

# Portal de creació i publicació de guies.

## Model de dades

BORJA IBAÑEZ CUMBRE 1460915

**Resum** — A moltes persones els hi agrada realitzar excursions habitualment, quasi sempre necessiten veure guies per plantejar la jornada i anar de forma segura.

Per aquest motiu, va sorgir una empresa anomenada GuideMeUp dedicada a l'àmbit turístic amb la missió principal d'aportar al sector turístic, cultural i oci un producte i servei. Fa uns anys, va desenvolupar un portal web que hi havia un conjunt de guies, rutes i llocs amb molta informació al respecte. En un moment donat, es va produir un problema de compatibilitat del llenguatge de programació amb el sistema operatiu.

GuideMeUp ha demanat realitzar un projecte que consti en desenvolupar un nou portal web. Aquest utilitza la mateixa base que portal web que ja tenia en producció, i s'inclourà l'opció de crear guies personals per guardar l'experiència de l'usuari. A més a més, es desenvolupa amb noves versions de llenguatge de programació per no tenir els problemes de compatibilitat que es van produir.

La UAB ha dividit el projecte en dos TFG diferents. Un s'encarrega de desenvolupar la part del Front-End i un altre la part del Back-End. En aquest informe s'explicarà el TFG orientat al desenvolupament del Back-End.

El Back-End consisteix en crear una nova estructura de dades però guardar-hi tota la informació de guies, rutes, llocs, etc. Per enllaçar la Base de Dades amb el Front-End, es desenvoluparà una API (Interfície de Programació per Aplicacions).

Un cop acabat el projecte, el Front-End utilitzarà la API per consultar o guardar informació que l'usuari desitja.

**Paraules clau**— Guidemeup, Back-End, Java, Base de Dades, Cassandra, Docker, Aplicació, Guia, Ruta, Lloc.

**Abstract**— Many people like to on excursions regularly, they almost always need to see guides to plan the day and go safely. For this reason, a company called GuideMeUp dedicated to the field of tourism emerged with the main mission of providing a product and service to the tourism, worship and leisure sector. A few years ago, he developed a web portal that had a set of guides, routes and places with a lot of information about it. At one point, there was a problem with the compatibility of the programming language with the operating system.

GuideMeUp has asked to carry out a project that consists of developing a new web portal. It uses the same database as the web portal it already had in production, and will include the option to create personal guides to save the user experience. In addition, it is developed with new versions of programming language so as not to have the compatibility issues that occurred.

The UAB has divided the project into two different TFGs. One is in charge of developing the Front-End part and the other is the Back-End part. This report will explain the TFG aimed at the development of the Back-End.

The Back-End consists of creating a new data structure but storing all the information of guides, routes, places, etc. To link the Database to the Front-End, an API (Application Programming Interface) will be developed.

Once the project is complete, the Front-End will use the API to query or save information that the user wants.

**Index Terms**— Guidemeup, Back-End, Java, Database, Cassandra, Docker, Application, Guide, Route, Place.



## 1 INTRODUCCIÓ - CONTEXT DEL TREBAL

LES guies online sorgeixen després que les persones necessitessin informació per orientar en diferents llocs que desconeixien. Google Maps és una de les guies més generals que s'han fet més famoses. WikiLoc en canvi, és una guia més específica orientat al àmbit de senderisme o excursions.

Aquestes guies tenen molta informació al respecte, però no contenen l'opció de crear guies personals per guardar les experiències viscudes en cada una. Per aquest motiu, va sorgir GuideMeUp[1], una empresa dedicada a l'àmbit turístic amb la missió principal d'aportar al sector turístic, cultural i oci un producte o servei:

- Facilitar la creació i publicació de guies de contingut.
- Ajudar als visitants i clients del sector a gaudir i guardar la seva experiència del viatge

Aquesta empresa tenia un portal web en funcionament però va tenir un problema de compatibilitat de la versió del llenguatge de programació amb el sistema operatiu.

Per aquest motiu, GuideMeUp va aprofitar l'oportunitat de crear un portal amb moltes més funcions, una d'elles, la incorporació de guies personals. Com s'ha comentat, aquestes guies contindran informació que l'usuari vulgui guardar per en recordar-se de l'experiència.

La UAB va rebre l'opció de participar en la creació del portal, s'ha dividit aquest projecte en dos TFGs. Un TFG desenvoluparà la part del Front-End i l'altre TFG s'encarregarà del Back-End.

En aquest document explica el procés de desenvolupament del Back-End del portal format per una Base de Dades i una interfície de programació per aplicacions (API). Una Api que rebrà les peticions dels usuaris a través del Front-End i les executarà consultant la BD i retornant la informació necessària per satisfer a l'usuari.

Durant el desenvolupament d'aquest informe es parlarà sobre els objectius del projecte (apd.2), l'estat l'art (apd.3), la metodologia que s'ha seguit (apd.4), la planificació (apd.5) i desenvolupament (apd.6), alguns testos (apd.7), i finalment els resultats obtinguts (apd.8) y una sèrie de conclusions (apd.9) i agraïments (apd.10).

## 2 OBJECTIUS

Per satisfer les necessitats del client/front-end, s'han marcat un objectius principal.

- Desenvolupar un Back-End per donar servei al Front-End. generals i específics.

Per aconseguir l'objectiu principal, s'ha definit objectius més específics:

- Creació d'una base de dades capaç d'emmagatzemar tota la informació necessària pel bon funcionament del sistema.
- Definir i implementar una interfície de programació d'aplicacions, coneguda com API.
- Realitzar una càrrega massiva amb totes les guies existents de Guidemeup.

- Implementar un mecanisme d'instal·lació de distribució àgil i senzill.
- Realitzar una segona càrrega massiva d'informació, però en aquest cas, serà informació d'altres fonts d'internet.

Tots aquests objectius, s'han realitzar segons la prioritat demanada per l'empresa.

Objectius del projecte		
ID	Objectiu	Prioritat
1	Creació d'una Base de Dades	Alta
2	Definir i implementar una API	Alta
3	Càrrega massiva de totes les guies de Guidemeup	Alta
4	Implementar el mecanisme d'instal·lació de distribució àgil i senzill.	Mitja
5	Càrrega massiva d'informació externa.	Baix

Taula 1: Objectius del projecte

Per realitzar el projecte de forma més senzilla, cada objectiu té assignades deferents tasques. Aquestes s'exploraran en l'apartat de Planificació del Projecte.

## 3 ESTAT DEL PROJECTE INICIAL

GuideMeUp disposava d'un portal web amb molta informació sobre les guies o rutes. El Font-End tenia implementat un mapa on es podia veure els diferents llocs de les guies, assenyalats amb un punter. També tenia una llista per poder filtrar tot el contingut segons les necessitats de l'usuari.

A part del portal web, també disposa de varies aplicacions android. Cada aplicació conté una de les guies de tot el contingut que disposava GuideMeUp. Per exemple, hi ha la possibilitat de descarregar-se una guia de Montserrat[2] de Google Play.

El portal web funcionava correctament fins que el navegador va canviar de versió, aquesta ja no era compatible amb la versió que utilitzaven en el portal. En canvi, les aplicacions android no van se perjudicades per problema de compatibilitat.

Degut Aquest problema l'empresa vol tornar a desenvolupar el portal amb les noves versions de llenguatges existents. Al final d'aquest projecte desenvolupat s'espera tenir el portal en funcionament.

La Base de Dades del portal estava construïda a partir de les següents taules:

Entitat
Identificador
Nom
Descripció
Adreça
Coordenades
Preu
Horari
Autor
URL
Imatge
Tags
Icona
Tipus
Categoria
Logotip
Públic o Privat

Entitat 1

Relació
Id Entitat Origen
Id Entitat Destí
Tipus
Pes

Entitat 2



Imatge 1: Exemple del sprint acabat el 16 d' Abril

A la *Entitat 1* tenim una taula anomenada Entitat, aquesta contenia tots els atributs necessaris per inserir qualsevol tipus d'entitat. Depenent el tipus d'entitat s'utilitzaven tots els paràmetres o no.

A la *Entitat 2* tenim una altre taula anomenada relació, aquesta conté només els atributs necessaris, l'identificador d'origen i final, el tipus i el número de relacions que té. És a dir, si parlem d'una guia, una guia conté molts llocs, cada relació de guia amb lloc tindrà un pes diferent, des de 0 fins la quantitat total de llocs de la guia.

Pel Back-End d'aquest projecte, s'utilitza una estructura de Base de Dades similar canviant alguns atributs, per no estar utilitzant la mateixa taula per totes les entitats. També s'utilitzarà tot el contingut de guies i rutes d'aquesta.

A continuació s'explicarà tot el desenvolupament del projecte en qüestió.

#### 4 MÈTODE DE DESENVOLUPAMENT

Per portar a terme la gran quantitat de treball de manera organitzada, aquest es dividirà en diferents blocs entre dos o tres setmanes. Cada bloc serà un sprint on es realitzaran diferents tasques que es tindran que desenvolupar durant el sprint.

Al final de cada sprint es realitzarà un reunió per comentar les tasques que s'han fet o per resoldre qualsevol dubte sorgit.

Per tenir un control de totes les tasques, s'aniran publicant a l'aplicació Trello[3].

Aquí podem veure la *Imatge 1*, un exemple d'una sprint en el Trello. En concret, al sprint del 16 d'abril. En aquest, es poden veure les tasques que hi havia assignades. En el meu cas, tenia assignada la tasca d'implementació de la Base de dades. Les altres tasques corresponen al company que s'encarrega del Front-End i al tutor per part de l'empresa.

Tenint en compte la data de finalització del projecte, i les tasques necessàries, s'han realitzat sprints cada dues setmanes on en cada un, es realitzaran unes tasques específiques.

A la taula *Taula 2* podem veure tots els sprints amb cada una de les tasques assignades:

Sprint	Tasca	Final
1	Definició de la API + exemples	18/03/21
2	Determinació de Proves de validació i Determinació de la tecnologia de la BD	04/03/21
3	Instal·lació del SW de la BD + API	18/04/21
4	Implementació del model de la BD	25/04/21
5	Implementació de la indexació de la BD	02/05/21
6	Programació de la API	16/05/21
7	Testeig unitari de la API	30/05/21
8	Càrrega de dades a la BD	06/06/21
9	Proves d'integració	13/06/21
10	Paquetització del servidor	21/06/21

Taula 2: Sprints amb les tasques corresponents

## 5 PLANIFICACIÓ DEL PROJECTE

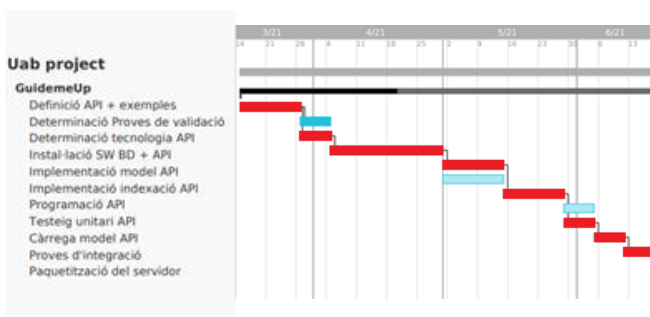
Com s'ha pogut veure en l'apartat anterior, cada sprint tenia un seguit de tasques assignades, aquestes tasques eren les següents:

- Definició de la API + exemples:  
A partir de l'estructura que ja es disposava, s'ha redefinit els mètodes que es volien utilitzar. Aquests mètodes eren els següents:
  - Mètodes CRUD (Create, Read, Update i Delete).
  - Mètodes de filtre, a partir d'alguns paràmetres, ens retornes totes les entitats que compleixin amb aquests.
- Determinació de Proves de validació:  
Definir les proves de validació per verificar que la API que s'haurà desenvolupat, compleixi amb tots els requisits.
- Determinació de la tecnologia de la BD:  
Depenent dels mètodes que s'hagin d'implementar a la API es necessitarà un tipus de Base de Dades o un altre.
- Instal·lació de SW de la BD + API:  
Un cop ja tenim tot definit, s'instal·larà tota l'estructura per començar a implementar tot.
- Implementació del model de la BD:  
En aquesta tasca, es crearà tota la estructura de la Base de Dades segons hem determinat en les funcions de la API.
- Implementació de la indexació de la BD:  
Per poder realitzar cerques grans, amb qualsevol atribut per tota la informació de la BD, es necessari indexar els atributs de les taules. En aquesta tasca, s'indexaran els atributs que s'utilitzen en les cerques de grans dimensions.
- Programació de la API:  
Aquesta tasca, és la més complicada, s'implementa totes les funcionalitats de la API. Els mètodes CRUD i els mètodes de filtre d'informació.
- Testeig unitari de la API:  
A partir de les proves de validació que s'han definit en les primeres tasques, determinar si ha d'haver-hi alguna modificació en el codi, o ja es pot donar com acabat.
- Càrrega de dades a la BD:  
En aquest punt, un cop ja està tot implementat, es l'hora de càrrega totes les dades de la BD del portal antic de GuideMeUp.
- Proves d'integració:  
Són proves per veure si el Front-End treballa de forma correcte amb el Back-End.

- Paquetització del servidor:  
Un cop finalitzat el desenvolupament, s'ha de guardar la BD i la API en un fitxer per tenir un portal molt més dinàmic.

Totes aquestes tasques explicades, s'han de desenvolupar en un ordre en concret, hi ha tasques que es poden realitzar conjuntament perquè no tenen res en comú. Altres que necessiten els resultats de la tasca anterior.

Amb tot això podem realitzar un diagrama de Gantt amb l'aplicació *TeamGantt*[4].



Imatge 2: Diagrama de Gantt

En la imatge 2 es pot veure totes les tasques amb l'ordre corresponent. A la dreta de la imatge podem observar en vermell el camí crític del diagrama, aquest les formen les tasques essencials i obligatòries per tenir el projecte desenvolupat en les dades corresponents.

## 6 DESENVOLUPAMENT DEL PROJECTE

En aquesta secció s'explicarà en detall les tasques que han anat realitzant-se durant el projecte.

### 6.0 Eines i Llenguatge

#### 6.0.1 Java

Java [5], és un llenguatge de programació orientat a objectes, independe de la plataforma hardware on es desenvolupa. Com a llenguatge de programació, ofereix un codi robust i una utilització automàtica de la memòria, cosa que redueix el nombre d'errors.

Actualment es troba entre els millors remunerats en la indústria de la programació.

En el projecte, s'utilitzarà aquest llenguatge per desenvolupar el back-end del portal, és a dir, per a la creació i implementació de la API. A més a més, també s'implementarà un conjunt de test per a comprovar el seu bon funcionament.

#### 6.0.2 Apache Cassandra

Cassandra[6], és un sistema de gestió de base de dades distribuïdes de codi obert dissenyat per manipular grans quantitats de dades en diversos servidors, oferint una alta disponibilitat sense un únic punt de fallada.

En el projecte, s'utilitza Cassandra per guardar les dades (guies, rutes, etc.) que es gestionaran en el portal.

S'ha escollit aquesta eina perquè és una base de dades molt ràpida en lectura i en la cerca. Principalment en el projecte es realitzaran operacions de cerca/lectura per mostrar tota la informació demanada per l'usuari.

### 6.0.3 Docker

Docker [7] és un sistema operatiu per a contenidors. El motor Docker s'instal·la en el servidor en el que es desitja executar contenidors i proporciona un conjunt de comandos senzilles que s'utilitzen per crear, iniciar o detenir contenidors.

En el projecte, s'ha utilitzat docker per instal·lar el servidor apache de Cassandra. Aquest facilitava la inicialització del servidor o la tancament d'aquest.

### 6.0.4 Swagger

Swagger[8] són una sèrie de regles, especificacions i eines que ens ajuden a documentar les APIs. Ens ajuda a crear documentació per a que tothom ho entengui.

En el projecte, l'utilitzem per crear i documentar la interfície de la API. Més endavant, quant estigui la API programada amb Java, es podrà utilitzar aquest interfície de API per realitzar proves de funcionament entre la API i la BD.

## 6.1 Aprenentatge

Per desenvolupar el projecte, s'han estudiat diferents servidors de Base de Dades. Com per exemple *MongoDB*[9], *Cassandra*[6], *Solr*[10], *HBase*[11], *Neo4j*[12].

Per cada servidor, hem analitzat diferents paràmetres: velocitat d'escriptura, velocitat de lectura, velocitat de cerca, ordenació de columnes, segmentació o escalat i si s'adaptava al llenguatge Java.

En els documents lliurats, podem trobar un fitxer anomenat "*Comparativa de BBDD.pdf*" on hi ha tota la informació trobada sobre les diferents característiques que tenen els servidors respecte els paràmetres que necessitaven.

Finalment, hem triat Cassandra per la Base de Dades i el llenguatge Java per a la implementació de la API. Cassandra per que és un servidor que permet realitzar cerques amb respostes molt ràpides, a més a més, utilitza el mètode de clau-valor per guardar les dades.

De Java ja es tenia un ampli coneixement però de Cassandra no hi havia. Va ser necessari dedicar una gran quantitat de temps revisant per internet diferents mètodes per instal·lar el servidor i com realitzar les consultes.

## 6.2 Captació de requisits i elaboració de la planificació

Per a la realització dels requisits del portal, es va dur a terme una primera reunió el 9 de febrer del 2021. En aquesta hi va assistir el tutor del TFG i també un tutor per part de l'empresa Guidemeup.

En aquesta reunió es va presentar el projecte amb totes les característiques desitjades. A més a més, es va ensenyar un exemple de portal per a telefons androids incloent el seu funcionament.

A partir de les conclusions extretes de la reunió inicial, es van poder extreure els requisits necessaris i els objectius més importants.

Un cop obtinguda tota la informació sobre com es vol que sigui el projecte, es va realitzar una aproximació de la planificació, des de l'inici fins al final del projecte.

## 6.3 Creació de la Base de Dades

Amb els requisits recollits, i amb el model definit de la Base de Dades es van formar diferents taules/entitats:

- **Usuari:** Taula en la que es guarda la informació de les persones. D'aquestes es guarden les dades bàsiques com el nom, l'email, i la contrasenya encriptada.
- **Entitat:** Aquesta taula guarda totes les dades dels punts d'interès, com restaurants, monuments arquitectònics, edificis emblemàtics, etc. També es guarda la informació sobre les imatges i de les rutes.
- **TipusEntitat:** En aquesta entitat, és de tipus enum, és a dir, només es guarden els tipus que pot esdevenir una entitat. Per exemple: Guia, Lloc, Group, Multimèdia, Ruta.
- **GuiaPersonal:** Taula on es guarda un conjunt d'entitats escollides per l'usuari. De tal manera, que l'usuari pot crear una guia per poder-la utilitzar en qualsevol moment.
- **EnllaçEntitat:** Taula que guarda les relacions entre entitats. Per exemple: ruta = entitat -> entitat. O també una entitat pot tenir una imatge.

L'arquitectura que s'ha escollit per implementar la base de dades ha sigut un servidor de Cassandra sobre un contenidor Docker[7].

Un Docker realitza la funció de contenidor, i en ell, es pot instal·lar qualsevol servidor de forma ràpida i senzilla.

La part problemàtica a l'hora d'instal·lar el servidor, ha sigut instal·lar el Docker. Un cop instal·lat, Cassandra ha sigut bastant fàcil de posar-ho en funcionament.

## 6.4 Definir i implementar la API

Abans de començar a implementar l'API, ha calgut definir quines necessitats té el front-end per tal de donar una bona resposta al client.

Per definir l'API s'ha utilitzat una aplicació web anomenada *Swagger*[8]. Aquesta aplicació permet definir les entitats, els enllaços entre ells i realitzar consultes a una base de dades local.

S'ha creat un miniprojecte online que permet visualitzar l'esquema de tots els atributs de totes les taules necessàries, a més provar els mètodes a una base de dades molt bàsica.

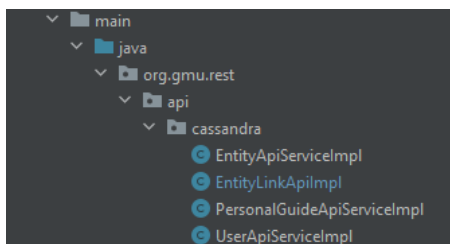
Aquest miniprojecte[13] està creat sobre el servidor de

Guidemeup, es poden veure les diferents entitats amb els seus esquemes corresponents i totes les funcions necessàries per cada una. Tot i que en la pagina de Swagger ja estan explicades, a continuació les detallarem:

- Entitat:  
Es defineixen les funcions CRUD (create, read, Update, delete). A més a més, també es defineix una funció que donat un seguit de atributs de la taula entitat ens retornen totes les entitats que els compleixen.
- Usuari:  
Només s'implementarà el mètode get, perquè es vol que les persones facin el login mitjançant el sistema de google. Per tant, només es necessari el mètode per saber quin usuari està utilitzant el portal.
- GuiaPersonal:  
La guia personal mètodes diferents als que té la taula Entitat. Es defineixen les funcions CRUD (create, read, Update i delete) i també un mètode que ens retorna totes les guies personals que l'usuari a creat.
- EnllaçEntitat:  
Es defineix el mètode Update, create i un mètode per filtrar els enllaços. Per exemple, el mètode de filtratge retorna totes les entitats que pertanyen a una mateixa guia, a un mateix grup, o a una mateixa ruta.

Un cop ja es tenia una bona definició de la API, s'ha continuat amb el projecte implementant la interfície amb el llenguatge Java.

Primer de tot, es va crear un projecte amb totes les classes necessàries per dur a terme el treball.



Imatge 3: Organització dels fitxers de la API

En la *Imatge 3* podem observar tres directoris diferents però les classes són exactament les mateixes, l'única diferència es la implementació de cada una.

A "cassandra" estan totes les classes implementades apuntant al servidor de Cassandra.

A "dummy" hi ha els mètodes de les classes implementats però apunten al servidor antic que tenia Guidemeup.

I "impl" les classes són abstractes, tenen els mètodes necessaris però no la implementació.

Com s'ha comentat, al directori de "Cassandra", hi ha totes les classes que s'han creat per implementar tots els mètodes definits prèviament.

El fitxer "BaseCassandra" és una classe singleton ja que no es poden interposar les peticions al servidor. També hi ha implementat la connexió al servidor mitjançant una IP i un port determinat. Per defecte Cassandra té el port 9042.

El fitxer "EntityApiServiceImpl" implementar tots els mètodes de la taula entitat definits en la API. Principalment implementar les funcions CRUD a més de la funció de filtratge. Aquesta funció, rep com a paràmetre atributs de l'entitat. Els atributs estan organitzats en llistes. Al final, retorna totes les entitats que tenen algun atribut dels paràmetres.

El fitxer "EntityLink" implementar les funcions que enllacen les entitats. A més de la funció del filtre. Aquesta simplement retorna les entitats involucrades d'origen o de destí.

En el fitxer "PersonalGuideApiServiceImpl" conté tots els mètodes necessaris per implementar les funcions definides prèviament en la API. Només té les funcions CRUD perquè no es necessari cap filtre.

Finalment en "UserApiServiceImpl" és la classe d'usuari, aquesta conté un mètode per saber quin usuari ha iniciat sessió.

## 7 TEST

A mesura que es va implementant la API, s'ha anat creant els diferents tests per comprovar el bon funcionament de les classes. Aquests s'han dividit en dos blocs.

El primer són testos comprovant cada mètode si el seu funcionament si és l'esperat i desitjat.

El segon bloc, són testos més generalitzats reproduint casos d'exemples pel front-end. En aquests, s'utilitzen els diferents mètodes implementats de la API.

## 8 RESULTATS

Un cop acabada la creació de la BD Cassandra i la implementació de la API. Podem passar a explicar els resultats obtinguts de tot el treball descrit en aquest document. Els resultats són els següents:

- S'ha construït un entorn amb una base de dades en Cassandra, capaç d'emmagatzemar tota la informació i retornar les peticions en forma de resposta ràpida i eficaç.
- També s'ha creat una API, capaç de realitzar totes les peticions que el Font-End demana de forma ràpida i organitzada. Prèviament,
- Un cop desenvolupat el projecte, es va realitzar una carga massiva amb tota la informació existent a altres guies de l'empresa.

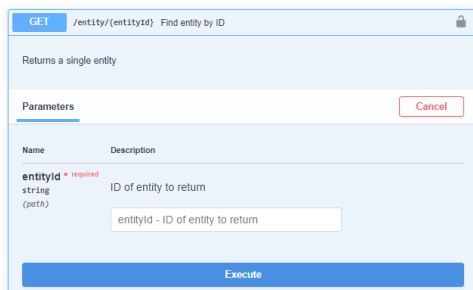
El projecte s'ha desenvolupat amb el temps previst, però no ens ha donat temps a unir el Front-End amb el Back-End. Més endavant, l'empresa s'encarregarà d'unir els dos.

Per tant el resultats, els ensenyarem a partir de la definició de la API, mitjançant el projecte amb *Swagger*[8] on podíem realitzar consultes a la Base de Dades de Cassandra sense la necessitat de tenir una interfície de Front-End.



Imatge 4: Interfície Swagger

Com podem veure a la *Imatge 4*, és la interfície de Swagger que es va crear prèviament per definir la API. Aquesta conté tots els mètodes que hem implementat. Per tant, quan s'utilitza algun mètode d'aquests, crida a la implementació amb Java que està connectat amb Cassandra.



Imatge 5: Exemple de funció get

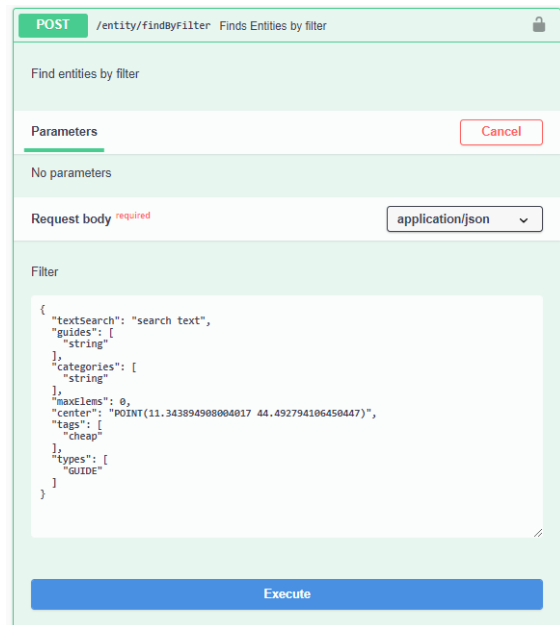
En la *imatge 5* podem veure un exemple del Swagger.

Aquesta pertany al mètode get de la taula Entitat. Com a paràmetre li enviem un identificador i ens retorna la fila de la taula que pertany la entitat.

Unes de les funcions més importants del Back-End, és el filtre d'entitats per atributs. Com podem veure a la *imatge 6* l'usuari introdueix els paràmetres que vol filtrar:

Pot donar una llista de categories i ens retorna totes les entitats que tenen aquestes categories, o el mateix amb el tipus d'entitat.

També podem filtrar amb una cerca, l'usuari introdueix el text que vol buscar y la Api ens retorna totes les entitats que tenen aquest text en les propietats de l'entitat.



Imatge 6: Exemple de funció EntityFilter

S'ha de dir, que hi ha algunes tasques no s'han finalitzat degut a la falta de temps i a la dificultat que comporta. Aquestes tasques són:

- Càrrega massiva d'informació de fonts externes.
- Proves d'integració Front-End amb Back-End.
- Paquetització del servidor.

## 9 CONCLUSIONS

En aquest projecte hem fet una versió avançada sobre el portal que volia desenvolupar Guidemeup. A partir del temps disponible per a realitzar-lo, es van acordar unes necessitats bàsiques, de les quals es tenien que entregar com a base per donar una escalabilitat en un futur.

El portal actual, compleix amb les necessitats acordades, a més a més, a partir de varies iteracions en el projecte, s'ha gestionat d'una forma molt còmoda per a l'organització segons l'estil del treball.

En relació als aspectes tècnics del portal, el desenvolupament de la API sobre una BD, la converteix en una aplicació accessible des de qualsevol front-end.

Valoro positivament l'experiència de desenvolupar un Back-End sobre una Base de Dades, ja que he entès com funciona el procés de desenvolupament d'un projecte. Des del recull dels requisits fins al final, fins tindre completament implementada la API.

A part d'entendre tot el procés, he tingut que aprendre com implementar una aplicació API i com crear una Base de Dades Cassandra des zero.

[10] Solr. (2021). Consultat el 12 / 04 / 2021, a <https://solr.apache.org/>

[11] HBase. (2021). Consultat el 12 / 04 / 2021, a <https://hbase.apache.org/>

[12] Neo4j. (2021). Consultat el 12 / 04 / 2021, a <https://neo4j.com/>

[13] Swagger. (2021). Consultat el 15 / 06 / 2021, a <http://www.guidemeup.com/swagger/ui/index.html>

## 10 AGRAÏMENTS

Agraïxo a la meva tutora Gemma Sánchez Albaladejo, l'ajuda me m'ha donat a orientar-me al projecte.

També vull agrair a l'empresa Guidemeup, per la confiança, per desenvolupar un portal tan important per ells. A demès, de la simpatia que ha mostrat alhora de reunir-nos per decidir l'avanç del projecte.

## 11 BIBLIOGRAFIA

[1] Guidemeup. (2013). Consultat el 2021 / 04 / 25, a LinkedIn: <https://www.linkedin.com/company/guidemeup/about/>

[2] Montserrat. (2020). Consultat el 25 / 04 / 2021, a Google Play: [https://play.google.com/store/apps/details?id=org.gmu.activities.stores.montserrat&hl=es\\_UY&gl=US](https://play.google.com/store/apps/details?id=org.gmu.activities.stores.montserrat&hl=es_UY&gl=US)

[3] Trello. (2021). Consultat el 24 / 04 / 2021, a <https://trello.com/>

[4] Team Gantt. (2020). Consultat el 14 / 05 / 2021, a <https://www.teamgantt.com/>

[5] Java. (2021). Consultat el 29 / 03 / 2021, a Wikipèdia: [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))

[6] Cassandra. (2021). Consultat el 15 / 03 / 2021, a Wikipèdia: [https://es.wikipedia.org/wiki/Apache\\_Cassandra](https://es.wikipedia.org/wiki/Apache_Cassandra)

[7] Docker. (2021). Consultat el 30 / 03 / 2021, a [https://hub.docker.com/\\_/cassandra](https://hub.docker.com/_/cassandra)

[8] Swagger. (2021). Consultat el 25 / 05 / 2021, a <https://swagger.io/>

[9] MongoDB. (2021). Consultat el 10 / 04 / 2021, a <https://www.mongodb.com/>

- 
- E-mail de contacte: borja.ibanezc@e-campus.uab.cat
  - Menció realitzada: Enginyeria del Software
  - Treball tutoritzat per: Gemma Sánchez Albaladejo (Departament de ciències de la computació i intel·ligència artificial)
  - Curs 2020/21