

# MIGRACIÓ D'UN SERVEI D'ENQUESTES EN EL NÚVOL DE SCALA A SPRING

Adrià Bergé Aguilar

## Resum

S'ha desenvolupat un servei web per tal de poder gestionar enquestes, rebre les respostes i després poder processar les dades rebudes i crear informació vàlida per les persones de màrqueting. El servei està desenvolupat amb el framework Spring i fent servir l'arquitectura REST. Internament s'ha aplicat arquitectures avançades com és un DAO o la injecció de dependències de Java EE. La implementació està orientada als futurs canvis que tindrà la plataforma, això ha condicionat la configuració del framework.

**Paraules clau**—Spring, Java, Enquestes, Servidor, Servei REST, DAO, IoD.

## Abstract—

A web service has developed to manage surveys, to receive answers and later to process the received information and to create information validates for the persons of marketing. The service is developed by the framework Spring and using the architecture REST. Internally have been applied advanced architectures as a DAO or the injection of dependences of Java EE. The implementation is oriented to future changes that will have the platform, this has conditioned the configuration framework.

**Index Terms**—Surveys, Java, Spring, Server, Service Rest, DAO, IoD.

## 1 INTRODUCCIÓ

L'activitat econòmica de l'empresa RateNow consisteix en recopilar informació mitjançant enquestes i, a posteriori, generar informes sobre la informació i la qualitat d'aquesta. El seu producte consisteix a fer enquestes curtes les quals es poden contestar en un minut de temps i es realitzen just després que el client hagi viscut l'experiència que valorarà a l'enquesta. Per tal de poder obtenir les respostes, instal·len en el lloc més adient els seus tablets que estan configurats amb l'enquesta. Els dispositius envien en temps real les respostes al servidor de RateNow on es processarà la resposta amb la informació ja emmagatzemada per tal que es pugui consultar el resultat de les enquestes tant per part de l'empresa com per part dels clients de RateNow. Hi ha un portal web que es connecta al servidor i genera els gràfics i la informació adient per tal de poder realitzar una interpretació clara dels resultats generats a partir de les respostes de les enquestes. Per poder gestionar

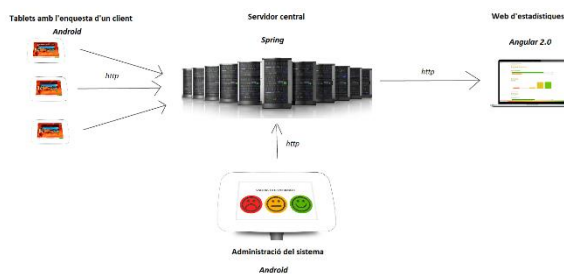


Fig. 1. Representació gràfica del sistema en totes les seves parts.

el sistema l'empresa té un conjunt de tablets amb un programa de gestió. En aquests tablets es pot portar a terme la definició d'una enquesta o introduir els nous usuaris.

Després de dos anys funcionant, volien arreglar defectes de la idea inicial del projecte i ampliar les funcionalitats que té el seu sistema d'enquestes. Per poder dur a terme aquest objectiu necessitaven trobar un tècnic amb amplis coneixements d'Scala, ja que la persona que va fer la primera versió del seu servei ara no està disponible. El perfil de tècnic especialista en Scala i serveis Rest no els ha sigut possible trobar-lo a Barcelona. Per aquest motiu, van decidir buscar una alternativa per poder seguir endavant. La

- E-mail de contacte: [adriaberge@gmail.com](mailto:adriaberge@gmail.com)
- Menció realitzada: Enginyeria del Software.
- Treball tutoritzat per: Ramón Baldrich Caselles (Departament de Ciències de la Computació i Intel·ligència Artificial)
- Curs 2015/16

decisió va consistir a refer tot el seu producte des de zero amb una tecnologia més madura i amb molts tècnics a la ciutat amb coneixements sobre aquesta.

D'aquesta manera, l'empresa no té una dependència de qui els desenvolupa el sistema, sempre podran trobar una persona que els segueixi mantenint i millorant el producte. Així poden corregir errors de concepte, que es vagin trobant en el transcurs del temps, i millorar la plataforma per ser més competitiu i flexibles, a nivell de prestacions.

## 2 ESTAT DE L'ART

En la actualitat Spring és una de les eines més utilitzades per crear portals webs i serveis web en Java, els motius del seu gran ús en l'àmbit són:

- **És fàcil de configurar:** tot es pot configurar mitjançant arxius xml, fins i tot les configuracions més avançades com pot ser la connexió a una base de dades o el login fent servi LDAP.
- **Integració amb altres eines de treball:** permet afegir altres eines (com: JUnit, Maven, Hivernate, Oauth 2.0, etc.) de forma senzilla i ràpida de configurar sense tenir conflictes entre les diferents eines.
- **Codi ordenat:** el codi quedarà ordenat gràcies a l'arquitectura MVC i a les anotacions que ens permet separar el codi en serveis o repositoris.
- **Ús d'anotacions:** ens facilita la configuració de mapejar una funció http amb una implementació Java i fa transparent al programador el mapeig dels paràmetres http en un objecte.
- **Fàcil creació d'objectes:** es requereixen molts objectes per crear la connexió a una base de dades, la configuració de la seguretat o la integració amb un sistema de generació de PDF's. Gràcies a la Injecció de Dependències, el sistema s'encarrega de generar tots els objectes i en l'ordre corresponent, fent que els programadors només hi hagin d'assignar usuaris, passwords i altres dades imprescindibles de configuració.
- **Estàndard de programació:** totes les aplicacions desenvolupades amb Spring tenen la mateixa estructura, de manera que si un programador canvia de projecte o d'empresa està en ple rendiment en poc temps.

Per tots aquests motius, i també perquè l'equip que desenvolupa aquesta eina està en constant millora i aplicant les noves tecnologies en el projecte, Spring és la millor eina per una empresa que vol tenir un producte escalable i mantenable en el temps.

Aquest framework de programació té una corba d'aprenentatge logarítmica. És molt fàcil conèixer els conceptes bàsics per tal de poder programar. Però quan vols augmentar la seguretat, afegir un framework molt innovador o desenvolupar una arquitectura molt estricta, es troben amb dificultats, ja que s'ha de sobrescriure components del sis-

tema. Aquest aprenentatge tan ràpid a l'inici dóna una facilitat molt gran a l'empresa per afegir nous desenvolupadors i formar-los en el framework. D'aquesta manera s'evitarà el problema inicial de no trobar professionals.

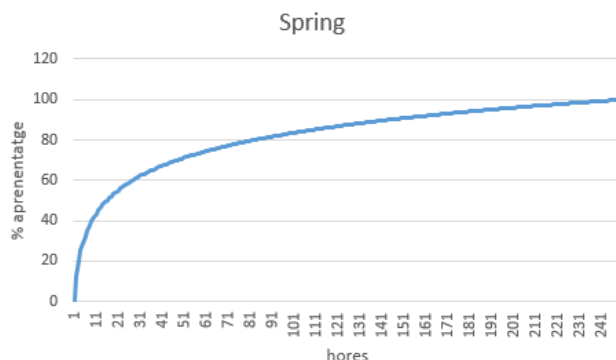


Fig. 2. Corba d'aprenentatge del framework Spring. Percentatge del coneixement de l'eina respecte hores invertides en aquest aprenentatge.

## 3 OBJECTIUS

L'objectiu del treball és iniciar la migració de la plataforma actual, implementada amb Scala i Akka, a una nova implementació en Java fent servir el framework Spring. S'utilitza com a documentació la informació que pot donar l'empresa i el codi del sistema que està en funcionament. El motiu d'utilitzar aquestes dos fonts d'informació és perquè no hi ha una documentació dels requeriments del sistema ni de com està confeccionada la base de dades MongoDB i tampoc del codi de comentaris del seu funcionament. Per aquests motius, s'ha de fer una interpretació del sistema actual.

S'ha d'implementar dos conceptes: el concepte d'enquesta i també el concepte d'usuari; ja que totes les tasques que es puguin realitzar seran sota unes credencials i mai es podrà fer de forma anònima.

Aquests objectius han estat modificats dues vegades en el procés de migració. En una primera instància es van voler abastar molts conceptes sense encara no conèixer prou bé la tecnologia Spring. Es va decidir eliminar del treball la recepció de respostes, el manteniment de dispositius i per últim les imatges de les preguntes. La segona vegada es va decidir deixar la documentació del codi per al final del projecte, ja que sinó no s'arribaria al final de les dues funcionalitats. Per altra banda, es van augmentar la quantitat de proves que es feien sobre una funcionalitat.

### 3.1 Gestió d'enquestes

Una enquesta està formada per un conjunt de preguntes i atributs que configuren el seu comportament. Els atributs d'una enquesta són:

- El temps màxim per contestar cada pregunta.
- El temps que passa entre el final d'una enquesta i el principi de la següent.
- Quin és l'idioma per defecte.
- Si les preguntes es mostren de forma desordenada.

- Nom de l'enquesta.
- Propietari de l'enquesta.
- Com es mostraran els objectes visuals per defecte en pantalla.

Totes les preguntes comparteixen una sèrie d'atributs:

- Identificador de la pregunta a la què s'ha de canviar si no es vol contestar la pregunta actual.
- Tipo de pregunta, fent referència a si les dades que recull són: emails, codis postals, NPS(Net Promoter Score), etc.
- La posició de la llista que ocupa la pregunta.
- Títol.
- Subtítol.

Les preguntes tenen un atribut diferent dependent de la tipologia de pregunta.

### 3.1.1 Pregunta Numèriques

Preguntes on un usuari pot fer una valoració dins d'un rang de números.

Els atributs de les preguntes numèriques són:

- Icona que representa un número.
- Rang mínim.
- Rang màxim.
- Llistat d'etiquetes per explicar el valor del número.

### 3.1.2 Pregunta Text Obert

Preguntes on l'usuari pot escriure text. Dependent del tipus de dades que sol·licita el contingut introduït pot ser alfanumèric o només numèric.

- Màxim de línies que es poden escriure.

### 3.1.3 Pregunta Seleccio d'Opcions

L'usuari ha de seleccionar opcions prefixades en el moment de dissenyar la pregunta.

- Si és múltiple selecció o única selecció.
- Si la última opció és de text lliure.
- En quantes columnes es distribuïran els elements seleccionables.
- Llista d'elements a poder seleccionar.

### 3.1.4 Pregunta Explicatives

Aquest tipus de preguntes no tenen atributs extrems, ja que no és una pregunta a consultar. La seva utilitat és de poder donar un missatge a l'usuari.

Per poder gestionar el concepte d'enquesta s'ha fragmentat en tres parts:

- **Funcions http necessàries:** definició de totes les URL's i els paràmetres que accepten i gestió dels possibles codis d'error que admet el protocol.
- **Elements que el defineixen:** classes que representaran els diferents conceptes que formen l'enquesta. Poden generar així un objecte complex.
- **Definició en SQL dels elements:** representació en SQL dels objectes definits anteriorment, de manera que podem fer persistent un objecte complex de Java.

## 3.2 Gestió d'usuaris

Per poder accedir a qualsevol funcionalitat del servei, que s'ha d'implementar, serà necessari estar validat com usuari del sistema. Un usuari del sistema té tres atributs que el diferencien:

- Nom d'usuari.
- La contrasenya.
- El seu rol del sistema, si és un usuari simple o un administrador.

L'usuari es gestionarà de la mateixa manera que en l'apartat anterior, ja que el procés és el mateix: Http, Model Java i representació SQL.

Dependent del rol que té un usuari podrà fer unes accions o unes altres.

### 3.2.1 Usuari simple

Un usuari amb aquest rol en l'estat actual només té permisos per poder consultar les enquestes de les quals ell és propietari. La seva forma d'accés és mitjançant la web de resultats.

### 3.2.2 Usuari administrador

Un usuari amb el rol d'administrador podrà fer totes les accions que permet el sistema. Encara que en el cas d'aquest rol no està permès tenir associat a ell enquestes. Només pot administrar el sistema, no pot utilitzar-ho per fer les seves enquestes.

## 3.3 Arquitectura robusta

L'arquitectura, els patrons de disseny i altres elements estructurals del servei Rest que s'ha d'implementar han de ser prou oberts i prou flexibles perquè sigui fàcil implementar noves funcionalitats i nous requeriments.

Es vol dissenyar un sistema modular que sigui capaç de poder afegir noves funcionalitats sense que això suposi una implementació nova del sistema, simplement una implementació de la funcionalitat afegida al sistema actual; cal tenir una idea de com fer el sistema flexible i, per altra banda, de com es poden fixar alguns aspectes.

Per tal de poder fer aquest objectiu es té accés a la llista de futurs canvis, millores i noves funcionalitats. També s'afegeixen nous requeriments i peticions especials a partir de cada reunió que es fa amb un client. D'aquesta manera es tindrà una visió més global que ens permetrà pensar en tot el que vindrà després i poder reaccionar sense haver de refactoritzar gran part del codi.

No són uns objectius molt ambiciosos, ja que s'ha de tenir en compte que en aquest procés s'ha d'aprendre el framework Spring, el mòdul de seguretat i també un dels quatre mòduls de persistència de dades: MongoDB, JDBC, JPA o Hibernate. Tenint aquest volum de requisits per aprendre abans de realitzar el treball, es pot després confirmar si als 30 primers dies s'aprèn la major part de l'eina de treball i es pot assolir tota la resta d'objectius abans descrits.

## 4 METODOLOGIA

La metodologia de treball que s'ha utilitzat ha sigut mitjançant objectius setmanals, ja que s'ha distribuït totes les

tasques en les diferents setmanes que dura el treball de final de grau. La metodologia ha estat inspirada en Scrum, sense ser d'estricta compliment el mètode Scrum.

Els dilluns a primera hora es fa una reunió amb el responsable de l'empresa i amb l'encarregat del projecte per tal de poder avaluar la feina feta la setmana anterior. En la reunió es rep un feedback de la feina feta i es defineixen les tasques a realitzar durant la setmana i si és necessari modificar la planificació per poder millorar la tasca anterior.

Durant la setmana s'implementaran les tasques a realitzar. Al matí es fa una reunió de 10 minuts amb el responsable del projecte en la qual es definirà l'objectiu del dia i es resoldran els dubtes que puguin sorgir.

Al final de la setmana s'entrega un fitxer en format war, el codi, la bateria de proves realitzades i la documentació del codi en format html, que serà el material a analitzar en la reunió del dilluns de la següent setmana.

El codi està sota el control de Git utilitzant Bitbucket com a servidor remot per tenir el control de les diferents versions. Es generaran 3 branques, la primera serà la master on es tindrà la versió entregada al client, la segona serà la desenvolup on hi ha els canvis en desenvolupament i que estan en creixement, però que encara no compleixen els requisits d'entrega, i per últim la tercera serà bugs on es repararan els errors trobats sobre la versió de master.

D'aquesta manera la reparació d'un error no afectarà el desenvolupament en curs i es podrà tenir la versió entregada actualitzada.

El test que es desenvoluparà serà un test de caixa negra fent servir JUnit i fent servir les tècniques de Test Drive Development per desenvolupar les funcionalitats mitjançant el test. També es realitzarà el test propi de Spring per provar si la configuració és la correcta.

Els responsables faran el test de les funcionalitats com a usuaris. No programaran un nou test, sinó que es connectaran les tabletetes d'administració al servei que s'està desenvolupant, fent proves de la funcionalitat com usuaris, esperant que les funcionalitats responguin bé.

La documentació setmanal serà mitjançant JavaDoc, tenint totes les classes i funcionalitats documentades. En finalitzar el projecte, la web que documenta tot el desenvolupament es generarà amb eines automàtiques.

Al final del projecte es generarà un manual d'usuari del servei Rest explicant cada un dels serveis, les seves dades d'entrades, les de sortida i les excepcions que pot generar. Aquest material serà acompanyat d'exemples en Java per connectar amb el servei i consumir cada una de les seves funcionalitats.

## 5 RESULTATS

En l'apartat de resultats s'explicaran les funcionalitats desenvolupades i també quina arquitectura s'ha aplicat en cada cas, si hi ha alguna peculiaritat o cas especial s'explicarà detalladament.

### 5.1 Mòdul d'usuaris

Totes les funcionalitats sobre les tasques d'usuari requereixen

estar autenticat en el sistema amb el rol d'administrador, ja que no està permès crear usuaris des d'un formulari anònim. Les tasques que s'ha implementat són:

#### 5.1.1 Consulta d'usuaris

Llistar tots els usuaris del sistema amb tota la seva informació, excepte les contrasenyes que estan xifrades i mai es podran consultar.

#### 5.1.2 Modificació d'un usuari

Es poden modificar totes les dades de l'usuari, excepte el password.

#### 5.1.3 Creació d'un usuari

S'afegiran un usuari nou al sistema. En el moment de la creació es genera la contrasenya i s'envia un correu electrònic a l'usuari amb les credencials d'accés a la plataforma.

#### 5.1.4 Regeneració de la contrasenya

Al regenerar la contrasenya es torna a reenviar les credencials mitjançant un e-mail a l'usuari.

UserController	
userService	UserService
get(int)	ResponseEntity<HttpUserRateNow>
add(HttpUserRateNow)	HttpUserRateNow
delete(int)	ResponseEntity<HttpUserRateNow>
generatePassword(int)	HttpUserRateNow
all	List<HttpUserRateNow>

Fig. 3. Classe on estan definides les funcions Http que fan referència a la gestió d'usuaris

L'arquitectura del servei està estructurada en tres capes que deleguen les responsabilitats.

La primera capa és la de funcions Http, on mitjançant l'anotació "@RestController" s'indica que una classe és un repositori de funcions web. Per definir cada una de les funcions s'utilitza l'anotació "@RequestMapping(value = "users", method = RequestMethod.GET)" on s'indica amb el valor la url que defineix la funció i el mètode del protocol que li correspon. Per poder obtenir la informació que es troba en el cos de la petició Http s'utilitza l'anotació "@RequestBody" la qual es assignada a una variable del mètode. Spring s'encarrega de transformar un JSON en l'objecte que correspon. La capa de funcions Http defineix la funció amb anotacions, com s'ha explicat anteriorment, gestiona la part d'aplicar restriccions d'usuaris, com pot ser la validació d'un usuari que només consulta les seves dades i per finalitzar crea la resposta Http depenent de quin ha sigut el resultat generat. Per generar un resultat el sistema delega la funcionalitat a la capa de Servei.

En aquesta capa s'aplica tota la lògica de negoci. Es transforma un objecte rebut per la petició web en un conjunt d'objectes els quals són representacions d'un registre de la base de dades. Per fer la persistència a la base de dades, la capa de servei delega la funcionalitat al Dao corresponent.

S'ha implementat una classe Dao per cada una de les taules de la base de dades on hi ha tots els mètodes necessaris per poder actuar amb la base de dades. Per tal d'estandarditzar la implementació dels Dao's s'ha generat dues interfícies; una pels mètodes comuns d'inserció, modificació i eliminació, i una altra amb els dos mètodes comuns per fer consultes a la base de dades. D'aquesta manera si un Dao representa una taula que relaciona dues taules que contenen dades no interessa fer consultes, ja que només es reben el valor de les id's dels dos objectes relacionats, per això s'implementarà només el Dao d'escriptura. Els Dao's que sí siguin necessaris per a la lectura tindran els dos mètodes bàsics i, si és imprescindible, nous mètodes per tal de poder fer Join's o altres tècniques que sigui possible.

Com es pot veure en la descripció de les tres capes d'implementació que hi ha, es pot apreciar que s'ha aplicat el patró de disseny MVC, on el model és la nostra capa de Dao's, la vista és el JSON que s'envia o es rep a la petició Http i el controlador està dividit entre la capa de servei i la definició de les funcions Http.

Per finalitzar amb la gestió d'usuaris manca per definir la funcionalitat d'autenticació. Com que un servei Rest no utilitza cookies ni tampoc sessions d'usuari a cada petició, s'ha d'introduir les credencials d'usuari. Per realitzar la validació de les credencials abans d'accedir a la funcionalitat requerida, s'ha hagut de posar un filtre a totes les peticions. Aquest filtre està delegat al framework d'Spring-Security que mitjançant un arxiu de configuració s'aplicaran els criteris necessaris. Per tal que el sistema de seguretat d'Spring coneixi els usuaris de la base de dades s'ha de generar un servei on s'implementarà una interfície pròpia del sistema per tal de poder donar-li informació de la base de dades corresponent. Un cop el sistema ha reconegut si és correcte, delegarà la funcionalitat a la funció web que correspon si no saltarà una excepció de seguretat i es contestarà la petició amb el codi 401 del protocol Http.

La configuració de la seguretat en el futur serà modificada per tal d'aplicar el protocol OAuth2.0, encara que per la part d'usuari s'aplicarà el mateix sistema d'autenticació.

## 5.2 Mòdul enquestes

Les funcionalitats sobre les tasques d'enquesta també requereixen l'autenticació de l'usuari amb el rol d'administrador. Les tasques implementades són:

### 5.2.1 Consulta d'una enquesta

Retorna l'enquesta sol·licitada. Per sol·licitar l'enquesta s'indica l'identificador a la url de la petició.

### 5.2.2 Consulta de totes les enquestes de les quals ets propietari amb filtres

Retorna un llistat d'enquestes les quals compleixen uns criteris, aquests són enviats en el cos de la petició en JSON.

### 5.2.3 Crear una enquesta

S'afegeix una enquesta nova al sistema. L'enquesta s'envia en el format JSON al cos de la petició Http.

### 5.2.4 Modificar una enquesta

Per modificar l'enquesta l'identificador de l'enquesta ha d'estar indicat a la url i es comprovarà que l'identificador de l'enquesta sigui el mateix que el de la url.

### 5.2.5 Eliminar una enquesta

No s'eliminarà mai una enquesta, simplement es marca com eliminada a la base de dades, però no es perden mai les dades.

QuestionnaireController	
questionnaireService	QuestionnaireService
userService	UserService
getQuestionnaires(boolean)	ResponseEntity<List<HttpQuestionnaire>>
get(String)	ResponseEntity<HttpQuestionnaire>
add(HttpQuestionnaire)	ResponseEntity<HttpQuestionnaire>
delete(String)	ResponseEntity<HttpQuestionnaire>
update(String, HttpQuestionnaire)	ResponseEntity<HttpQuestionnaire>

Fig. 4. Classe on estan definides les funcions Http que fan referència a la gestió d'enquestes

Com que una enquesta té un nivell de complexitat molt més gran que un usuari, s'ha hagut de modificar considerablement la capa de servei d'aquest element. La modificació consisteix a generar tres elements nous on es delega la funcionalitat de:

- Consulta d'una enquesta.
- Modificació d'una enquesta
- Creació d'una enquesta.

El motiu d'aquest aspecte és que un objecte de tipus enquesta està distribuït a la base de dades en un conjunt elevat de taules on cada una representa un element o un conjunt, des de les dades bàsiques de l'enquesta a llenguatges de l'enquesta, llistat de preguntes i les preguntes tenen les seves llistes de títols, opcions i altres elements. Com s'ha explicat prèviament, cada un dels elements de la base de dades té un objecte Dao que representa una taula i les seves accions a realitzar. En el procés de creació de l'enquesta intervenen setze taules diferents: LanguageDao, LanguagesDao, AtributsElementDao, TextDao textDao, TermsDao termsDao, TitleDao titleDao, SubTitleDao, HintDao, ChoiceDao, ChoiceTitleDao, NumericQuestionDao, ExploratoriQuestionDao, TextQuestionDao, SelectQuestionDao, QuestionnaireDao, RangeDao.

En el procés de consulta com que hi ha moltes consultes amb Join's es redueix a deu Dao's a consultar. Com es pot apreciar, un objecte tan complex requereix processos llargs d'implementar per les accions més bàsiques i comuns, ja que en realitat s'estan realitzant accions bàsiques de consulta, eliminació, modificació i creació per a cada un dels elements simples que el formen. Al fer la separació d'aquests processos i no unir-los al servei on hi ha la lògica de negoci, s'ha mantingut la simplicitat del codi de negoci i la independència de les classes. D'aquesta manera per fer modificacions futures l'arquitectura no està compromesa. Un punt important a tenir en compte és que un element de dades no és mai eliminat de la base de dades. Per eliminar

un element de dades el que es porta a terme és marcar que ha sigut eliminat i s'introdueix tant la data com l'identificador de l'usuari que l'eliminat. Tot i així, tots els elements de relació d'elements de dades sí que s'eliminen. Aquest comportament està per poder fer un control de qui ha eliminat una base de dades o una pregunta o qualsevol element, ja que en el passat s'ha trobat ocasions estranyes.

El concepte enquesta és el més important de la plataforma, ja que és el centre i rebrà moltes modificacions futures, per aquest motiu s'ha de prestar atenció en el moment de definir les funcionalitats i com es divideix el codi per tal que una modificació en el concepte d'enquesta sigui trivial modificar i no s'hi hagi de replantejar el procés de dalt a baix.

## 8 CONCLUSIONS

S'ha assolit els objectius fixats en la planificació del projecte i es pot confirmar que s'ha generat una arquitectura prou independent entre ella per ser flexible a modificacions i canvis futurs.

Es pot determinar també que en el transcurs del projecte és possible aprendre l'eina Spring el suficient per realitzar el gruix del projecte. Hi ha uns apartats que no formen part del treball i que no formen part del coneixement obtingut, però com s'ha explicat les especialitzacions de la plataforma és la part més complexa i difícil d'aprendre.

La selecció de l'eina per part de l'empresa es considera que és una bona opció, ja que ha sigut demostrada la facilitat d'aprenentatge, d'escalabilitat i manteniment del codi. Són els punts més importants per una empresa que està creixent i millorant el seu producte per ser totalment competitiu en el mercat.

El projecte encara no es pot determinar com un producte finalitzat per l'empresa perquè falten moltes funcionalitats a desenvolupar per poder executar en un servidor de producció i substituir l'actual, però és el principi per la futura actualització de la seva aplicació d'enquestes. Ja que s'ha elaborat un sistema prou obert i independent entre els seus elements per a poder créixer sense complexitats en el desenvolupament.

## 9 LÍNIES FUTURES

Els pròxims passos a desenvolupar seran:

- Obtenció de respostes.
- Gestió de dispositius.
- Implementació del protocol OAuth2.0.
- Permisos avançats d'usuari.
- Calendari d'activació automàtica i desactivació d'una enquesta en un dispositiu.
- Obtenció de respostes via pàgina web.
- Filtre de respostes per Spam.
- Poder pujar imatges per personalitzar enquestes.
- Permetre a un usuari poder dissenyar la seva pròpia enquesta.

Les implementacions futures també vindran determinades

per les demandes dels clients i de com evoluciona el mercat. En dos anys ha canviat molt des que es va confeccionar la primera plataforma i el camí que queda no serà diferent.

La nova plataforma encara li queda un llarg camí per a estar llesta i finalitzada, encara que un cop s'assoleixin les funcionalitats bàsiques ja es podrà posar en producció tenint algunes restriccions de funcionament. D'aquesta manera cada mes es podrà donar noves funcionalitats i així també els clients veuran una constant evolució del producte i un valor afegit.

## AGRAÏMENTS

En primer lloc, voldria agrair a l'empresa Lean Lemon SL per brindar-me l'oportunitat de realitzar les pràctiques, el treball final de grau i l'aprenentatge d'Spring amb ells i el suport del seu personal.

En segon lloc, al tutor del treball Ramon Baldrich i Caselles pel seguiment i la paciència que ha hagut de demostrar amb mi. Les crítiques dures i consells en les parts on més dificultats he tingut m'ha ajudat molt a aconseguir el resultat.

Per últim, a la meua parella que gràcies a ella s'ha pogut fer més tolerables aquests anys de carrera universitària i sense la qual no hagués aconseguit continuar endavant.

## BIBLIOGRAFIA

- [1] Oracle, «DAO Oracle,» [En línia]. Available: <http://www.oracle.com/technetwork/java/dataaccessobject-138824.html>.
- [2] <https://www.scrum.org/>, «Scrum,» SCRUM, [En línia]. Available: <https://www.scrum.org/>. [Últim accés: 3 3 2016].
- [3] Oracle, «Java Doc,» 15 02 2016. [En línia]. Available: <https://docs.oracle.com/javase/7/docs/api/>.
- [4] C. Walls, Spring in Action. Third Edition, Tercera edició ed., Madrid, Madrid: Ediciones Anaya Multimedia (Grupo Anaya, S.A.), 2012, p. 479.
- [5] MongoDB Inc, «MongoDB,» 22 2 2016. [En línia]. Available: <https://www.mongodb.org/>.
- [6] R. C. Martín, Código Limpio, Madrid: Ediciones Anaya Multimedia, 2009.
- [7] L. Debrauwer, Patrones de diseño en Java, Barcelona: Ediciones ENI, 2013.
- [8] Pivotal Software, Inc, «Spring,» 15 02 2016. [En línia]. Available: <http://spring.io/>.
- [9] F. J. C. Sierra, Java 2 Curso de programación, Madrid: RA-MA Editorial, 2005.
- [10] «hascode,» [En línia]. Available: <http://www.hascode.com/2015/07/integrating-swagger-into-a-spring-boot-restful-webservice-with-springfox/>. [Últim accés: 19 4 2016].
- [11] stackoverflow, «stackoverflow,» [En línia]. Available: <http://stackoverflow.com/>.
- [12] <https://maven.apache.org/> [Últim accés: 24 6 2016]
- [13] <http://www.oracle.com/technetwork/java/javase/tech/persistence-jsp-140049.html> [Últim accés: 24 6 2016]
- [14] <http://www.oracle.com/technetwork/java/javase/jdbc/index.html> [Últim accés: 24 6 2016]
- [15] <http://hibernate.org/orm/documentation/5.2/> [Últim accés: 24 6 2016]
- [16] <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/beans.html> [Últim accés: 24 6 2016]

- [17] <http://docs.spring.io/spring/docs/current/spring-framework-reference/html/mvc.html> [Últim accés: 24 6 2016]
- [18] <https://spring.io/guides/gs/rest-service/> [Últim accés: 24 6 2016]
- [19] <https://spring.io/tools/sts/all/> [Últim accés: 24 6 2016]
- [20] <https://www.mysql.com/> [Últim accés: 24 6 2016]

## APÈNDIX

### A.1 Diagrames parcials base de dades MySql

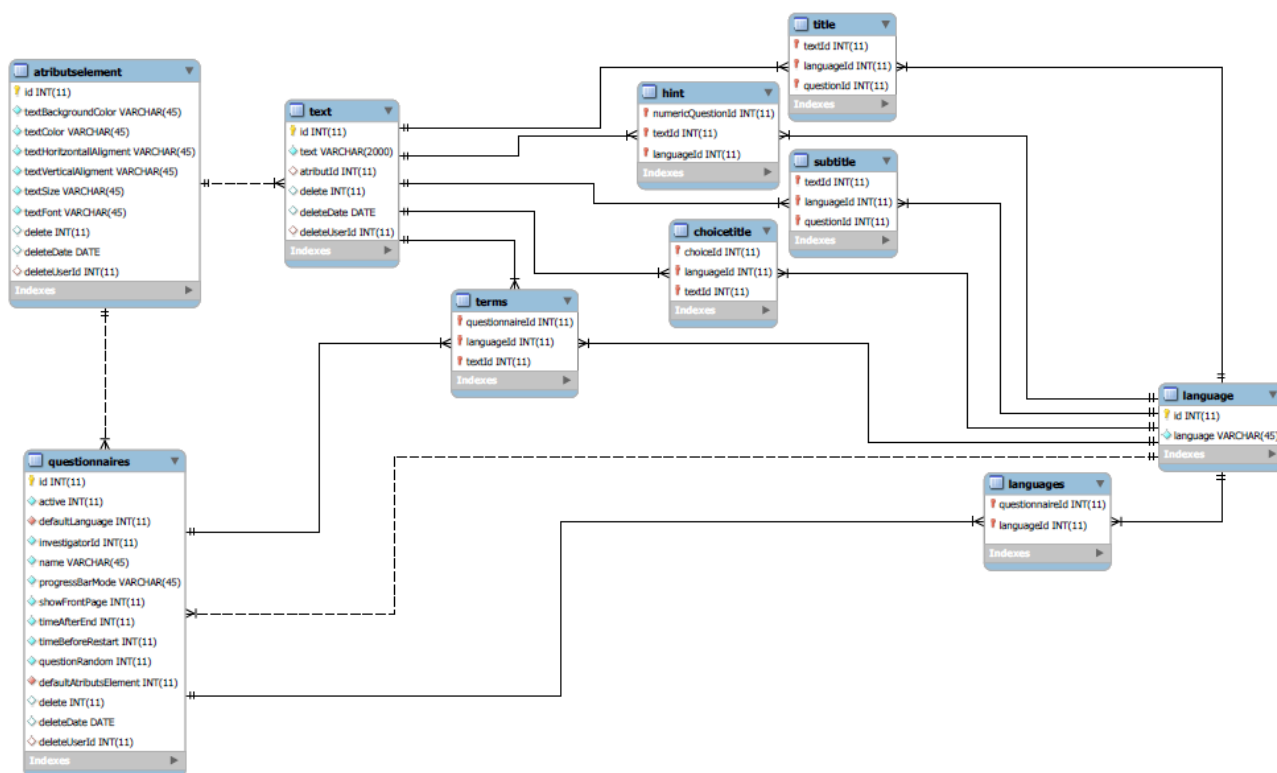


Fig. 4. Implementació del multilinguatge en les enquestes, tots els texts de les enquestes estan a la taula `text`, a la taula `language` es troben els llenguatges del sistema i per finalitzar hi ha un conjunt de taules que determinen l'element que representa el text, com pot ser un títol, un subtítol o els termes i condicions de l'enquesta, tots ells indiquen la llengua en què estan escrits. Es pot observar la relació que hi ha amb la configuració visual dels elements a la taula `atributselement`. I finalment com hi ha una relació de llengües amb l'enquesta, d'aquesta manera hi ha una llista ràpida de totes les llengües en què està traduïda l'enquesta, així cada usuari pot seleccionar en quin idioma contestar l'enquesta.

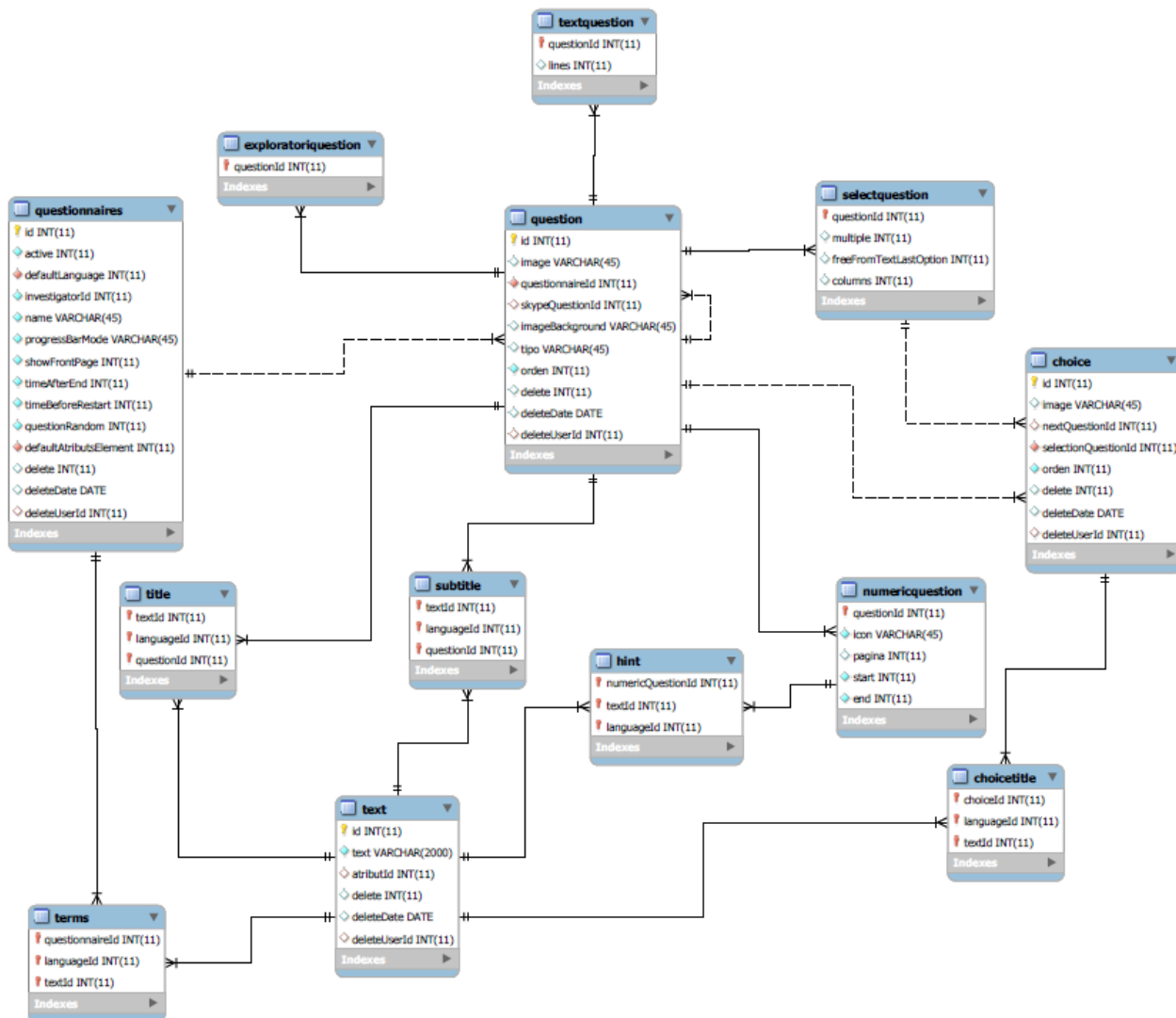


Fig. 5. Representació de tots els elements que formen una enquesta, des de la descripció de l'enquesta a tots els tipus de preguntes diferents que hi ha. Tots ells relacionats pels identificadors únics. Es pot observar com les preguntes estan dividides en dues taules, una comuna entre totes, on es troben les dades que són comunes i són més de configuració del sistema, després cada tipus de pregunta té la seva taula amb les dades específiques. El resultat per representar un objecte de tipus pregunta és unint els camps de les dues taules. El diagrama no és definitiu, està subjecte a futurs canvis de la plataforma.

