les següents:

- [Estat estacions de Bicing](#)
- [Informació sobre les estacions de Bicing](#)
- [Informació sobre tots els vehicles elèctrics](#)
- [Estacions de recàrrega per a vehicles elèctrics](#)
- [Informació de totes les marques de vehicles](#)
- [Informació de les estacions de recàrrega de vehicles elèctrics](#)

## Modificació de llibreries lliures

Durant aquest sprint vam anar modificant dues llibreries, ja que no disposaven de les funcionalitats necessàries per al nostre objectiu.

La primera és la de Google Map, encara que estigui bastant completa, li falten funcions com per exemple retornar informació sobre una ruta, poder crear rutes amb més d'un punt, poder crear clústers de marcadors de manera senzilla. Per aquesta raó vam importar aquesta llibreria i la vam afegir al nostre projecte per a poder-la modificar a les nostres necessitats.

D'altra banda, quant als clústers de marcadors, per a evitar haver de crear des de zero l'algorisme, utilitzem una llibreria que per a la plataforma d'Android funcionava com era d'esperar però en Web no. Per aquesta raó, la vam haver de modificar per a fer-la compatible amb la nostra solució.

Cal destacar que totes aquestes llibreries tenen llicències que permeten la seva modificació.

## Utilització d'eines de desenvolupament

### Ús de Frameworks

#### - Google Maps:

Aquest és usat per tal de poder tenir un mapa a l'app, consultar rutes, mirar punts de recàrrega...

#### - Flutter:

Al començar aquest projecte no teníem clar quina tecnologia fer servir per desenvolupar l'aplicació pels nostres requisits inicials, per tant, vam haver de fer un petit estudio sobre quines eines teníem disponibles. Al ser multiplataforma un dels requisits, la nostra gamma d'opcions es va reduir bastant degut a que no volíem haver de desenvolupar la versió web i la versió d'Android en paral·lel.

Per aquesta raó vam decantar-nos per Flutter o React, en veure el que oferia cadascun finalment vam escollir Flutter, ja que feia ús d'una sintaxi molt similar a la de C++(llenguatge que ja coneixíem prèviament). A més, en estar creat per Google, el vam veure un avantatge per utilitzar els serveis que ofereixen com per exemple el LogIn.

- Express JS:

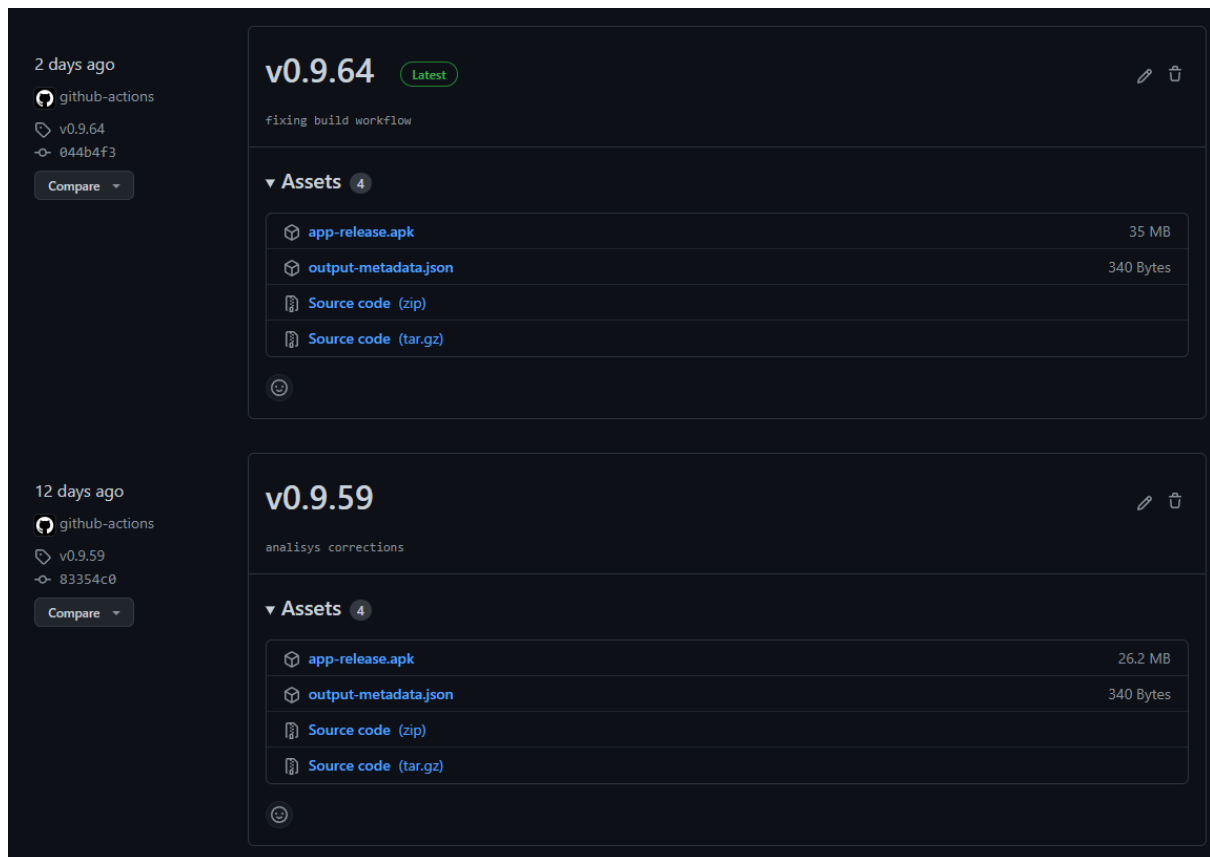
És un framework basat en node.js amb l'objectiu de crear una API que pot accedir a diversos bases de dades, entre elles MongoDB.

### Integració Continua

Pel que fa a la integració continua, tenim el Github automatitzat amb l'objectiu que amb cada push a una de les branques, tant principals com secundàries, estiguin sense errors i compleixin amb els requisits de qualitat. Per aconseguir tot això, farem servir les eines natives de flutter; flutter analyze. Tot això s'implementarà amb un workflow automatitzat de Github.

### Desplegament (deployment)

En quant al Deployment continuat, farem servir un workflow de Github per tal de tenir automatitzat la tasca de fer builds del projecte i penjar-les directament al repositori de Github. D'aquesta manera podrem tenir un seguiment de totes les versions de l'aplicació segons vagi avançant durant els diferents Sprints.



Tan mateix com el CI, utilitzarem l'eina native flutter build per fer això. Cal destacar

que es farà build tant de l'aplicació web com la d'android, amb la possibilitat de penjar la web per poder accedir públicament des de qualsevol lloc, encara que això està de moment en investigació per veure com funciona les Github Pages.

### Configuració del servidor

Per a començar, la nostra base de dades va estar allotjada en Amazon Web Services (AWS) ja que és un dels proveïdors més grans de servidors i volíem aprendre com funciona el seu ecosistema. D'altra banda, volíem evitar l'ús de Virtech per a no haver de bregar amb configurar la VPN en els mòbils.

Dit això, vam triar utilitzar el servei Elastic Compute Cloud (EC2) ja que ens oferia una màquina virtual amb un sistema operatiu tipus Ubuntu. La primera elecció va ser usar una de les ISO que oferia Amazon. Però desafortunadament, després d'intentar instal·lar els programes necessaris i la seva respectiva configuració, vam decidir tornar enrere i utilitzar Ubuntu ja que estem més familiaritzats amb això. Al final va resultar molt més senzill.

AWS ofereix aquest servei EC2 de manera gratuïta al llarg d'1 any conjuntament amb 30GB d'emmagatzematge, més que suficient per al nostre propòsit.

La primera configuració que vam haver de fer és la del propi servidor, en quant a processador vam triar t2.micro (el pla gratuït), configurat amb dos discos d'emmagatzematge; un d'ells per al programari necessari i l'altre dedicat solament per a la base de dades amb 15GB cadascun. Finalment el firewall perquè només nosaltres puguem accedir; deixant accés sobretot als ports 22 (SSH), 27017(MongoDB) i 3000(RestAPI).

Quant a l'últim punt de configuració del servidor, aquest va ser la clau ja que és el que permet connectar-se al servidor. La primera vegada el vam configurar perquè qualsevol IP pogués entrar al servidor tenint les corresponents credencials. La nostra sorpresa va ser que vam ser hackejats no només una, sinó que dues vegades, esborrant-nos les dades. Per sort encara no teníem res compromès ja que eren solament dades obertes, però ens va fer aprendre al fet que havíem de restringir l'accés. La part positiva d'aquesta experiència va ser que teníem a algú que comprovava les vulnerabilitats.

Per tant, la configuració de servidor quedaria així:

## Firewall

▼ Reglas de entrada				
Q Filtrar reglas				
ID de la regla del grupo ...	Intervalo de p...	Protocolo	Origen	Grupos de seguridad
sgr-012d16ff088ed68ed	3784	TCP	0.0.0.0/0	launch-wizard-1
sgr-089321fed5ae455bf	27344	TCP	0.0.0.0/0	launch-wizard-1
sgr-070ffef25b1b1f0f3	22	TCP	0.0.0.0/0	launch-wizard-1
▼ Reglas de salida				
Q Filtrar reglas				
ID de la regla del grupo ...	Intervalo de p...	Protocolo	Destino	Grupos de seguridad
sgr-00da5bc1a701ce1d0	Todo	Todo	0.0.0.0/0	launch-wizard-1

## Emmagatzematge

ID de volumen	Nombre del d...	Tamaño del vol...	Estado de la con...	Hora de conexión	Cifrado	ID de clave de KMS	Eliminar cuando termine
vol-058bfb75f54dbcc45	/dev/sda1	15	Asociado	Sat Mar 12 2022 17:01:54 ...	No	-	Sí
vol-04d66c5170b8274d2	/dev/sdb	15	Asociado	Sat Mar 12 2022 17:01:54 ...	No	-	Sí

## Configuració dels programes del servidor

### GIT

Per a tenir un còpia de seguretat de totes les dades, usem un repositori de git per a emmagatzemar tots els codis o script que usarem en el backend. Per començar també guardem les dades mongoDB com a redundància, la qual cosa ens va ajudar quan vam sofrir els atacs de seguretat.

En cas de no venir instal·lat el GIT es pot instal·lar de manera senzill amb:

1. `sudo apt-get install git`
- 2.

I per a accedir al repositori remotament vam decidir utilitzar una clau SSH, que en cas de no disposar ja d'una, per a generar-la s'utilitza les següents comandes:

1. ssh-keygen
- 2.

El següent pas ja seria en github.com, anant a la configuració general del compte, hi ha un apartat “SSH and GPG keys” on podràs afegir de manera senzilla la key prèviament copiada.

## MongoDB

Per a l'emmagatzematge de dades vam decidir usar MongoDB pel fet d'usar una base de dades no Relacional. Per a configurar-la seguim les instruccions oficials de la documentació.

<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-ubuntu/>

1. `wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -`
- 2.
3. `touch /etc/apt/sources.list.d/mongodb-org-5.0.list`
- 4.
5. `echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list`
- 6.
7. `sudo apt-get update`
- 8.
9. `sudo apt-get install -y mongodb-org`
- 10.
- 11.
12. `#Para iniciar/parar/reiniciar/estado el servicio de MongoDB`
13. `sudo systemctl start/stop/restart/status mongod`
- 14.
15. `#Para eliminar MongoDB`
16. `sudo service mongod stop`

```
17.  
18. sudo apt-get purge mongodb-org*  
19.  
20. #En caso de path default  
21. sudo rm -r /var/log/mongodb  
22. sudo rm -r /var/lib/mongodb  
23.
```

Cal destacar que segons en què servei cloud o PC local s'està utilitzant unes certes versions, les més noves per exemple, poden no funcionar. Per tant, s'hauria de fer un downgrade de versió.

Com hem esmentat anteriorment, volíem tenir una còpia de seguretat en GitHub en conseqüència vam haver de canviar el directori default de Mongo. Per a això vam haver de fer el següent:

```
1. #En el directorio destino  
2. mkdir database #nombre opcional  
3. sudo systemctl stop mongod  
4.  
5. #en caso de querer guardar las tablas actuales  
6. mv /etc/lib/mongo/* ./database/  
7.  
8. sudo chown mongod:mongodb -R ./database  
9.  
10.  
11. #hay que cambiar la configuración de mongoDB  
12. sudo nano /etc/mongod.conf  
13.  
14. #En el apartado "dbpath" hay que cambiarlo hacia el anteriormente creado
```



```
15. #p.e. ./database
```

Ara ja podem obrir de tornada el servidor i podrem crear les taules i usuaris necessaris. Per entrar al servidor i crear un usuari admin devem a través de terminal:

```
1. mongosh
2.
3. use admin
4.
5. db.createUser({user: "admin", pwd: "admin", roles:["root"]})
6.
```

Per a poder accedir de manera remota segura cal configurar de la següent manera l'arxivi mongod.conf:

```
1. #Volvemos a entrar en mongod.conf
2. Sudo nano /etc/mongod.conf
3.
4. #En el apartado bindIp deberemos agregar la DNS ipv4 publica del AWS justo después de añadir una coma
5. #Ademas, para mas seguridad en el sistema, podemos activar que para cualquier acceso se requiera de
   credencial, por ejemplo, el admin creado anteriormente.
6.
7. #Descomentamos la línea "#security:" y en la siguiente línea agregamos "authorization: enabled"
8.
```

He de destacar que per a tenir una major seguretat s'hauria de modificar el port default assignat al MongoDB.

A partir d'ara, per a accedir localment haurem de fer-ho de la següent manera:

```
1. Mongosh -u admin -p admin
```

I si volem accedir remotament es farà de la següent manera:

```
1. Mongosh mongodb://admin:admin@ipv4:27017/admin
```

Per a crear les taules de manera més senzilla hem utilitzat MongoDB Compass ja que és un IDE que permet gestionar les teves bases de dades remotament.

Express .js

Per a la part de la API personal, hem decidit utilitzar una RESTAPI ja que la nostra base de dades serà bastant estàtica al llarg del desenvolupament pel aquest motiu les peticions no canviaran massa.

Per a començar, hem de crear una carpeta on estarà allotjat tota la configuració i instal·lar tots programes necessaris:

```
1. mkdir API
2. cd API
3.
4. sudo apt update
5. sudo apt -y install curl dirmngr apt-transport-https lsb-release ca-certificates
6. curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
7. sudo apt -y install nodejs
8.
9. npm install -g n
10. npm install -g npm@8.5.4 #por si no se instalo la ultima versión
11. npm install express
12. npm install mongodb
13. npm install cors
14.
```

Una vegada fet això ja podem crear un fitxer .js i crear una API amb express.js amb mongoDB.

## Proxy CORS

En cas de tenir problemes amb CORS o peticiones HTML en general, una solució ràpida i senzilla es utilitzar una proxy. Encara que hi ha proxy gratuïtes, com Electrike podria ser una aplicació comercial, necessitem una solució pròpia. Per tant, vam haver de configurar un petit script de JS que s'està executant en un servidor heroku. Per a això vam haver de seguir els següents passos.

El primer pas és crear un nou repositori de GitHub per a poder fer push al servidor de Github. Dins d'aquest, es genera un fitxer index.js que allotgen el script de proxy, el qual s'utilitza la llibreria CORS-\*Anywhere. Fet això podem executar els següents comandos. (Partint de la base que ja tenim instal·lat Express .js de l'apartat anterior)

```
1. curl https://cli-assets.heroku.com/install-ubuntu.sh | sh
2. heroku login -i
3. heroku create
4.
5. sudo npm install cors-anywhere
6. sudo npm init
7.
8. git add .
9. git commit -m "first commit to heroku"
10. git push heroku main(master) #dependiendo de tu repositorio
11.
```

Una vegada fets tots aquests passos, la proxy serà l'enllaç a la web, que es podrà trobar en la dashboard de Heroku.

## 6. Referències

[1]. Documentació Flutter. Flutter. Consultada el 29 de març del 2022.

<https://esflutter.dev/docs/resources/faq>

[2]. Plantilla de Especificación de Requisitos Volere (Febrer 2006). James & Suzanne Robertson rectores del Atlantic Systems Guild. Consultada el 27 de març del 2022.

[https://www.volere.org/wp-content/uploads/2018/12/template\\_es.pdf](https://www.volere.org/wp-content/uploads/2018/12/template_es.pdf)

[3]. Java Script. Mozilla.org Individuales. Consultada el 28 de febrer del 2022.

<https://developer.mozilla.org/es/docs/Web/JavaScript>

[4]. MongoDB (2021). MongoDB. Consultada el 26 de febrer del 2022.

<https://www.mongodb.com/docs/>

[5]. Deloitte (2022). Deloitte Touche Tohmatsu Limited. Consultada el 27 de febrer del 2022.

<https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html>



# ANNEX

# DOCUMENTACIÓ API PRÒPIA

<b>USUARIS</b>	<b>76</b>
Inserir un usuari	76
Eliminar un usuari	76
Consultar la informació d'un usuari	77
Consultar l'idioma de l'aplicació d'un usuari	78
Canviar l'idioma de l'aplicació d'un usuari	78
Canviar el CO2 produït per un usuari	79
Canviar els KM conduïts per un usuari	79
Canviar el contador de cotxes d'un usuari	80
Canviar el contador de rutes d'un usuari	80
Comprovar si existeix un usuari	81
Obtenir el número de notificacions d'un usuari	81
Canviar el número de notificacions d'un usuari	81
<b>COTXES</b>	<b>82</b>
Obtenir tots els cotxes per defecte del sistema	82
Obtenir les marques de cotxes del nostre sistema.	82
Obtenir models de cotxes d'una marca.	83
Obtenir informació d'un vehicle	83
<b>CARREGADORS ELÈCTRICS</b>	<b>84</b>
Obtenir carregadors elèctrics de Catalunya.	84
Obtenir informació d'un carregador.	85
Obtenir informació d'endolls.	86
Obtenir carregadors elèctrics de Barcelona.	87
<b>BICINGS</b>	<b>88</b>
Obtenir tots els punts Bicing per defecte del sistema	88
Obtenir informació d'un punt bicing.	89
Obtenir informació d'un punt bicing donat un carrer.	90
Obtenir informació de l'estat d'un punt Bicing.	90
<b>TROFEUS</b>	<b>91</b>
Obtenir tots els trofeus d'un usuari.	91
Posar com a obtingut un trofeu d'un usuari.	92
Posar com a no obtingut un trofeu d'un usuari.	92
<b>NOTIFICACIONS</b>	<b>93</b>
Obtenir totes les notificacions d'un usuari.	93
Afegir una notificació a un usuari.	94
Esborrar una notificació d'un usuari.	94
Desactivar una notificació d'un usuari.	95
Activar una notificació d'un usuari.	95
<b>OCUPACIÓ</b>	<b>96</b>

Obtenir l'ocupació d'un punt de recàrrega.	96
<b>PUNTS BICING PREFERITS</b>	<b>97</b>
Afegir un punt Bicing preferit a un usuari.	97
Eliminar un punt Bicing preferit d'un usuari.	97
Obtenir tots els punts Bicing preferits d'un usuari	98
<b>CARREGADORS DE VEHICLES ELÈCTRICS PREFERITS</b>	<b>99</b>
Afegir un carregador preferit a un usuari.	99
Eliminar un carregador preferit d'un usuari.	99
Obtenir tots els carregadors preferits d'un usuari	100
<b>VEHICLES DELS USUARIS</b>	<b>101</b>
Afegir un vehicle propi a un usuari.	101
Eliminar un vehicle elèctric d'un usuari.	101
Obtenir tots els carregadors preferits d'un usuari	102
<b>DOCUMENTACIÓ SERVEI A BUILD GREEN</b>	<b>103</b>
Documentació API Happy Lungs	107



## 1. USUARIS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb els usuaris.

### a. Inserir un usuari

Aquesta funció permet introduir un usuari a la base de dades.

- Tipus de funció: POST
- Endpoint: /insert\_user
- Paràmetres:
  - name : String
  - email : String
  - img : String (el path de una imatge d'internet)
- Retorna: "User created correctly"
- Exemple de crida:  
`http://lanostraweb/insert_user?name=Joan&email=joan@gmail.com&img=hola.png`
- Impacte: Insereix l'usuari indicat a la BD i crea una entrada a les taules de: vehicles d'usuari, trofeus, notificacions, punts Bicing favorits i carregadors elèctrics preferits.

### b. Eliminar un usuari

Aquesta funció permet eliminar un usuari de la base de dades i del nostre sistema.

- Tipus de funció: POST
- Endpoint: /delete\_user
- Paràmetres:
  - email : String
- Retorna: "User deleted correctly"
- Exemple de crida: `http://lanostraweb/delete_user?email=joan@gmail.com`
- Impacte: Elimina l'usuari indicat a la BD i elimina totes les seves dependències de les altres taules.

### c. Consultar la informació d'un usuari

Aquesta funció permet consultar la informació d'un usuari determinat del nostre sistema.

- Tipus de funció: GET
- Endpoint: /user\_info
- Paràmetres:
  - email : String
- Retorna: Fitxer JSON amb aquest format:

```
{
  "items": [
    {
      "Name": "Nom Usuari",
      "Email": "correu@google.com",
      "Img":
        "https://lh3.googleusercontent.com/a-/AOh14GjxxXYQf35rDdDGqGUJLgZZMstmNvweN4
        OnQPO_",
      "km": 0,
      "CO2": 0,
      "cars_counter": 0,
      "routes_counter": 0,
      "Language": "ca",
      "lastIdCreated": 0
    }
  ]
}
```

- Exemple de crida: [http://lanostraweb/user\\_info?email=joan@gmail.com](http://lanostraweb/user_info?email=joan@gmail.com)
- Impacte: Cap

## **d. Consultar l'idioma de l'aplicació d'un usuari**

Aquesta funció permet consultar l'idioma de l'aplicació de l'usuari indicat que forma part del nostre sistema.

- Tipus de funció: GET
- Endpoint: /user\_language
- Paràmetres:
  - email : String
- Retorna: Fitxer JSON amb aquest format:

```
{  
  "items": "ca"  
}
```

El contingut dins de “items” pot ser: “ca” de català, “en” de anglès, “es” de espanyol. Si l'usuari no indica cap idioma de preferència, serà “en” per defecte.

- Exemple de crida: [http://lanostraweb/user\\_language?email=joan@gmail.com](http://lanostraweb/user_language?email=joan@gmail.com)
- Impacte: Cap

## **e. Canviar l'idioma de l'aplicació d'un usuari**

Aquesta funció permet canviar l'idioma de l'aplicació de l'usuari indicat que forma part del nostre sistema.

- Tipus de funció: POST
- Endpoint: /change\_language
- Paràmetres:
  - email : String
  - language: String (ha de ser “ca”, “en” o “es”)
- Retorna: Fitxer JSON amb aquest format:

```
“Language changed”
```

- Exemple de crida:  
[http://lanostraweb/change\\_language?language=ca&email=joan@gmail.com](http://lanostraweb/change_language?language=ca&email=joan@gmail.com)
- Impacte: Es canvia l'idioma de l'usuari indicat a la base de dades.

## **f. Canviar el CO2 produït per un usuari**

Aquesta funció permet canviar el CO2 produït per l'usuari indicat que forma part del nostre sistema.

- Tipus de funció: POST
- Endpoint: /change\_co2
- Paràmetres:
  - email : String
  - co2: Float
- Retorna: Fitxer JSON amb aquest format:

“CO2 changed”

- Exemple de crida:  
[http://lanostraweb/change\\_co2?co2=7.05&email=joan@gmail.com](http://lanostraweb/change_co2?co2=7.05&email=joan@gmail.com)
- Impacte: Es canvia el co2 de l'usuari indicat a la base de dades.

## **g. Canviar els KM conduïts per un usuari**

Aquesta funció permet canviar el CO2 produït per l'usuari indicat que forma part del nostre sistema.

- Tipus de funció: POST
- Endpoint: /change\_km
- Paràmetres:
  - email : String
  - km: Float
- Retorna: Fitxer JSON amb aquest format:

“KM changed”

- Exemple de crida:  
[http://lanostraweb/change\\_km?km=7.05&email=joan@gmail.com](http://lanostraweb/change_km?km=7.05&email=joan@gmail.com)
- Impacte: Es canvia els km de l'usuari indicat a la base de dades.

## **h. Canviar el contador de cotxes d'un usuari**

Aquesta funció permet canviar el número de cotxes que té l'usuari indicat.

- Tipus de funció: POST
- Endpoint: /change\_car\_counter
- Paràmetres:
  - email : String
  - num: Integer
- Retorna: Fitxer JSON amb aquest format:

“Data changed”

- Exemple de crida:  
[http://lanostraweb/change\\_car\\_counter?num=7&email=joan@gmail.com](http://lanostraweb/change_car_counter?num=7&email=joan@gmail.com)
- Impacte: Es canvia el número de cotxes de l'usuari indicat a la base de dades.

## **i. Canviar el contador de rutes d'un usuari**

Aquesta funció permet canviar el número de rutes que ha fet l'usuari indicat.

- Tipus de funció: POST
- Endpoint: /change\_routes\_counter
- Paràmetres:
  - email : String
  - num: Integer
- Retorna: Fitxer JSON amb aquest format:

“Data changed”

- Exemple de crida:  
[http://lanostraweb/change\\_routes\\_counter?num=7&email=joan@gmail.com](http://lanostraweb/change_routes_counter?num=7&email=joan@gmail.com)
- Impacte: Es canvia el número de rutes de l'usuari indicat a la base de dades.

## **j. Comprovar si existeix un usuari**

Aquesta funció permet comprovar si l'usuari indicat existeix al nostre sistema.

- Tipus de funció: GET
- Endpoint: /exist\_user
- Paràmetres:
  - email : String
- Retorna:
  - Si l'usuari existeix: Fitxer JSON amb les dades de l'usuari.
  - Si no existeix: Fitxer JSON buit.
- Exemple de crida: [http://lanostraweb/exist\\_user?email=joan@gmail.com](http://lanostraweb/exist_user?email=joan@gmail.com)
- Impacte: Cap

## **k. Obtenir el número de notifikacions d'un usuari**

Aquesta funció permet obtenir el número de notifikacions de l'usuari indicat.

- Tipus de funció: GET
- Endpoint: /not\_counter\_user
- Paràmetres:
  - email : String
- Retorna:
  - Fitxer JSON amb el número de notifikacions.
- Exemple de crida: [http://lanostraweb/not\\_counter\\_user?email=joan@gmail.com](http://lanostraweb/not_counter_user?email=joan@gmail.com)
- Impacte: Cap

## **l. Canviar el número de notifikacions d'un usuari**

Aquesta funció permet canviar el número de notifikacions de l'usuari indicat.

- Tipus de funció: POST
- Endpoint: /set\_not\_counter
- Paràmetres:
  - email : String
  - counter : Integer
- Retorna:
  - Fitxer JSON amb: "Notification counter changed"
- Exemple de crida:  
[http://lanostraweb/set\\_not\\_counter?counter=18&email=joan@gmail.com](http://lanostraweb/set_not_counter?counter=18&email=joan@gmail.com)
- Impacte: Es canvia el número de notifikacions de l'usuari indicat a la base de dades.

## 2. COTXES

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb els vehicles elèctrics.

### a. Obtenir tots els cotxes per defecte del sistema

Aquesta funció permet obtenir tot els cotxes que tenim per defecte definits al nostre sistema.

- Tipus de funció: GET
- Endpoint: /cars
- Paràmetres: None
- Retorna: Fitxer JSON amb aquest format:

```
{
  "_id": "628a69f8d1e8309b2b3b03d3",
  "Brand": "Dacia",
  "Vehicle": "Spring Electric",
  "Battery(kWh)": "26.8",
  "Acc(sec)": "15.0",
  "TopSpeed(Km/h)": "125",
  "Range(Km)": "170",
  "Efficiency(Wh/Km)": "158"
}
```

- Exemple de crida: <http://lanostraweb/cars>
- Impacte: Cap

### b. Obtenir les marques de cotxes del nostre sistema.

Aquesta funció permet obtenir les marques de cotxes del nostre sistema.

- Tipus de funció: GET
- Endpoint: /cars\_brands
- Paràmetres: None
- Retorna:  
Fitxer JSON amb un array de les marques.
- Exemple de crida: [http://lanostraweb/cars\\_brands](http://lanostraweb/cars_brands)
- Impacte: Cap

### c. Obtenir models de cotxes d'una marca.

Aquesta funció permet obtenir tots els models de cotxes d'una marca determinada.

- Tipus de funció: GET
- Endpoint: /cars\_models
- Paràmetres:
  - Brand
- Retorna:  
Fitxer JSON amb un array dels vehicles.
- Exemple de crida: [http://lanostraweb/cars\\_models?Brand=Tesla](http://lanostraweb/cars_models?Brand=Tesla)
- Impacte: Cap

### d. Obtenir informació d'un vehicle

Aquesta funció permet obtenir tots els models de cotxes d'una marca determinada.

- Tipus de funció: GET
- Endpoint: /car\_info
- Paràmetres:
  - Vehicle
- Retorna:  
Fitxer JSON amb la informació del vehicle:

```
{  
  "Brand": "Tesla",  
  "Vehicle": "Model S Plaid",  
  "Battery(kWh)": "90",  
  "Acc(sec)": "2.1",  
  "TopSpeed(Km/h)": "322",  
  "Range(Km)": "535",  
  "Efficiency(Wh/Km)": "168"  
}
```

- Exemple de crida: [http://lanostraweb/car\\_info?Vehicle=Model S Plaid](http://lanostraweb/car_info?Vehicle=Model S Plaid)
- Impacte: Cap



### 3. CARREGADORS ELÈCTRICS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb els carregadors de vehicles elèctrics. Cal destacar que dels carregadors de Barcelona, disposarem a més de la informació a temps real del seu estat.

#### a. Obtenir carregadors elèctrics de Catalunya.

Aquesta funció permet obtenir tots els carregadors de vehicles elèctrics de Catalunya.

**IMPORTANT:** No retorna els carregadors de Barcelona. Per això anar [aquí](#).

- Tipus de funció: GET
- Endpoint: /chargers\_cat
- Paràmetres: None
- Retorna:  
Fitxer JSON amb un array dels carregadors. Vista de un carregador:

```
{
  "_id": "628a6d7cd285247ae7ed2425",
  "Station_name": "Girona- Pl.Catalunya (S-W)",
  "Station_address": "Plaça Catalunya, 24",
  "Station_provincia": "Girona",
  "Station_municipi": "Girona",
  "Station_lat": 41.98145668149892,
  "Station_lng": 2.823227494955063,
  "Reservable": false,
  "Wifi": false,
  "Sockets": [
    {
      "Connector_id": 0,
      "Connector_types": "1",
      "Charge_modes": 3,
      "State": 6,
      "MaxChargingTime": null,
      "Slot_space": null,
      "Slot_level": null
    }
  ]
}
```

- Exemple de crida: [http://lanostraweb/chargers\\_cat](http://lanostraweb/chargers_cat)

- Impacte: Cap

## b. Obtenir informació d'un carregador.

Aquesta funció permet obtenir informació sobre un carregador indicat de Catalunya.

- Tipus de funció: GET
- Endpoint: /charger\_information\_cat
- Paràmetres:
  - latitud : Float
  - longitud : Float
- Retorna:  
Fitxer JSON amb el carregador. Vista de un carregador:

```
{
  "_id": "628a6d7cd285247ae7ed2425",
  "Station_name": "Girona- Pl.Catalunya (S-W)",
  "Station_address": "Plaça Catalunya, 24",
  "Station_provincia": "Girona",
  "Station_municipi": "Girona",
  "Station_lat": 41.98145668149892,
  "Station_lng": 2.823227494955063,
  "Reservable": false,
  "Wifi": false,
  "Sockets": [
    {
      "Connector_id": 0,
      "Connector_types": "1",
      "Charge_modes": 3,
      "State": 6,
      "MaxChargingTime": null,
      "Slot_space": null,
      "Slot_level": null
    }
  ]
}
```

- Exemple de crida:  
[http://lanostraweb/charger\\_information\\_cat?latitud=41.98145668149892&longitud=2.823227494955063](http://lanostraweb/charger_information_cat?latitud=41.98145668149892&longitud=2.823227494955063)
- Impacte: Cap

### c. Obtenir informació d'endolls.

Aquesta funció permet obtenir la informació dels endolls d'un punt de recàrrega de vehicles elèctrics.

- Tipus de funció: GET
- Endpoint: /plug\_info\_bcn
- Paràmetres:
  - latitud
  - longitud

- Retorna:

Fitxer JSON amb la informació dels carregadors d'un punt de recàrrega:

```
"items": [  
  {  
    "Connector_id": 2,  
    "Connector_types": "1",  
    "Charge_modes": "3",  
    "State": 0,  
    "MaxChargingTime": 120,  
    "Slot_space": null,  
    "Slot_level": null  
  },  
  {  
    "Connector_id": 1,  
    "Connector_types": "1",  
    "Charge_modes": "3",  
    "State": 0,  
    "MaxChargingTime": 120,  
    "Slot_space": null,  
    "Slot_level": null  
  }  
]
```

- Exemple de crida:  
[http://lanostraweb/plug\\_info\\_bcn?latitud=41.98145668149892&longitud=2.823227494955063](http://lanostraweb/plug_info_bcn?latitud=41.98145668149892&longitud=2.823227494955063)
- Impacte: Cap

#### **d. Obtenir carregadors elèctrics de Barcelona.**

Aquesta funció permet obtenir tots els carregadors de vehicles elèctrics de Barcelona

- Tipus de funció: GET
- Endpoint: /chargers\_bcn
- Paràmetres: None
- Retorna:  
Fitxer JSON amb un array dels carregadors de Barcelona..
- Exemple de crida: [http://lanostraweb/chargers\\_bcn](http://lanostraweb/chargers_bcn)
- Impacte: Cap

## 4. BICINGS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb els punts Bicing de Barcelona.

### a. Obtenir tots els punts Bicing per defecte del sistema

Aquesta funció permet obtenir tot els punts Bicing que tenim per defecte definits al nostre sistema.

- Tipus de funció: GET
- Endpoint: /bicings
- Paràmetres: None
- Retorna: Fitxer JSON amb aquest format:

```
{
  "_id": "628bba11316a19f682cdc40c",
  "station_id": 52,
  "name": "PL. SANT MIQUEL, 4",
  "physical_configuration": "ELECTRICBIKESTATION",
  "lat": 41.381708,
  "lon": 2.177292,
  "altitude": 11,
  "address": "PL. SANT MIQUEL, 4",
  "post_code": "08002",
  "capacity": 32,
  "is_charging_station": true,
  "nearby_distance": 1000
}
```

- Exemple de crida: <http://lanostraweb/bicings>
- Impacte: Cap

## b. Obtenir informació d'un punt bicing.

Aquesta funció permet obtenir informació sobre un punt bicing indicat de Catalunya.

- Tipus de funció: GET
- Endpoint: /bicing\_info
- Paràmetres:
  - latitud : Float
  - longitud : Float

- Retorna:

Fitxer JSON amb el punt Bicing. Vista d'un punt bicing:

```
"items": [  
  {  
    "station_id": 52,  
    "name": "PL. SANT MIQUEL, 4",  
    "physical_configuration": "ELECTRICBIKESTATION",  
    "lat": 41.381708,  
    "lon": 2.177292,  
    "altitude": 11,  
    "address": "PL. SANT MIQUEL, 4",  
    "post_code": "08002",  
    "capacity": 32,  
    "is_charging_station": true,  
    "nearby_distance": 1000  
  }  
]
```

- Exemple de crida:  
[http://lanostraweb/bicing\\_info?latitud=41.98145668149892&longitud=2.823227494955063](http://lanostraweb/bicing_info?latitud=41.98145668149892&longitud=2.823227494955063)
- Impacte: Cap

### **c. Obtenir informació d'un punt bicing donat un carrer.**

Aquesta funció permet obtenir informació sobre un punt bicing d'un carrer indicat de Barcelona.

- Tipus de funció: GET
- Endpoint: /bicing\_info\_street
- Paràmetres:
  - adress : String
- Retorna:  
Fitxer JSON amb el punt Bicing
- Exemple de crida: [http://lanostraweb/bicing\\_info\\_street?adress=](http://lanostraweb/bicing_info_street?adress=\)”Carrer Pere Girona”
- Impacte: Cap

### **d. Obtenir informació de l'estat d'un punt Bicing.**

Aquesta funció permet obtenir informació sobre l'estat un punt bicing.

- Tipus de funció: GET
- Endpoint: /bicing\_status
- Paràmetres:
  - longitud: Float
  - latitud : Float
- Retorna:  
Fitxer JSON amb l'estat del punt Bicing.
- Exemple de crida:  
[http://lanostraweb/bicing\\_status?latitud=41.98145668149892&longitud=2.823227494955063](http://lanostraweb/bicing_status?latitud=41.98145668149892&longitud=2.823227494955063)
- Impacte: cat

## 5. TROFEUS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb els trofeus dels usuaris de la nostra aplicació.

### a. Obtenir tots els trofeus d'un usuari.

Aquesta funció permet aconseguir tot els trofeus d'un usuari del nostre sistema.

- Tipus de funció: GET
- Endpoint: /get\_user\_logros
- Paràmetres:
  - email: String
- Retorna: Fitxer JSON amb aquest format:

```
"items": [  
  {  
    "id": 0,  
    "Obtenido": false,  
    "Limite": 1  
  },  
  {  
    "id": 1,  
    "Obtenido": false,  
    "Limite": 3  
  }  
  ....  
]
```

- Exemple de crida: [http://lanostraweb/get\\_user\\_logros](http://lanostraweb/get_user_logros)
- Impacte: Cap



### **b. Posar com a obtingut un trofeu d'un usuari.**

Aquesta funció permet indicar que un usuari ha aconseguit el trofeu en el nostre sistema.

- Tipus de funció: POST
- Endpoint: /modify\_logro
- Paràmetres:
  - email : String
  - id : String
- Retorna: Fitxer JSON amb aquest format:  
"Trophy changed to true"
- Exemple de crida: http://lanostraweb/modify\_logro
- Impacte: S'ha canviat el parametre 'Obtenido' de la base de dades a true.

### **c. Posar com a no obtingut un trofeu d'un usuari.**

Aquesta funció permet indicar que un usuari ha obtingut el trofeu en el nostre sistema.

- Tipus de funció: POST
- Endpoint: /remove\_logro
- Paràmetres:
  - email : String
  - id : String
- Retorna: Fitxer JSON amb aquest format:  
"Trophy changed to false"
- Exemple de crida: http://lanostraweb/remove\_logro
- Impacte: S'ha canviat el parametre 'Obtenido' de la base de dades a false.

## 6. NOTIFICACIONS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb el nostre sistema de notifikacions.

### a. Obtenir totes les notifikacions d'un usuari.

Aquesta funció permet aconseguir totes les notifikacions d'un usuari del nostre sistema.

- Tipus de funció: GET
- Endpoint: /get\_user\_notifications
- Paràmetres:
  - email: String
- Retorna: Fitxer JSON amb aquest format:

```
"items": [  
  {  
    "id": 0,  
    "Obtenido": false,  
    "Limite": 1  
  },  
  {  
    "id": 1,  
    "Obtenido": false,  
    "Limite": 3  
  }  
  ....  
]
```

- Exemple de crida: [http://lanostraweb/get\\_user\\_logros](http://lanostraweb/get_user_logros)
- Impacte: Cap

## **b. Afegir una notificació a un usuari.**

Aquesta funció permet inserir una notificació a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /insert\_notification
- Paràmetres:
  - email: String
  - id : String
  - lat : float
  - lon : float
  - day : integer
  - hour : Integer
  - minute : Integer
- Retorna: Fitxer JSON amb aquest format:  
“added notification to the user”
- Exemple de crida:  
[http://lanostraweb/insert\\_notification?email=hola@gmail.com&lat=3.0&lon=4.0&id=4&day=2&hour=4&minute=8](http://lanostraweb/insert_notification?email=hola@gmail.com&lat=3.0&lon=4.0&id=4&day=2&hour=4&minute=8)
- Impacte: S'afegeix una notificació d'un usuari a la taula de notificacions.

## **c. Esborrar una notificació d'un usuari.**

Aquesta funció permet eliminar una notificació a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /remove\_notification
- Paràmetres:
  - email: String
  - id : String
- Retorna: Fitxer JSON amb aquest format:  
“Removed notification to the user”
- Exemple de crida:  
[http://lanostraweb/insert\\_notification?email=hola@gmail.com&id=3](http://lanostraweb/insert_notification?email=hola@gmail.com&id=3)
- Impacte: S'elimina la notificació de l'usuari de la taula de notificacions.

#### **d. Desactivar una notificació d'un usuari.**

Aquesta funció permet desactivar una notificació a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /deactivate\_notification
- Paràmetres:
  - email: String
  - id : String
- Retorna: Fitxer JSON amb aquest format:  
"Changed"
- Exemple de crida:  
[http://lanostraweb/deactivate\\_notification?email=hola@gmail.com&id=3](http://lanostraweb/deactivate_notification?email=hola@gmail.com&id=3)
- Impacte: Es desactiva la notificació de l'usuari de la taula de notificacions.

#### **e. Activar una notificació d'un usuari.**

Aquesta funció permet activar una notificació a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /activate\_notification
- Paràmetres:
  - email: String
  - id : String
- Retorna: Fitxer JSON amb aquest format:  
"Changed"
- Exemple de crida:  
[http://lanostraweb/activate\\_notification?email=hola@gmail.com&id=3](http://lanostraweb/activate_notification?email=hola@gmail.com&id=3)
- Impacte: S'activa la notificació de l'usuari de la taula de notificacions.

## 7. OCUPACIÓ

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb el sistema d'ocupació dels punts de recàrrega de vehicles elèctrics.

### a. Obtenir l'ocupació d'un punt de recàrrega.

Aquesta funció permet aconseguir totes les notificacions d'un usuari del nostre sistema.

- Tipus de funció: GET
- Endpoint: /get\_occupation
- Paràmetres:
  - lat: float
  - lon: float
- Retorna: Fitxer JSON amb aquest format:

```
{
  "Adress": "C/ Marqués de Mulhacen, 51 B",
  "Capacity": 8,
  "Hour": 12,
  "Lat": 41.394408,
  "Lon": 2.114755,
  "Name": "Marquès de Mulhacén",
  "Occupation": 1,
  "WeekDay": "Tuesday"
},
{
  "Adress": "C/ Marqués de Mulhacen, 51 B",
  "Capacity": 8,
  "Hour": 5,
  "Lat": 41.394408,
  "Lon": 2.114755,
  "Name": "Marquès de Mulhacén",
  "Occupation": 0,
  "WeekDay": "Thursday"
}
...
}
```

- Exemple de crida: [http://lanostraweb/get\\_user\\_logros](http://lanostraweb/get_user_logros)
- Impacte: Cap

## 8. PUNTS BICING PREFERITS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb el sistema de punts Bicing preferits dels usuaris del nostre sistema.

### a. Afegir un punt Bicing preferit a un usuari.

Aquesta funció permet inserir un punt Bicing com a preferit a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /add\_fav\_bicing
- Paràmetres:
  - email: String
  - lat : float
  - lon : float
- Retorna: Fitxer JSON amb aquest format:  
"added fav\_point"
- Exemple de crida:  
[http://lanostraweb/add\\_fav\\_bicing?email=hola@gmail.com&lat=3.0&lon=4.0](http://lanostraweb/add_fav_bicing?email=hola@gmail.com&lat=3.0&lon=4.0)
- Impacte: S'afegeix un punt preferit de bicing a la taula de preferits de bicing.

### b. Eliminar un punt Bicing preferit d'un usuari.

Aquesta funció permet inserir un punt Bicing com a preferit a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /remove\_fav\_bicing
- Paràmetres:
  - email: String
  - lat : float
  - lon : float
- Retorna: Fitxer JSON amb aquest format:  
"removed fav\_point"
- Exemple de crida:  
[http://lanostraweb/removed\\_fav\\_bicing?email=hola@gmail.com&lat=3.0&lon=4.0](http://lanostraweb/removed_fav_bicing?email=hola@gmail.com&lat=3.0&lon=4.0)
- Impacte: S'elimina un punt preferit de bicing de la taula de preferits de bicing.

### c. Obtenir tots els punts Bicing preferits d'un usuari

Aquesta funció permet obtenir tot els punts Bicing preferits d'un usuari.

- Tipus de funció: GET
- Endpoint: /get\_user\_fav\_bicings
- Paràmetres:
  - email : String
- Retorna: Fitxer JSON amb aquest format:

```
{
  "_id": "628bba11316a19f682cdc40c",
  "station_id": 52,
  "name": "PL. SANT MIQUEL, 4",
  "physical_configuration": "ELECTRICBIKESTATION",
  "lat": 41.381708,
  "lon": 2.177292,
  "altitude": 11,
  "address": "PL. SANT MIQUEL, 4",
  "post_code": "08002",
  "capacity": 32,
  "is_charging_station": true,
  "nearby_distance": 1000
}
```

- Exemple de crida:  
[http://lanostraweb/get\\_user\\_fav\\_bicings?email=joan@gmail.com](http://lanostraweb/get_user_fav_bicings?email=joan@gmail.com)
- Impacte: Cap

## 9. CARREGADORS DE VEHICLES ELÈCTRICS PREFERITS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb el sistema de carregadors preferits dels usuaris del nostre sistema.

### a. Afegir un carregador preferit a un usuari.

Aquesta funció permet inserir un carregador elèctric com a preferit a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /add\_fav\_charger
- Paràmetres:
  - email: String
  - lat : float
  - lon : float
- Retorna: Fitxer JSON amb aquest format:  
"added fav\_point"
- Exemple de crida:  
[http://lanostraweb/add\\_fav\\_charger?email=hola@gmail.com&lat=3.0&lon=4.0](http://lanostraweb/add_fav_charger?email=hola@gmail.com&lat=3.0&lon=4.0)
- Impacte: S'afegeix un carregador elèctric a la taula de carregadors preferits.

### b. Eliminar un carregador preferit d'un usuari.

Aquesta funció permet inserir un carregador elèctric com a preferit a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /remove\_fav\_charger
- Paràmetres:
  - email: String
  - lat : float
  - lon : float
- Retorna: Fitxer JSON amb aquest format:  
"removed fav\_point"
- Exemple de crida:  
[http://lanostraweb/remove\\_fav\\_charger?email=hola@gmail.com&lat=3.0&lon=4.0](http://lanostraweb/remove_fav_charger?email=hola@gmail.com&lat=3.0&lon=4.0)
- Impacte: S'elimina un carregador elèctric de la taula de carregadors preferits.



### c. Obtenir tots els carregadors preferits d'un usuari

Aquesta funció permet obtenir tots els carregadors elèctrics preferits d'un usuari.

- Tipus de funció: GET
- Endpoint: /get\_user\_fav\_chargers
- Paràmetres:
  - email : String
- Retorna: Fitxer JSON amb aquest format:

```
{
  "_id": "628a6d7cd285247ae7ed2425",
  "Station_name": "Girona- Pl.Catalunya (S-W)",
  "Station_address": "Plaça Catalunya, 24",
  "Station_provincia": "Girona",
  "Station_municipi": "Girona",
  "Station_lat": 41.98145668149892,
  "Station_lng": 2.823227494955063,
  "Reservable": false,
  "Wifi": false,
  "Sockets": [
    {
      "Connector_id": 0,
      "Connector_types": "1",
      "Charge_modes": 3,
      "State": 6,
      "MaxChargingTime": null,
      "Slot_space": null,
      "Slot_level": null
    }
  ]
}
```

- Exemple de crida:  
[http://lanostraweb/get\\_user\\_fav\\_chargers?email=joan@gmail.com](http://lanostraweb/get_user_fav_chargers?email=joan@gmail.com)
- Impacte: Cap

## 10. VEHICLES DELS USUARIS

En aquesta part de la documentació de l'API, comentarem els mètodes que estan relacionats amb el sistema de vehicles propis dels usuaris del nostre sistema.

### a. Afegir un vehicle propi a un usuari.

Aquesta funció permet inserir un vehicle elèctric propi a la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /insert\_car\_user
- Paràmetres:
  - email: String
  - name : String
  - brand : String
  - vehicle : String
  - battery : String
  - efficiency : String
  - chargers : String
- Retorna: Fitxer JSON amb aquest format:  
"added user's car"
- Exemple de crida:  
[http://lanostraweb/add\\_fav\\_charger?email=hola@gmail.com&name=Jaone&brand=Tesla&vehicle=Model 3&battery=56&efficiency=12&chargers=0010](http://lanostraweb/add_fav_charger?email=hola@gmail.com&name=Jaone&brand=Tesla&vehicle=Model 3&battery=56&efficiency=12&chargers=0010)
- Impacte: S'afegeix un vehicle elèctric a la taula de vehicles d'usuari.

### b. Eliminar un vehicle elèctric d'un usuari.

Aquesta funció permet eliminar un vehicle elèctric de la BD del nostre sistema donat un usuari.

- Tipus de funció: POST
- Endpoint: /remove\_car\_user
- Paràmetres:
  - email: String
  - vehicle\_id : String
- Retorna: Fitxer JSON amb aquest format:  
"removed user's car"
- Exemple de crida:  
[http://lanostraweb/remove\\_car\\_user?email=hola@gmail.com&vehicle\\_id=1](http://lanostraweb/remove_car_user?email=hola@gmail.com&vehicle_id=1)
- Impacte: S'elimina un vehicle elèctric de la taula de vehicles d'usuari.

### c. Obtenir tots els carregadors preferits d'un usuari

Aquesta funció permet obtenir tots els vehicles elèctrics d'un usuari.

- Tipus de funció: GET
- Endpoint: /get\_cars\_user
- Paràmetres:
  - email : String
- Retorna: Fitxer JSON amb aquest format:

```
[  
{  
  "_id": "628a69f8d1e8309b2b3b03d3",  
  "Brand": "Dacia",  
  "Vehicle": "Spring Electric",  
  "Battery(kWh)": "26.8",  
  "Acc(sec)": "15.0",  
  "TopSpeed(Km/h)": "125",  
  "Range(Km)": "170",  
  "Efficiency(Wh/Km)": "158"  
}  
]
```

- Exemple de crida: [http://lanostraweb/get\\_cars\\_user?email=joan@gmail.com](http://lanostraweb/get_cars_user?email=joan@gmail.com)
- Impacte: Cap

# DOCUMENTACIÓ SERVEI A BUILD GREEN

La API ofrece los puntos cercanos de Bicing y cargadores eléctricos de Barcelona. Dado una ubicación (latitud y longitud) y una distancia en kilómetros se devuelve un listado en formato JSON con la información de los puntos que se haya escogido (bicing o cargadores). A continuación, voy a hacer una explicación de como usarla. Esta documentación se ha desarrollado en castellano por petición del equipo Green Build.

## BICINGS CERCANOS

Esta función te permite conseguir en un formato JSON todos los bicings cercanos dada una ubicación y una distancia. Dada la distancia, te va a dar todos los puntos Bicings que estén dentro del radio establecido.

Para hacer una petición de esta función, mejor os pongo un ejemplo: Se tiene que hacer una petición **GET** con este URI:

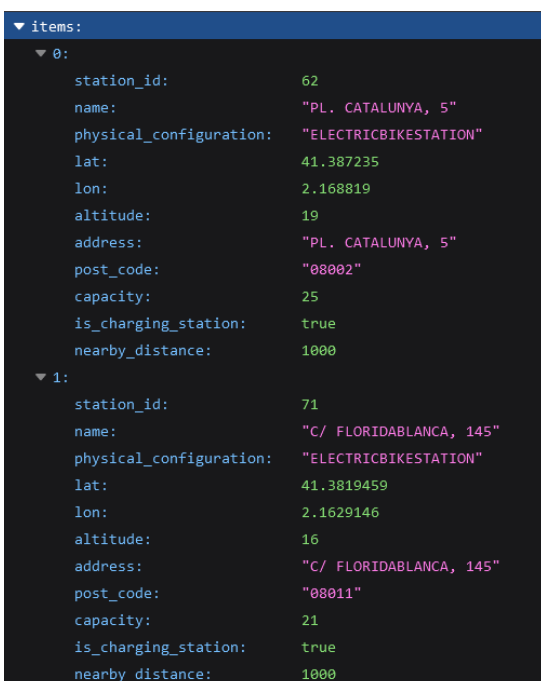
[http://electricke.ddns.net:3784/near\\_bicings?lat=41.385503&lon=2.1634768&dist=0.5](http://electricke.ddns.net:3784/near_bicings?lat=41.385503&lon=2.1634768&dist=0.5)

Esta función devuelve todos los bicings cercanos dada la ubicación que vemos en lat y lon, que están dentro del radio de distancia 0.5 km (500 metros).

Es muy importante que los parámetros estén en minúscula. Probadlo en postman antes de ponerlo dentro del código.

Los parámetros que tiene que tener la URI es:

- [http://electricke.ddns.net:3784/near\\_bicings](http://electricke.ddns.net:3784/near_bicings)
- lat : La latitud de la ubicación. Cuantos más decimales, más preciso. Es importante que no haya ninguna coma, siempre poner los decimales separados por puntos. Ej: lat=41.0234 lat=43.23456 lat=12.3434
- lon : La longitud de la ubicación. Cuantos más decimales, más preciso. Es importante que no haya ninguna coma, siempre poner los decimales separados por puntos. Ej: lon=31.0234 lon=23.23456 lon=12.3434
- dist : Distancia máxima de los bicings cercanos que se desea. La distancia es en kilómetros. Si se quiere poner decimales, importante que sean separados por puntos. Ej: dist=0.5 dist=1 dist=4



Esto es un ejemplo de una respuesta de la API.

Todo se engloba en un items, y a partir de allí, es un array.

Podéis ver lo que incluye una estación de bicing en la imagen.



## CHARGERS CERCANOS

Esta función te permite conseguir en un formato JSON todos los cargadores cercanos dada una ubicación y una distancia. Dada la distancia, te va a dar todos los cargadores que estén dentro del radio establecido.

Para hacer una petición de esta función, mejor os pongo un ejemplo: Se tiene que hacer una petición **GET** con este URI:

[http://electricke.ddns.net:3784/near\\_chargers?lat=41.385503&lon=2.1634768&dist=5](http://electricke.ddns.net:3784/near_chargers?lat=41.385503&lon=2.1634768&dist=5)

Esta función devuelve todos los cargadores cercanos dada la ubicación que vemos en lat y lon, que están dentro del radio de distancia 5 km.

Es muy importante que los parámetros estén en minúscula. Probadlo en postman antes de ponerlo dentro del código.

Los parámetros que tiene que tener la URI es:

- [http://electricke.ddns.net:3784/near\\_bicings](http://electricke.ddns.net:3784/near_bicings)
- lat : La latitud de la ubicación. Cuantos más decimales, más preciso. Es importante que no haya ninguna coma, siempre poner los decimales separados por puntos. Ej: lat=41.0234 lat=43.23456 lat=12.3434
- lon : La longitud de la ubicación. Cuantos más decimales, más preciso. Es importante que no haya ninguna coma, siempre poner los decimales separados por puntos. Ej: lon=31.0234 lon=23.23456 lon=12.3434
- dist : Distancia máxima de los cargadores cercanos que se desea. La distancia es en kilometros. Si se quiere poner decimales, importante que sean separados por puntos. Ej: dist=0.5 dist=1 dist=4



Esto es un ejemplo de una respuesta de la API.

Todo se engloba en un items, y a partir de allí, es un array.

Dentro de Sockets, tenemos los diferentes cargadores que ofrece el punto de cargar.

Podéis ver lo que incluye una estación de carga en la imagen.

## *Documentació API Happy Lungs*



