Desenvolupament de la segona versió d'una aplicació web per a la gestió d'exercicis de bases de dades del sistema LearnSQL

Autor: Mingjian Chen
Directora: Carme Quer
Titulació: Grau en Enginyeria Informàtica
Especialitat: Enginyeria Del Software
Universitat: Universitat Politècnica de Catalunya
Data: 23/06/2017

Resum

LearnSQL (*Sistema Learning Environment for Automatic Rating Notions of SQL*) és un sistema que s'usa en les assignatures de bases de dades de la FIB per donar suport l'aprenentatge als estudiants, el sistema permet per una banda la correcció automàtica dels exercicis que es fan a les assignatures de bases de dades, oferir *feedback* immediat de la seva solució. Per altra banda permet mantenir un repositori dels exercicis que es poden consultar i re-aprofitar al llarg dels anys.

El sistema està format per diversos subsistemes. Un d'ells, és l'anomenada *Authoring Tool* (AT), és una eina que permet als professors gestionar els exercicis del repositori. Aquesta aplicació la va desenvolupar Marc Fernàndez l'any 2010 [1], posteriorment degut a que l'*Authoring Tool* tenia limitacions de disponibilitat i distribució, per tant l'Albert Boada va desenvolupar un projecte partint de l'AT, i la va portar cap a la plataforma web, a partir de llavors la va anomenar *Authoring Tool Web* (ATW) [2]. S'ha desenvolupat una aplicació amb un disseny i implementació que pot ser fàcilment extensible i modificable.

Per tant, en aquest projecte es desenvolupa una segona versió que parteix d'una llista de requisits proporcionats pels professors, i aprofitar al màxim la bona feina realitzada de la primera versió, la segona versió de l'ATW segueix utilitzant el mateix disseny, els mateixos patrons, i les mateixes tecnologies com PostgresSQL, Apache Struts2, AngularJS, JSON etc...

Abstract

LearnSQL (System for Automatic Learning Environment Rating Notions of SQL) is a system used in the database subjects in FIB to support student learning, on the one hand the system allows the automatic correction exercises of the subjects database, providing immediate feedback of their solution. On the other hand it provides a repository of exercises that can be consulted and re-take over the years.

The system consists of several subsystems. One of them is called Authoring Tool (AT) which is a tool that allows teachers to manage exercises of the repository. This application was developed by Marc Fernàndez in 2010 [1], and later because the Authoring Tool had issue in terms of availability and distribution, so Albert Boada developed a project based on the AT, and migrated it to the web platform , thereafter it's called the Web Authoring Tool (ATW) [2]. He made an application design and implementation that can be easily extensible and modifiable.

Therefore, this project is developing a second version based on a list of requirements provided by teachers, and take the most of the good work done in the first version, the second version of the ATW is still using the same design, the same patterns and the same technologies such as PostgresSQL Apache Struts2, AngularJS, JSON etc ...

Índex

Índex de taules

Índex de il·lustracions

1. Context

1.1. Introducció i formulació del problema

Aquest projecte és un Treball Final de Grau (TFG) de l'especialitat d'Enginyeria del Software de la Facultat d'Informàtica de Barcelona (FIB) de la Universitat Politècnica de Catalunya (UPC). El projecte és de modalitat A ja que es tracta d'un TFG vinculat al departament d'Enginyeria dels Serveis i els Sistemes d'Informació (ESSI) de la UPC proposat per la Carme Quer, directora del TFG.

En aquest projecte es desenvoluparà una segona versió d'una aplicació web per a la gestió d'exercicis de bases de dades del sistema *LearnSQL.* Parteix del projecte final de carrera anomenat *Desenvolupament d'una interfície web per a una aplicació per a la gestió de preguntes (LearnSQL Authoring Tool)* realitzat per l'estudiant Albert Boada [1].

Aquest projecte se situa en el marc del sistema *Learning Environment for Automatic Rating Notions of SQL (LearnSQL)* [3]. Aquest sistema s'usa en les assignatures de bases de dades de la FIB per donar suport a l'aprenentatge d'aquesta matèria. El sistema LearnSQL actualment està tenint un efecte positiu en les assignatures on s'utilitza. Tant els professors com els estudiants surten beneficiats del seu ús. Per una banda, facilita el treball dels professors. Per altra banda, els estudiants poden tenir *feekback* immediat de les solucions dels exercicis que se'ls hi plantegen gràcies al procés d'autocorrecció que ofereix el sistema.

El LearnSQL està format per diferents subsistemes. Un d'ells, l'anomenada *Authoring Tool* (AT), és una eina que permet als professors accedir al repositori d'exercicis (també anomenats qüestions) del sistema i gestionar-lo a nivell de parametrització tant de la pròpia qüestió com del procés d'auto-correcció. Actualment l'AT que utilitzen els professors en producció és una aplicació *JAVA*, que només permet accedir al repositori de qüestions des d'aquells dispositius on hagi estat prèviament instal·lada. A l'hora de distribuir noves versions de l'aplicació és difícil d'assegurar que tots els professors l'han descarregat i estan usant la nova versió. Amb l'ús continuat del sistema, i la necessitat de mantenir-lo, cal plantejar fer la migració cap a una plataforma que pugui satisfer la necessitat de disponibilitat i distribució que és la *plataforma web*.

Degut a que les múltiples funcionalitats de l'AT, en el projecte de l'Albert Boada no es va poder finalitzar el disseny i implementació de tots elles. No obstant, es va desenvolupar una aplicació web estable, fàcil d'utilitzar i amb una interfície agradable a l'usuari. Amb la finalitat de dissenyar i implementar les funcionalitats que faltaven, i d'afegir noves funcionalitats que ajudaran als professors en la seva feina, va sorgir aquest projecte.

Les funcionalitats que es van desenvolupar en la primera versió de l'aplicació web es mantindran, seguint els patrons arquitectònics i utilitzant les tecnologies que són base de l'aplicació que s'agafa com a punt de partida. Aquestes tecnologies són: *AngularJS* [4], *Apache Struts 2* [5], *JSON.*

1.2. Stakeholders (parts interessades)

El projecte a desenvolupar va adreçat a un tipus d'usuari específic, concretament els professors de les assignatures de bases de dades de la FIB. De totes maneres, hi ha altres persones interessades en el projecte.

a) Desenvolupador del projecte

El desenvolupador serà la persona que dissenyarà, desenvoluparà i testejarà les funcionalitats el sistema software, i serà l'encarregat de comprovar que l'aplicació funciona correctament un cop acabada. Es considera que l'autor d'aquest projecte, és a dir jo mateix, serà el que ha de tenir aquest rol.

b) Directora del projecte

La directora del projecte és la professora Carme Quer, del departament d'ESSI [6], professora de l'assignatura Bases de Dades de la FIB. La directora s'encarregarà de proposar els requisits de l'aplicació , i de guiar i supervisar la feina a realitzar en el projecte.

c) Usuaris de l'aplicació

Com s'ha comentat, els usuaris seran els professors d'assignatures de bases de dades. Els professors ja són usuaris de l'aplicació JAVA actual, i tenen interès en tenir una aplicació accessible via web, usable i que els permeti més flexibilitat per a gestionar el repositori d'exercicis.

d) Altres desenvolupadors

El desenvolupament de la segona versió de l'aplicació AT pot també afectar a altres desenvolupadors que tinguin intenció de migrar l'aplicació a altres plataformes en un futur.

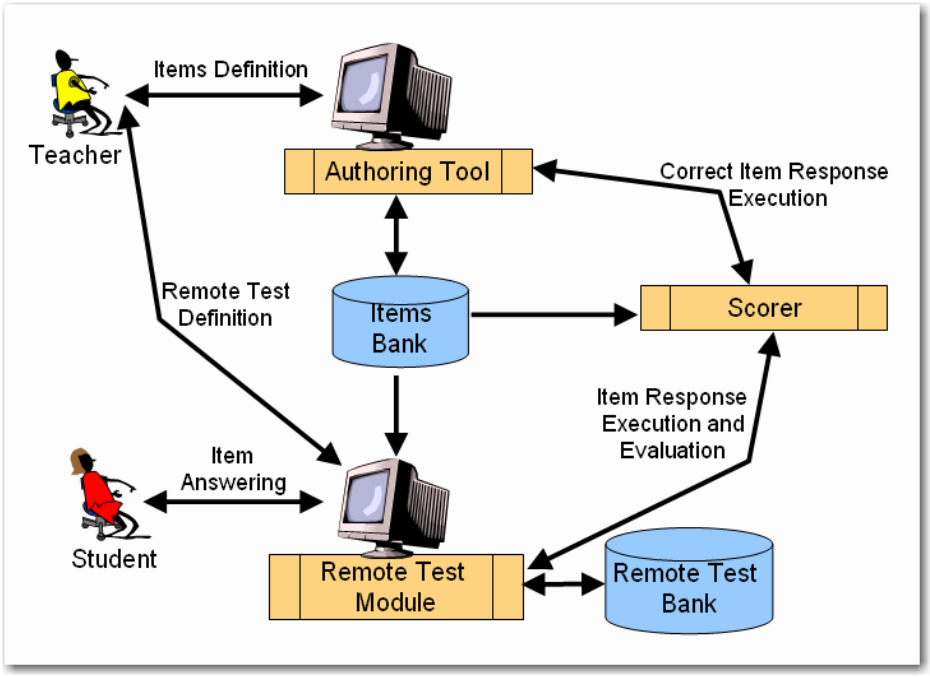
1.3. Estat de l'art

1.3.1. Sistema LearnSQL

El sistema LearnSQL és un sistema complex enfocat a l'ensenyament *Bases de Dades Relacionals* [7] en diferents assignatures de la UPC. Els seus objectius principals són:

a) Proveir exercicis sobre el temari de Bases de Dades Relacionals que els estudiants puguin resoldre tant des del laboratori de la FIB com des de casa.
b) Automatitzar l'avaluació i correcció de les solucions d'estudiants sense la intervenció de personal docent.
c) Ajudar als professors a proposar exercicis o qüestions, així com revisar les solucions d'estudiants.
d) Facilitar l'autoaprenentatge de la matèria als estudiants.
e) Proporcionar *feedback* immediat a l'estudiant.

A aquest diagrama *Figura 1* podem veure els subsistemes del LearnSQL i les relacions entre ells.



Il·lustració 1: L'arquitectura del sistema LearnSQL

El sistema LearnSQL està format per tres subsistemes:

- **L'Authoring Tool:**
  - *L'AT* permet als professors dissenyar i gestionar els exercicis, establir la solució correcta que considera el professor, i provar possibles solucions d'estudiants per tal de saber si la definició de l'exercici és correcta. Tant la solució del professor com altres possibles respostes dels exercicis s'envien a corregir des de l'AT a un o més serveis web correctors (Scorer).

- ***Remote Test Module: Plugin de Moodle* [8] *per a* LearnSQL.**
  - *Moodle* és un entorn virtual per a la creació i gestió de cursos enfocat en l'ensenyament i aprenenetage.
  - El Plugin per a LearnSQL permet crear qüestionaris remots dins d'un curs de Moodle. Aquests qüestionaris agrupen exercicis de bases de dades disponibles en el repositori d'exercicis, prèviament creats amb l'AT. També permet als estudiants contestar aquests qüestionaris des de casa o des del laboratori, enviant a corregir les seves solucions i rebent feedback sobre la correctesa d'aquestes solucions.

- **Scorer o servei web corrector**
  - *El Scorer* és un sistema que està formant per un conjunt dels serveis web correctors, la seva tasca fonamental és enviar a executar solucions dels diferents exercicis en una Sistema de Gestió de Bases de Dades (SGBD) [9].
  - El Scorer té un servei web per cada tipus d'exercici que el LearnSQL pot corregir.

1.3.2. Sistemes existents

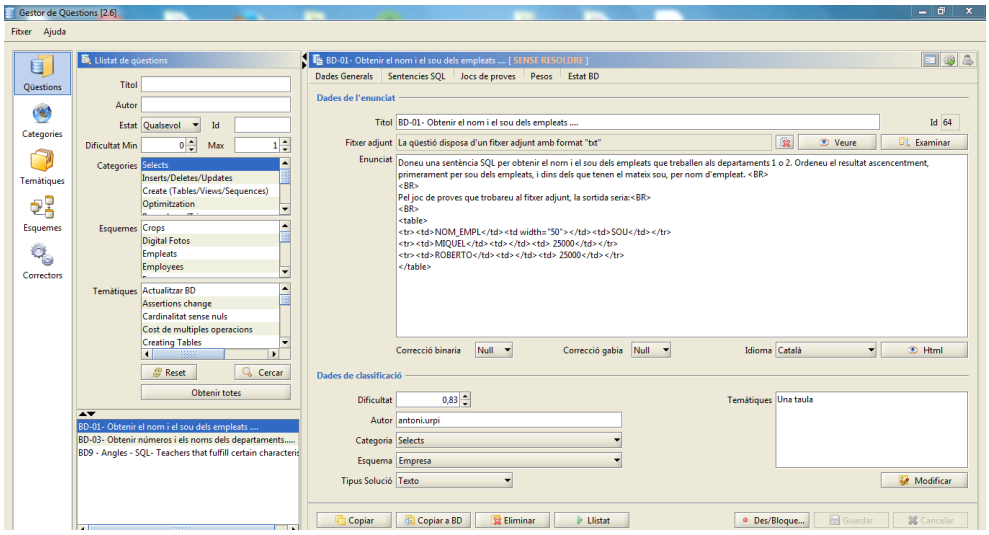
En aquest projecte no hi ha alternatives d'aplicacions software amb la mateixa funcionalitat que l'AT. Per altra banda, tal com es veu en l'estudi d'eines existents per a la aprenentatge de la matèria bases de dades realitzat pels professors de bases de dades de la FIB a [10], no hi ha cap altre sistema amb les mateixes funcionalitats que els autors hagin pogut trobar després de fer una cerca sistemàtica.

L'únic que existeix és l'aplicació d'escriptori *AT* del propi sistema LearnSQL, i la primera versió de la ATW.

Com s'ha comentat l'aplicació AT d'escriptori té dos problemes: només permet accedir al repositori de qüestions des d'aquells dispositius on hagi esta prèviament instal·lada i a l'hora de distribuir noves versions de l'aplicació és difícil d'assegurar que tots els professors l'han descarregat i estan usant la nova versió.

L'AT d'escriptori per altra banda, també li falta una funcionalitat que els professors han trobat a faltar des del seu desenvolupament, que és el permetre guardar models de solucions alternatives i/o incorrectes dels estudiants per a cada exercici, i poder corregir en bloc aquestes solucions incorrectes veient la nota que traurien si els estudiants fan aquestes solucions en un exercici d'examen.

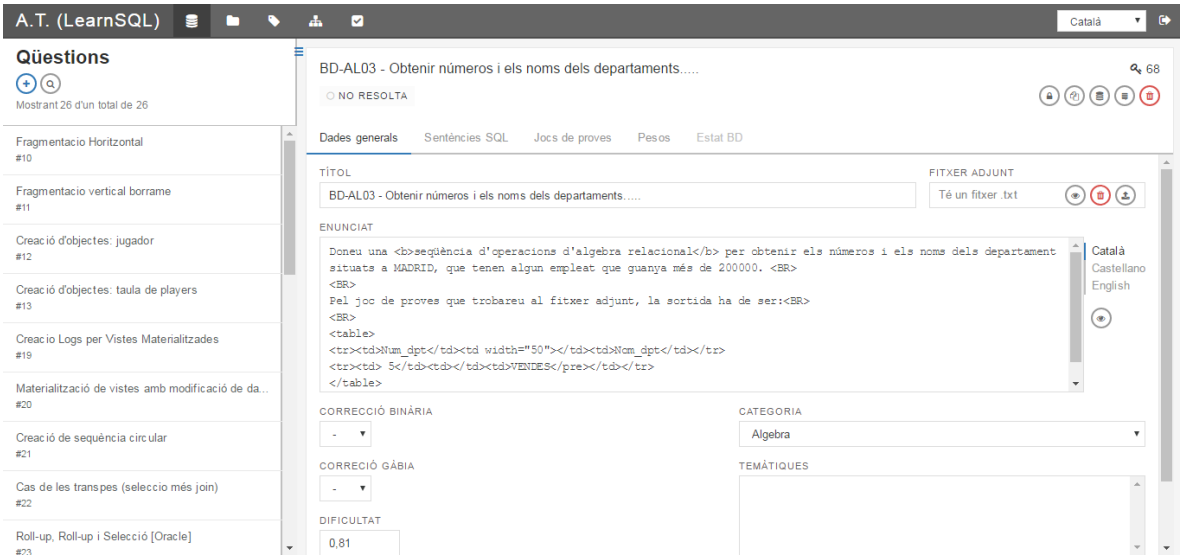
A la *Figura* 2 es pot veure el panell d'una qüestió de l'aplicació d'escriptori.



Il·lustració 2: Interfície de l'aplicació d'escriptori

Respecte a l'ATW desenvolupada posteriorment, té una molt bona experiència d'usuari, i d'usabilitat. Però el problema que té és que no està totalment desenvolupada, li falten funcionalitats existents a l'aplicació d'escriptori. Per altra banda, també li falta la nova funcionalitat que els professors demanen tenir que permeti guardar i executar en bloc solucions incorrectes dels exercicis.

A la *Figura* 3 podem veure la interfície de l'ATW.



Il·lustració 3: Interfície de l'aplicació web

1.3.3. Conclusions

No hi ha cap sistema alternatiu que iguali en funcionalitats al LearnSQL.

Les aplicacions de gestió d'exercicis del LearnSQL que hi ha disponibles actualment no cobreixen totes les necessitats dels professors, encara que l'ATW proporciona aspectes que l'AT no té, com són l'accés sempre a la darrera versió de l'aplicació sense necessitat d'instal·lar res.

Per tant, cal desenvolupar una nova versió de l'ATW que inclogui totes les funcionalitats existents a l'AT més la nova funcionalitat que requereixen els professors.

1.3.4. Objectius del projecte

Partint d'un sistema on els components són funcionals i estables, l'objectiu principal d'aquest projecte es desenvolupar la segona versió de l'aplicació ATW, que en aquest projecte anomenarem ATW2. A continuació es descomposa aquest objectiu en objectius més concrets del projecte:

- Fer un estudi previ per tal d'entendre les funcionalitats de l'AT i de la versió de partida de l'ATW. Conèixer les tecnologies que utilitzen i entendre el disseny arquitectònic de cadascuna de les aplicacions, el model conceptual i l'estructuració del codi que les implementa.

- Estudiar les tecnologies que es van usar per a desenvolupar la primera versió de l'ATW. Aquestes tecnologies i en especial el *AngularJS* usat en la part de *Frontend*, caldrà estudiar-lo amb tutorials i altra documentació degut al desconeixement previ d'aquesta tecnologia.

- Aprofitar el màxim possible de la lògica de negoci de l'AT d'escriptori a l'hora d'afegir noves funcionalitats a l'ATW.

- Configurar l'entorn de treball, ja que hi han diversos eines i programaris imprescindibles que s'utilitzen en aquest projecte.

- Desenvolupar les noves funcionalitats de l'ATW. Entre elles cal destacar com a prioritàries: desenvolupar la gestió dels correctors d'exercicis (anteriorment anomenats scorers), l'enviament de les solucions de professor a executar, la gestió i execució de possibles respostes d'estudiants, la consulta de determinades dades que poden ajudar als professors en la definició de les preguntes.

- Desenvolupar algunes funcionalitats de la gestió de qüestions que varen quedar pendents enla primera versió de l'ATW (generar llistats de les qüestions, duplicar qüestions, copiar qüestions entre repositoris de qüestions, etc) i assegurar que la resta de funcionalitats tenen un correcte funcionament.

- Millorar aspectes d'interfície, es tracta de desenvolupar els requisits que comportin millores en quan a la usabilitat.

2. Abast del projecte

2.1. Abast

Un cop sabem quin és el problema a resoldre, podem definir l'abast del projecte a partir els objectius escrits a l'anterior apartat.

- L'aplicació ATW2 adaptarà en la mesura que es pugui la lògica de negoci de l'AT d'escriptori, i seguirà el disseny arquitectònic i tecnologies fetes servir en l'ATW.

- L'aplicació ATW2 permetrà modificar, crear, i eliminar correctors, es podrà assignar i desassignar qüestions als correctors, i es podrà executar totes les qüestions d'un corrector.

- L'aplicació ATW2 afegirà una secció d'execució de la solució del professor. Aquesta secció permetrà executar la solució indicada pel professor per a la qüestió, assignar i desassignar els correctors existents, i modificar l'ordre d'execució dels jocs de proves d'aquesta qüestió.

- L'aplicació ATW2 afegirà la gestió de les possibles respostes dels estudiants per la qüestió. En aquesta secció es podran veure les solucions típiques dels estudiants amb la nota de cada solució segons la definició de la qüestió que hagi fet el professor. També permetrà afegir, eliminar i executar solucions típiques.

- L'aplicació ATW2 afegirà la possibilitat de poder definir un fitxer zip genèric per a les preguntes tipus *JDBC*, que posteriorment el professor podrà generar amb informació específica de cadascuna de les preguntes JDBC.

- L'aplicació ATW2 afegirà la possibilitat de seleccionar un fitxer local, i fer *upload* manual d'un fitxer necessari per a les preguntes de tipus *JDBC,* també es podrà fer una generació automàtica del fitxer zip per a les preguntes de tipus *JDBC.*

- L'aplicació ATW2 permetrà generar arxius auxiliars que necessiten els estudiants per resoldre la qüestió.

- L'aplicació ATW2 afegirà una secció de resultats que permetrà consultar informació que ajudi als professors en la definició de preguntes. Concretament es mostrarà per cadascuna de les possibles respostes de la qüestió, els joc de proves (i en cas que n'hi hagi les comprovacions) que aquestes respostes passen i els que no. També es podrà mostrar per cadascuna de les vegades que la pregunta s'ha inclòs en un curs de Moodle-LearnSQL, les notes mitjanes dels estudiants, i el número d'estudiants que l'han fet per cada grup d'estudiants definit en el curs de Moodle.

- L'aplicació ATW2 estarà limitada als usuaris que tinguin accés al repositori de qüestions.

2.2. Possibles obstacles

Els obstacles més destacats durant el desenvolupament del projecte poden ser:

- **Restricció temporal**

El TFG té una data límit especifica, la durada aproximada del projecte és de 6 mesos, per tant caldrà dedicar hores diàriament de forma constant al projecte per assolir els objectius establerts.

- **Comprensió de codi**

Una de les tasques més complicades és entendre el codi que ha implementat una altra persona, ja que el projecte es desenvoluparà partint d'un altre projecte, es dedicarà molt de temps llegint tota la documentació i analitzant l'estructura de les aplicacions AT i ATW, estudiar les relacions entre les classes, i com tot això es plasma al codi inicial per no perdre els fluxos de treball als que estan acostumats.

- **Configuració de l'entorn de treball**

Una altra dificultat que estarà present durant el període inicial del projecte és la configuració de l'entorn de treball. S'haurà d'instal·lar les eines que fa servir l'ATW, sobretot l'IDE Eclipse, i trobar totes les llibreries adequades per a que funcioni correctament l'aplicació.

- **Desconeixement de la tecnologia**

El fet de que l'ATW principalment utilitzava les tecnologies com el *framework* d'aplicacions web *JAVA Apache Struts 2* i el *framework* d'aplicació client *AngularJS,* fa que s'haurà d'anar aprenent aquestes tecnologies mentre es realitzarà el desenvolupament de l'ATW2, això també suposarà molt més temps a invertir.

2.3. Metodologia i rigor

2.3.1. Mètodes de treball

Per tal de desenvolupar el projecte de forma organitzada s'establirà una metodologia de treball que s'usarà al llarg temps del projecte. Amb aquest objectiu, s'ha decidit seguir la metodologia àgil on s'ha anat desenvolupant d'una manera iterativa i incremental per prevenir qualsevol canvis en terme de requisits i funcionalitats del projecte.

Donat els objectius del projecte, es farà una llista de tots els requisits de l'ATW2, s'ordenaran segons la seva prioritat, i un cop fet això, s'agruparan aquests requisits en diferents iteracions.

Cada iteració o *Sprint* consistirà en planificació, anàlisi de requisits, disseny, codificació i proves. Aquests fases d'iteració es fan d'una manera seqüencial i la idea d'aquesta metodologia àgil és obtenir una versió millor i amb més funcionalitats que l'anterior versió. Quan s'arribi a la fase de proves, s'han de testejar totes les funcionalitats existents més les noves que s'han afegit. Un cop assegurem que les funcionalitats noves han funcionat correctament, arribem al final de *Sprint*, i anem a l'entorn d'integració on es puja tot el codis modificat a un sistema de control de versions *Bitbucket* [11] per evitar la pèrdua de dades.

2.3.2. Eines de seguiment

Per tal de mantenir un control en les tasques es crearà un document Excel on es podrà mantenir un control visual de les tasques del projecte, les hores dedicades a cada una d'elles i la data de realització d'aquesta, a més aquest Excel es pot ajudar quan es realitza el diagrama de *Gantt.*

Un altre sistema molt important serà la comunicació constant amb la directora del projecte, aproximadament cada dues setmanes es mantindrà una reunió per supervisar les tasques realitzades, i indicar certs aspectes les possibles modificacions per assolir el millor treball possible. La comunicació serà mitjançant correus electrònics per temes de reunions i incidències que hi pot haver.

2.3.3. Mètodes de validació

- **Avaluació de final de tasca**

Al final de cada tasca d'implementació es farà les proves per tal establir si el resultat és satisfactori i decidir si es pot plantejar cap a nova tasca.

- **Avaluació de final de *Sprint***

Al final de cada *Sprint* es farà una avaluació de totes de tasques d'aquest *Sprint,* i caldrà assegurar que les funcionalitats afegit d'aquest *Sprint* funcionin correctament juntament amb les de les funcionalitats d'altres *Sprints* anteriors.

- **Supervisió de la directora**

Tant en les reunions periòdiques com l'avaluació final abans de posar el projecte en producció, la directora farà proves d'acceptació per assegurar que s'està desenvolupant el projecte correctament tal com s'ha previst.

3. Planificació

3.1. Calendari

El projecte està previst que tingui una durada estimada de sis mesos amb data inicial el dia 3 de gener de 2017, dia en que es va començar a configurar l'entorn de desenvolupament, i com a data final el dia de defensa del projecte que serà l'ultima setmana de mes juny. S'intentarà acabar el projecte unes quantes setmanes abans del dia de defensa per tal de preparar l'exposició final i per tenir un temps de marge de prevenir qualsevol problema que hi pugui haver.

3.2. Fases del projecte

En aquest apartat es descriu detalladament en què consisteix cada fase dins la planificació del projecte. El projecte, bàsicament, es divideix en 9 fases descrites a continuació..

3.2.1. Configuració de l'entorn de desenvolupament

La primera fase consisteix en baixar el codi de la primera versió de l'aplicació web (*Authoring Tool Web,* ATW), que és el punt de partida del projecte, instal·lar totes les eines per a l'entorn de desenvolupament, i configurar aquest entorn tant per a l'aplicació de client com per a l'aplicació de servidor.

3.2.2. Autoaprenentatge

Degut a que a la primera versió de l'ATW s'usa la tecnologia *AngularJS,* i degut al meu desconeixement d'aquesta tecnologia, aquesta fase consistirà en consolidar els coneixements necessaris sobre aquesta tecnologia per poder desenvolupar el projecte. A partir de mig període d'autoaprenentatge es començarà la primera iteració de desenvolupament.

3.2.3. Gestió del projecte

En aquesta fase se segueixen les indicacions marcades al mòdul de GEP amb l'objectiu de fer una bona gestió del projecte. En concret es defineixen l'abast, contextualització, objectius i els requisits que haurà de complir el projecte, la planificació temporal, la gestió econòmica i l'estudi de sostenibilitat. Les entregues dels documents amb aquestes definicions tenen unes dates estrictes que s'han de complir. Aquesta fase té una durada aproximadament d'un mes i es fa paral·lelament amb la primera iteració.

3.2.4. Primera iteració: Construcció del panell de correctors

A partir d'aqueta fase comencen les iteracions del projecte. L'objectiu d'aquesta fase es desenvolupar el panell de correctors que permetrà a l'usuari afegir, modificar, o eliminar correctors, i també executar les qüestions assignades a un corrector i refrescar la vista del panell.

Dins la iteració es realitzarà totes les tasques del cicle de vida de la metodologia àgil (anàlisi de requisits, especificació, implementació i *testing*), cada una és dependent de l'anterior i en ordre seqüencial.

3.2.5 Segona iteració: Construcció del panell d'execució

Aquesta és la segona iteració corresponent al desenvolupament del projecte. La ATW de partida té un panell per a fer afegir, modificar o eliminar qüestions. L'objectiu d'aquesta fase és afegir un sub-panell d'execució al aquest panell. El sub-panell d'execució es podrà usar per assignar i des-assignar els correctors que s'utilitzaran per executar la solució de la qüestió, modificar l'ordre de jocs de proves de la qüestió, generar el fitxer adjunt que es proporciona als estudiants per resoldre la qüestió, i executar la qüestió visualitzant el resultat de l'execució.

S'afegirà funcionalitat com pujar fitxer adjunt, generar el fitxer .txt que conté informació de la qüestió i copiar bases de dades de la qüestió.

Es seguirà també la mateixa metodologia àgil que en la primera iteració basat en el cicle de vida del desenvolupament de *software*.

3.2.6 Tercera iteració: Construcció del panell solucions d'estudiants i millorar funcionalitats del panell de qüestió

Aquesta és la tercera iteració corresponent al desenvolupament del projecte. L'objectiu d'aquesta fase és desenvolupar un altre sub-panell nou adjunt al panell de gestió de qüestions. El sub-panell d'execució es podrà usar per a que el professor pugui provar i emmagatzemar solucions típiques que els estudiants fan de la qüestió. Aquestes solucions poden ser correctes i alternatives a la solució del professor, o incorrectes. Des d'aquest panell es podrà afegir o eliminar solucions, i executar solucions individualment o en bloc, veient quina nota traurien tenint en compte els jocs de proves i pesos dels jocs de proves definits per a la qüestió.

Es seguirà també la mateixa metodologia àgil que en la primera iteració basat en el cicle de vida del desenvolupament de *software*.

3.2.7 Quarta iteració: Construcció del panell de zip genèric i construcció de la secció de Resultats

Aquesta és la quarta iteració corresponent al desenvolupament del projecte. Els objectius d'aquesta fase són: primerament desenvolupar una secció on els professors poden definir el zip genèric seleccionant un fitxer zip local, i posteriorment es podrà fer una generació automàtica del fitxer zip per a les preguntes de tipus JDBC. Segon, construir una subsecció dins de la secció qüestió on es pot visualitzar per cadascuna de les possibles respostes de la qüestió, els joc de proves, comprovacions són correcte o no, i una taula on mostra la informació dels grups per cada curs de Moodle-LearnSQL. Finalment, es desenvolupar la funcionalitat de copiar un joc de proves, i copiar la comprovació a tots els jocs de proves d'una qüestió.

3.2.8 Cinquena iteració: Testing de funcionalitats totals

Un cop acabat de complir tots els requisits i objectius, es canviarà la base de dades de desenvolupament per la de producció que conté el repositori de preguntes de les assignatures, els professors faran *testings* i proves de les funcionalitats implementades de les primeres 4 iteracions.

3.2.9 Etapa final

Durant aquesta última etapa del projecte, s'haurà de fer la documentació de la memòria final del projecte, que constarà tota la documentació de la fase GEP, amb l'anàlisi de requisits i disseny de cadascun de les iteracions.

3.3. Temps estimat

A la *Taula 1* es pot veure les hores dedicades de diferents fases

Fase	Hores
Configuració de l'entorn de desenvolupament	48
Autoaprenentatge	46
Gestió del projecte	104
Primera iteració	96
Segona iteració	105
Tercera iteració	120
Quarta iteració	56
Cinquena iteració	18
Etapa final	52
	Total de 645 hores

Taula 1: Estimació d'hores

3.4. Recursos

En aquesta apartat es detallen els recursos necessaris per tal de desenvolupar el projecte.

- Recursos humans

Aquest projecte el fa una sola persona que s'assumeix tots els rols necessaris (analista, programador, *tester*) amb una dedicació de 30 hores setmanals aproximadament.

- Recursos *hardware*

Un portàtil *Acer* amb sistema operatiu Windows 7 (64 bits) i servidors de la Facultat d'Informàtica de Barcelona (FIB) de la Universitat Politècnica de Catalunya.

- Recursos *software*
    - Eclipse: S'utilitza per al desenvolupament de l'aplicació de servidor.
    - pgAdmin III: S'utilitza per gestionar bases de dades del projecte.
    - Sublime Text 3: S'utilitza com editor per al desenvolupament de l'aplicació de client.
    - Juniper Network[12]: S'utilitza per connectar al VPN de a la Universitat Politècnica de Catalunya, així com es pot treballar el projecte des de casa.
    - Google Chrome: S'utilitza per entrar al localhost del projecte.
    - Windows PowerShell x86: S'utilitza per compilar l'aplicació de client, i instal·lar les dependències necessàries d'AngularJS.
    - Git Shell: S'utilitza per fer el control versió per tal de emmagatzemar els codis al repositori *Bitbucket.*
    - Gantter: Eina *on-line* que es farà servir per construir el diagrama de Gantt.

3.5. Valoració d'alternatives i pla d'acció

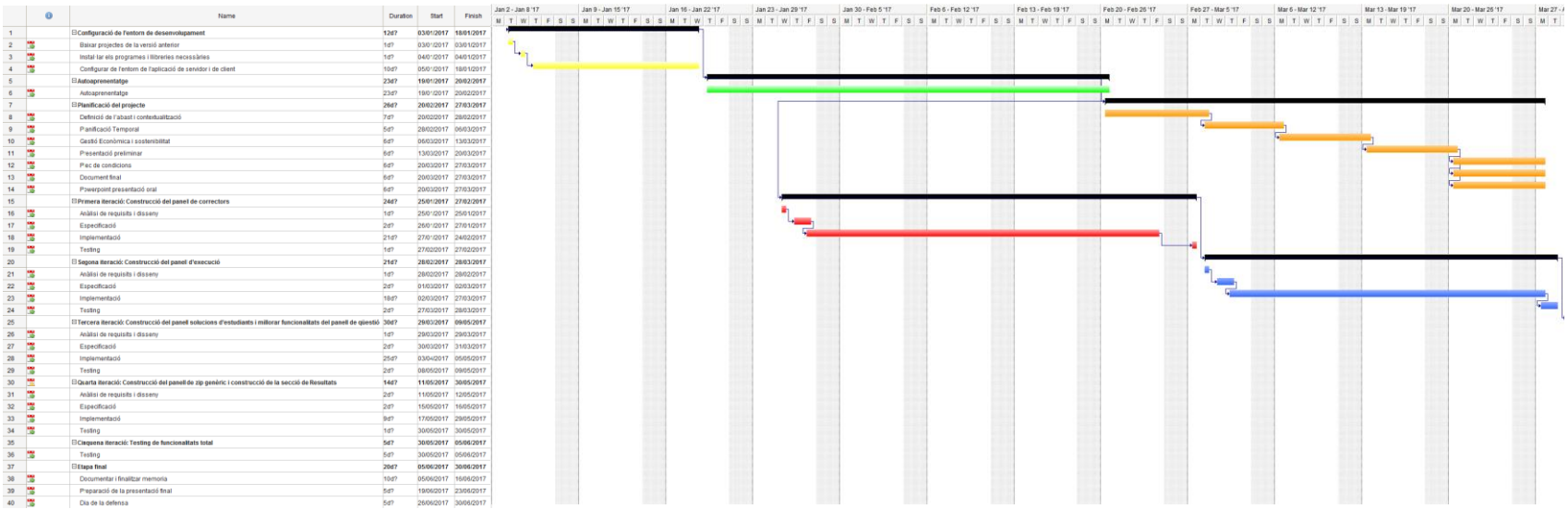
Per poder gestionar les alternatives i el pla d'acció en el cas d'una desviació important en el calendari previst es prendrà com a referència el final de cadascuna de les tasques anteriorment descrites.

Pel fet de que el projecte l'ha de realitzar una sola persona, això pot suposar certes desviacions en la planificació, per tant, s'ha de tenir compte un marge de 4 setmanes des de l'etapa final fins al dia de la defensa del projecte. D'aquesta manera es garantirà un marge de temps en cas de que hi pugui haver alguna funcionalitat d'iteració a la que s'hagi de dedicar més temps del que s'havia planificat. Ara bé, en el cas d'acumular diversos retards que sumin un total de 7 dies, s'haurà de fer una replanificació per tal d'ajustar el calendari i incrementar el temps de dedicació en les tasques pendents, i finalitzar el projecte en el termini indicat.

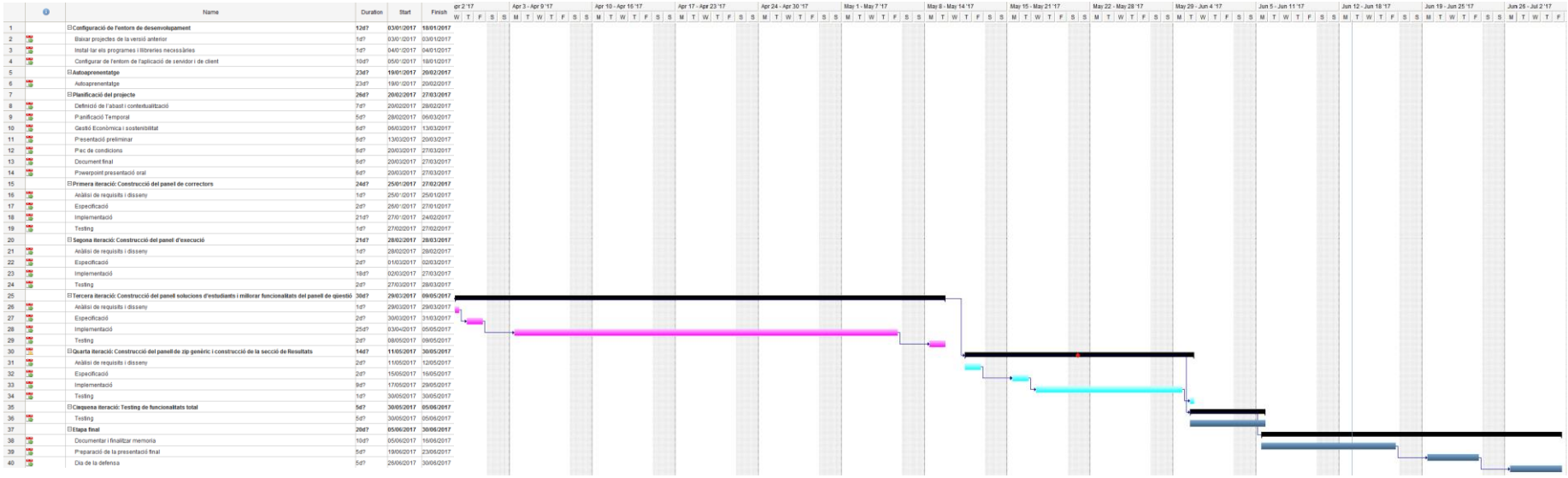
En qualsevol cas, les possibles desviacions afectaran principalment als recursos humans, i estaria disposat a incrementar el nombre d'hores de dedicació per tal de garantir la finalització del projecte en el temps establert.

3.6. Diagrama de Gantt

A la *Figura* 4 es pot veure el diagrama de Gantt que representa la planificació temporal del projecte, inclòs totes les fases i les tasques mencionades amb la seva data inici, data final i la duració en dies.



	Name	Duration	Start	Finish
1	**Configuració de l'entorn de desenvolupament**	**12d?**	**03/01/2017**	**18/01/2017**
2	Baixar projectes de la versió anterior	1d?	03/01/2017	03/01/2017
3	Instal·lar els programes i llibreries necessàries	1d?	04/01/2017	04/01/2017
4	Configurar de l'entorn de l'aplicació de servidor i de client	10d?	05/01/2017	18/01/2017
5	**Autoaprenentatge**	**23d?**	**19/01/2017**	**20/02/2017**
6	Autoaprenentatge	23d?	19/01/2017	20/02/2017
7	**Planificació del projecte**	**26d?**	**20/02/2017**	**27/03/2017**
8	Definició de l'abast i contextualització	7d?	20/02/2017	28/02/2017
9	Panificació Temporal	5d?	28/02/2017	06/03/2017
10	Gestió Econòmica i sostenibilitat	6d?	06/03/2017	13/03/2017
11	Presentació preliminar	6d?	13/03/2017	20/03/2017
12	Pec de condicions	6d?	20/03/2017	27/03/2017
13	Document final	6d?	20/03/2017	27/03/2017
14	Powerpoint presentació oral	6d?	20/03/2017	27/03/2017
15	**Primera iteració: Construcció del panel de correctors**	**24d?**	**25/01/2017**	**27/02/2017**
16	Anàlisi de requisits i disseny	1d?	25/01/2017	25/01/2017
17	Especificació	2d?	26/01/2017	27/01/2017
18	Implementació	21d?	27/01/2017	24/02/2017
19	Testing	1d?	27/02/2017	27/02/2017
20	**Segona iteració: Construcció del panell d'execució**	**21d?**	**28/02/2017**	**28/03/2017**
21	Anàlisi de requisits i disseny	1d?	28/02/2017	28/02/2017
22	Especificació	2d?	01/03/2017	02/03/2017
23	Implementació	16d?	02/03/2017	27/03/2017
24	Testing	2d?	27/03/2017	28/03/2017
25	**Tercera iteració: Construcció del panell solucions d'estudiants i millorar funcionalitats del panell de qüestió**	**30d?**	**29/03/2017**	**09/05/2017**
26	Anàlisi de requisits i disseny	1d?	29/03/2017	29/03/2017
27	Especificació	2d?	30/03/2017	31/03/2017
28	Implementació	25d?	03/04/2017	05/05/2017
29	Testing	2d?	08/05/2017	09/05/2017
30	**Quarta iteració: Construcció del panell de zip genèric i construcció de la secció de Resultats**	**14d?**	**11/05/2017**	**30/05/2017**
31	Anàlisi de requisits i disseny	2d?	11/05/2017	12/05/2017
32	Especificació	2d?	15/05/2017	16/05/2017
33	Implementació	8d?	17/05/2017	29/05/2017
34	Testing	1d?	30/05/2017	30/05/2017
35	**Cinquena iteració: Testing de funcionalitats total**	**5d?**	**30/05/2017**	**05/06/2017**
36	Testing	5d?	30/05/2017	05/06/2017
37	**Etapa final**	**20d?**	**05/06/2017**	**30/06/2017**
38	Documentar i finalitzar memoria	10d?	05/06/2017	16/06/2017
39	Preparació de la presentació final	5d?	19/06/2017	23/06/2017
40	Dia de la defensa	5d?	26/06/2017	30/06/2017



	Name	Duration	Start	Finish
1	**Configuració de l'entorn de desenvolupament**	**12d?**	**03/01/2017**	**18/01/2017**
2	Baixar projectes de la versió anterior	1d?	03/01/2017	03/01/2017
3	Instal·lar els programes i llibreries necessàries	1d?	04/01/2017	04/01/2017
4	Configurar de l'entorn de l'aplicació de servidor i de client	10d?	05/01/2017	18/01/2017
5	**Autoaprenentatge**	**23d?**	**19/01/2017**	**20/02/2017**
6	Autoaprenentatge	23d?	19/01/2017	20/02/2017
7	**Planificació del projecte**	**26d?**	**20/02/2017**	**27/03/2017**
8	Definició de l'abast i contextualització	7d?	20/02/2017	28/02/2017
9	Planificació Temporal	5d?	28/02/2017	06/03/2017
10	Gestió Econòmica i sostenibilitat	6d?	06/03/2017	13/03/2017
11	Presentació preliminar	6d?	13/03/2017	20/03/2017
12	Pac de condicions	6d?	20/03/2017	27/03/2017
13	Document final	6d?	20/03/2017	27/03/2017
14	Powerpoint presentació oral	6d?	20/03/2017	27/03/2017
15	**Primera iteració: Construcció del panell de correctors**	**24d?**	**25/01/2017**	**27/02/2017**
16	Anàlisi de requisits i disseny	1d?	25/01/2017	25/01/2017
17	Especificació	2d?	26/01/2017	27/01/2017
18	Implementació	21d?	27/01/2017	24/02/2017
19	Testing	1d?	27/02/2017	27/02/2017
20	**Segona iteració: Construcció del panell d'execució**	**21d?**	**28/02/2017**	**28/03/2017**
21	Anàlisi de requisits i disseny	1d?	28/02/2017	28/02/2017
22	Especificació	2d?	01/03/2017	02/03/2017
23	Implementació	18d?	02/03/2017	27/03/2017
24	Testing	2d?	27/03/2017	28/03/2017
25	**Tercera iteració: Construcció del panell solucions d'estudiants i millorar funcionalitats del panell de qüestió**	**30d?**	**29/03/2017**	**09/05/2017**
26	Anàlisi de requisits i disseny	1d?	29/03/2017	29/03/2017
27	Especificació	2d?	30/03/2017	31/03/2017
28	Implementació	25d?	03/04/2017	05/05/2017
29	Testing	2d?	08/05/2017	09/05/2017
30	**Quarta iteració: Construcció del panell de zip genèric i construcció de la secció de Resultats**	**14d?**	**11/05/2017**	**30/05/2017**
31	Anàlisi de requisits i disseny	2d?	11/05/2017	12/05/2017
32	Especificació	2d?	15/05/2017	16/05/2017
33	Implementació	9d?	17/05/2017	29/05/2017
34	Testing	1d?	30/05/2017	30/05/2017
35	**Cinquena iteració: Testing de funcionalitats total**	**5d?**	**30/05/2017**	**05/06/2017**
36	Testing	5d?	30/05/2017	05/06/2017
37	**Etapa final**	**20d?**	**05/06/2017**	**30/06/2017**
38	Documentar i finalitzar memoria	10d?	05/06/2017	16/06/2017
39	Preparació de la presentació final	5d?	19/06/2017	23/06/2017
40	Dia de la defensa	5d?	26/06/2017	30/06/2017

Il·lustració 4: Diagrama de Gantt

4. Pressupost i sostenibilitat

4.1. Identificació i estimació del cost

En aquest apartat es fa una anàlisis sobre l'estimació de costos del projecte. L'estimació de costos té en compte els recursos humans, els recursos de *hardware*, de *software*, altres despeses, i les despeses dels imprevistos que poden sorgir durant el període de desenvolupament del projecte.

4.1.1. Recursos humans

Aquest projecte el realitzarà una sola persona amb una dedicació de 30 hores setmanal. Aquesta persona assumirà tots els rols que intervenen en un projecte de desenvolupament de software (cap del projecte, analista, dissenyador, programador, *tester*). Per poder estimar el pressupost en recursos humans, assignem una remuneració per hora en funció dels rols que hi han en aquest projecte, a continuació es pot veure la taula de cost per hora del diferents rols basats en l'estudi de remuneració de *PagePersonnel* de l'any 2016 [13]:

Rol	Remuneració (€/h)
Cap del projecte	25
Analista	18
Dissenyador	18
Programador	9
Tester	12

Taula 2: Cost per hora de cada rol

A continuació, es pot veure una taula que mostra les hores totals que dedicarà cada rol en cada fase, tenint en compte que en cada iteració del projecte hi ha part de cada fase. El cap del projecte haurà de dedicar hores a totes les fases, ja que és la persona qui s'encarrega de planificar, i prendre decisions durant tot el període del desenvolupament. El perfil d'analista dedicarà hores a la fase de definició d'objectius i requisits, i la fase d'especificació. El perfil de dissenyador dedicarà hores a la fase de disseny. El perfil de programador es dedicarà hores en la fase d'implementació. Finalment el perfil de *tester* dedicarà hores a la fase de l'etapa final i durant *testing* de cada iteració.

Rol	Fase						Total
	Planificació	Definició d'objectius i requisits	Especificació	Disseny	Implementació	Etapa final	
Cap del projecte	44	18	24	24	16	22	148
Analista		54	42				96
Dissenyador				36			36
Programador					299		299
Tester					36	30	66

Taula 3: Estimació d'hores total de cada rol

Tenint les hores totals i la seva remuneració per hora, podem calcular la remuneració total del cada rol, a continuació es pot veure la remuneració assignada a cadascun dels rols:

Rol	Hores dedicades (h)	Remuneració per hora (€/h)	Cost estimat (€)
Cap del projecte	148	25	3700
Analista	96	18	1728
Dissenyador	36	18	648
Programador	299	9	2691
Tester	66	12	792
Total	645	-	9559

Taula 4: Estimació de costos per rol

4.1.2. Recursos software

En aquest apartat es detallaran els costos dels recursos *software* que es faran servir en el projecte, Suposem que els recursos *software* tenen una vida útil de dos anys a un ritme de treball de 8 hores diàries i 250 dies hàbils. Així el cost d'amortització del software per hora és:

Preu / (2 anys * 250 dies * 8 hores/dia).

A continuació es pot veure una taula que mostra el cost, hores dedicades, cost d'amortització cada recurs.

Recurs *software*	Cost del recurs (€)	Hores estimades (/h)	Cost d'amortització (€/h)	Cost d'amortització (€)
Windows 7 Professional (64 bits)	150 [13]	589	0,0375	22,1
Eclipse	0	420	0	
pgAdmin III	0	420	0	
Sublime Text 3	0	450	0	
Google Chrome	0	249	0	
Git shell	0	30	0	
Microsoft Office 2013	60 [14]	100	0,015	1,5
Gantter	0	5		
Gmail	0	1		
Bitbucket	0	50		
Total	-	-	-	23,6

Taula 5: Estimació de costos de recursos software

4.1.3. Recursos hardware

En aquest apartat es detallaran els costos dels recursos *hardware* que es faran servir en el projecte, s'estima que el recurs *hardware* té una vida útil de 4 anys, així el cost d'amortització serà:

Preu / (4 anys * 250 dies * 8 hores/dia)

A continuació es pot veure una taula que mostra el cost i la vida útil del *hardware.*

Recurs *hardware*	Vida útil (any)	Cost del recurs (€)	Hores estimades (/h)	Cost d'amortització (€/h)	Cost d'amortització (h)
Portàtil Acer Aspire E15 - **Processador: Intel Core i5** - **Targeta gràfica: NVIDIA Geforce 820M**	4	600	589	0,075	44,2
Total	-	-	-	-	44,2

Taula 6: Estimació de costos de recursos hardware

4.1.4. Altres despeses

Durant el desenvolupament del projecte s'utilitzarà el portàtil i el servidor amb el consum d'electricitat que això suposa, sobretot el portàtil i el servidor sempre estarà encès.

També és imprescindible la connexió a Internet per tal d'assegurar l'accés al VPN de la UPC i connexió a la base de dades.

Descripció	Preu	Quantitat	Cost estimat (€)
Consum energètic	0,12 €/kWh [15]	589 h	70,68
Connexió Internet	40 € /mes	6 mesos	240
Total			310,68

Taula 7 Estimació de costos d'altres despeses

4.1.5. Imprevistos

Durant el desenvolupament del projecte pot sorgir diversos imprevistos que afecten a la planificació de tasques definides, i al seu pressupost. Els imprevistos habitualment són:

- Retràs en hores

Es pot produir el retràs quan per alguna funcionalitat o tasca es necessiti més temps del que s'havia previst per a desenvolupar-la que el inclòs en la planificació. En aquest cas, el rol que correspongui haurà de dedicar més hores en aquesta funcionalitat o tasca. Aquest aspecte pot tenir una probabilitat aproximada d'un 50%.

- Avaries de dispositius

Es pot produir avaries de portàtil, en aquest cas es necessitarà una reparació. L'avaria pot tenir una probabilitat baixa de 10%.

- Pèrdua de dades i codi

La probabilitat és relativament baixa, ja que les dades i el codi seran emmagatzemats *en* el *BitBucket*, per tant té una probabilitat de 5%.

A continuació obtenim una taula dels imprevistos

Descripció	Quantificació	Preu (€)	Probabilitat (%)	Cost estimat (€)
Retràs en hores	30h aprox.	(25+9) €/h	50	1080
Avaries de dispositius	1 cop	100 € aprox.	10	100
Pèrdua de dades i codis	1 cop	0	5	0
Total				1180

Taula 8: Estimació de costos dels imprevistos

4.1.6. Contingències

Com a mesura de contingència s'estableix un marge del 7% de la suma dels costos directes i indirectes, per tant tenint el cost total és 10559,48 €, aplicant els 7% de contingència serà de 739,16 €.

4.1.7. Cost total

A continuació es pot veure una taula que mostra la suma de tots els recursos, altres despeses, els imprevistos, i les contingències.

Recurs	Cost estimat (€)
Recurs humà	9559
Recurs *software*	23,6
Recurs *hardware*	44,2
Altres despeses	310,68
Imprevistos	1180
Contingències	739,16 (7%)
Total	11856,64

Taula 9: Estimació de costos totals

4.2. Control de gestió

Pel fet de que alguns costos no són fixes com els de *software* i *hardware*, és necessàri fer el càlcul de les desviacions d'aquests costos per ajustar el pressupost del projecte.

- Desviació en el preu d'un recurs humà = (cost recurs humà estimat - cost recurs humà real)*hores
- Desviació en el preu d'electricitat = (consum energètic estimat - consum energètic real)*hores d'ús
- Desviació en el preu de cost imprevistos = (costos imprevistos estimats – costos imprevistos reals)
- Diferència de cost total de recursos = |cost total recursos estimat – cost total recursos real|

5. Sostenibilitat i compromís social

Es considera la sostenibilitat i compromís social des de les tres dimensions següents: la sostenibilitat econòmica, la sostenibilitat social i la sostenibilitat ambiental.

5.1. Sostenibilitat econòmica

Pel que fa a la dimensió econòmica del projecte, existeix una avaluació de costos, tant de recursos humans com materials tangibles i intangibles. S'han tingut en compte el cost dels imprevistos com reparació o retràs de hores.

Des del punt de vista de la viabilitat del projecte, el fet de que aquest projecte sigui d'àmbit acadèmic, fa que no el puguem valorar a nivell de mercat. Des del punt de vista de si el projecte, es podria fer amb menys temps, evidentment si s'augmentés el cost de recursos humans, és a dir que en compte de desenvolupar-ho una sola persona, el desenvolupessin un equip, el temps es reduiria significativament. Una altra opció seria que el desenvolupador d'aquest projecte fos una persona experta en les tecnologies que s'utilitzen en aquest projecte, cosa que també podria fer que es reduís el temps de desenvolupament.

5.2. Sostenibilitat social

La situació actual del sistema LearnSQL no és gaire favorable ja que a partir del 2009, la Generalitat de Catalunya i la Universitat Politècnica de Catalunya no es donaven suport econòmic per mantenir i estendre el sistema.

La nova versió web de l'Authoring Tool (ATW2) facilitarà als professors la gestió de qüestions ja que tindran més facilitat a l'hora de treballar i simplificar la complexitat del procés de la gestió. En concret, la nova funcionalitat de gestió de solucions comuns dels estudiants farà que el temps necessari per definir noves qüestions o fer canvis en les existents es redueixi.

5.3. Sostenibilitat ambiental

Per desenvolupar aquest projecte es necessitarà diferents dispositius, connexió a Internet, l'energia que aquests puguin consumir i els recursos *software*. Com a conseqüència d'utilitzar aquests materials, estarem consumint uns recursos que tenen un seguit de conseqüències perjudicials pel medi ambient. Ara bé, es considera que aquest impacte serà petit.

Durant el període de realització del projecte, s'intentarà utilitzar els recursos *software ja* disponibles i *Open Source Software* per estalviar temps i diners de desenvolupament reduint l'impacte del reciclatge.

5.4. Taula de sostenibilitat

La qualificació assignada a cada dimensió s'ha decidit seguint com a guia la Matriu de Sostenibilitat del TFG proporcionada com a documentació per a la realització del TFG [14]. L'anàlisi de la sostenibilitat del projecte es divideix en 3 part: PPP (*Proyecto Puesto en Producción*) que inclou la planificació, el desenvolupament i la implantació del projecte, la seva qualificació de cada dimensió de sostenibilitat es fa sobre un rang de 0 a 10; Vida útil que és el període que un cop comença un implantat fins la seva desmantellament, i la seva qualificació de cada dimensió de sostenibilitat es fa sobre un rang de 0 a 20; Riscos que són riscos que podran haver durant tota seva construcció i la vida útil, i la seva qualificació de cada dimensió de sostenibilitat es fa sobre un rang de -20 a 0.

- **PPP**

En l'apartat econòmic es considera que és molt sostenible sempre tenint en compte el factor acadèmic i que no es possible la seva comercialització.
Mentre que en l'apartat social és on la sostenibilitat serà quasi excel·lent ja que millorarà considerablement la qualitat del treball i simplificarà la complexitat del procés de la gestió, ja que es tracta del desenvolupament d'una aplicació que ofereix una necessitat el qual altres aplicacions no poden satisfer.
Finalment, en l'apartat ambiental estimo que és notable. Els recursos necessaris per al desenvolupament del projecte són ordinadors i servidors que tenen un consum energètic que té un impacte petit.

- **Vida útil**

En l'apartat econòmic l'impacte serà petit, ja que es necessitarà un manteniment i una reparació considerablement menor.
Mentre que en l'apartat ambiental i social l'explicació serà similar a l'apartat de PPP.

- **Riscos**

Només es considera un risc de baix nivell el que l'usuari de l'aplicació no tingui suficients coneixements per utilitzar les noves funcionalitats de l'aplicació. Aquest és un risc baix ja que el desenvolupador intentarà dissenyar una interfície de l'aplicació que sigui el més usable possible i que utilitzi els conceptes coneguts dels professors.

Sostenibilitat	PPP	Vida útil	Riscos	Total
Sostenibilitat econòmica	9	5	0	
Sostenibilitat social	9	10	-5	
Sostenibilitat ambiental	8	9	0	
Total de sostenibilitat	26	24	-5	45

Taula 10: Valoració de sostenibilitat

6. Estudi intern de la primera versió de l'ATW

Una part molt important del projecte és entendre bé l'aplicació de partida, ATW per entendre com funciona, i com continuar desenvolupant a partir de les funcionalitats existents, i les tecnologies utilitzades. Aquest estudi doncs, consisteix bàsicament en analitzar les funcionalitats existents en l'ATW, veure com es va dissenyar i implementar les parts servidor i client de l'aplicació, i saber les tecnologies que s'utilitzaven.

En l'apartat 6.1 s'enumeren les funcionalitats de domini existents en l'aplicació de partida. En l'apartat 6.2 es detallen altres funcionalitats d'interfície que complementen i faciliten l'ús de les funcionalitats de domini. En l'apartat 6.3 s'introdueix el com es va fer el disseny i implementació de les parts client i servidor. Finalment l'apartat 6.4 indica algunes conclusions a les que vaig poder arribar després de l'estudi.

6.1. Funcionalitats de domini existents

Les funcionalitats de domini existents que ofereix la primera versió de l'ATW són:

- Gestió de qüestions (crear | eliminar | modificar).
- Gestió de jocs de proves de la qüestió (crear | eliminar | modificar).
- Gestió de comprovacions de jocs de proves (crear | eliminar | modificar).
- Gestió de categories (crear | eliminar | modificar).
- Gestió de temàtiques (crear | eliminar | modificar).
- Gestió d'esquemes (crear | eliminar | modificar).
- Modificació de pesos de cada joc de proves.
- Cerques de qüestions aplicant filtres segons uns paràmetres que s'introdueixen.

6.2. Funcionalitats d'interfície existents

Les funcionalitats d'interfície existents que ofereix la primera versió de l'ATW són:

- S'accedeix a l'aplicació mitjançant una URL, i es disposa d'una pantalla d'autenticació.
- L'interfície de l'aplicació es pot usar en Català, Castellà, i Anglès.
- Es mostra un llistat de totes les qüestions a la barra lateral.
- Es mostra un llistat de totes les categories a la barra lateral.
- Es mostra un llistat de totes les temàtiques a la barra lateral.
- Es mostra un llistat de tots els esquemes a la barra lateral.
- Es mostra un llistat de tots els jocs de proves d'una qüestió.
- Es mostra un llistat de totes les comprovacions del joc de proves.
- La barra superior conté el menú de seccions principals.
- Es pot fer un seguiment de paràmetres de la qüestió que s'estan modificant.
- Es manté l'estat del que s'està fent en canviar de pestanya.
- Indicadors d'estat d'entitat (qüestió, categoria, temàtica, esquema, etc..).

6.3. Disseny i implementació

La primera versió de l'ATW es va construir com una una *Single Page Application* [15]. El disseny seguint aquest enfocament consisteix en desenvolupar dues aplicacions (client i servidor) que estan connectades, i comunicades entre si utilitzant el format d'intercanvi *JSON (Javascript Object Notation)* [16].

L'arquitectura d'aplicacions que segueixen una SPA es representa de la següent manera:

Il·lustració 5: L'arquitectura SPA

A continuació es descriu com era l'arquitectura de cadascuna de les aplicacions client i servidor.

6.3.1. Aplicació client

L'aplicació client s'encarrega d'interactuar amb l'usuari, mostrar informació a l'usuari, i d'enviar peticions a l'aplicació servidor.

6.3.1.1. Arquitectura

Dins de l'aplicació client es va usar una arquitectura en tres capes: capa de presentació, capa de domini, i capa de gestió de dades. La capa de gestió de dades es comunica amb els serveis web de l'aplicació de servidor. La capa de presentació s'implementa utilitzant el patró de disseny *Model-Vista-Controlador (MVC)* [17], i s'ha escollit *frameworks MVC AngularJS*.

6.3.1.2. Mòduls

A la capa de presentació hi havia una quantitat elevada de components, i per poder facilitar-ne l'organització i representació es van agrupar els components amb una mateixa temàtica en mòduls. Els mòduls de l'aplicació de partida eren els següents:

- **App**: agrupa els components que són essencials per l'arrancada i la configuració de l'aplicació.
- **Commons**: agrupa tots els components que són útils per a la resta de mòduls.
- **Questions**: agrupa tots els components relacionats amb els casos d'ús de Qüestió.
- **Categories**: agrupa tots els components relacionats amb els casos d'ús de Categoria.
- **Thematics**; agrupa tots els components relacionats amb els casos d'ús de Temàtica.
- **Schemas**: agrupa tots els components relacionats amb els casos d'ús d'Esquema.

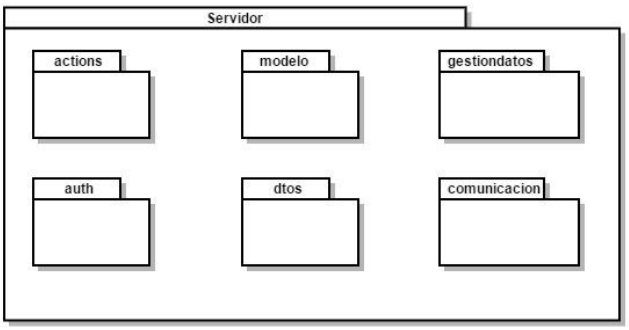
6.3.2. Aplicació servidor

L'aplicació servidor s'encarrega de rebre les dades que envia l'aplicació client, de processar-les, i de tornar a enviar les dades un cop processades a l'aplicació client, té la responsabilitat de la lògica de negoci del sistema. L'aplicació servidor es va desenvolupar en *JAVA*, utilitzant el *framework Struts2* [18], i el patró d'implementació *Front-Controller* [19] per a processar les peticions de l'aplicació client.

6.3.2.1. Arquitectura

L'aplicació servidor està organitzada en els següents paquets:

- **actions**: conté les classes que s'encarrega d'atendre les peticions *HTTP* que venen de l'aplicació client.
- **modelo**: conté les classes *Service* que tenen la responsabilitat de fer intermediari entre *Actions* i les operacions d'accés a dades, i les classes d'entitat de domini.
- **gestiondatos:** conté les classes *Controller* que fan l'accés a bases de dades del sistema.
- **auth:** conté les classes que gestiona el procés d'autenticació.
- **dtos:** conté les classes *DTO* [20] *(Data Transfer Object).*
- **comunicació:** conté les classes que fan la comunicació amb servei web corrector.



Il·lustració 6: Organització de paquets de domini

6.3.2.2. Taula d'operacions *API REST*

L'aplicació servidor està dissenyada com una *REST API* [21], on cada petició es correspon amb un cas d'ús del sistema, utilitza els 4 mètodes proporcionats per *HTTP* per usar els recuros. Els 4 mètodes són: GET, POST, PUT, DELETE. Cada petició s'identifica amb una *URL*. Només permet fer les peticions que tinguin el namespace */ws* un cop l'aplicació client ja ha fet el logueig al servidor. Les peticions implementades en la primera versió de l'ATW són les que es poden veure a la Taula 10, que corresponen a les funcionalitats indicades en les seccions anteriors (6.1 i 6.2).

URL	Mètode *HTTP*	Petició	Resposta	Descripció
/ **/qüestions*** **/categories*** **/thematics***	GET		HTML	Serveix l'aplicació client
/login	GET		HTML	Mostra el formulari d'accés
/login	POST		Redirect	Processa una petició d'accés
/logout	GET		Redirect	Finalitza la sessió
/ws/questions	GET		JSON	Retorna el llistat de totes les qüestions
/ws/questions	POST	Dades de la nova qüestió (*JSON)*	JSON	Crea una nova qüestió
/ws/qüestions/{id}	PUT	Noves dades de la qüestió {id}	JSON	Modifica la qüestió {id} i retorna dades actualitzada de la qüestió
/ws/qüestions/{id}	DELETE		JSON	Esborra la qüestió {id}
/ws/qüestions/{id}/copy	POST		JSON	Copia la qüestió {id} a la mateixa base de dades
/ws/qüestions/{id}/lock	POST		JSON	Bloqueja la qüestió {id}
/ws/categories	POST	Dades de la nova categoria (*JSON)*	JSON	Crea una nova categoria
/ws/categories/{id}	PUT	Noves dades de la categoria	JSON	Modifica la categoria {id} i retorna dades actualitzada de la categoria
/ws/categories/{id}	DELETE		JSON	Esborra la categoria {id}
/ws/thematics	GET		JSON	Retorna el llistat de totes les categories
/ws/thematics	POST	Dades de la nova temàtica (*JSON)*	JSON	Crea una nova temàtica
/ws/thematics/{id}	PUT	Noves dades de la temàtica	JSON	Modifica la temàtica {id} i retorna dades actualitzada de la temàtica
/ws/thematics/{id}	DELETE		JSON	Esborra la temàtica {id}
/ws/schemas	GET		JSON	Retorna el llistat de tots els esquemes
/ws/schemas	POST	Dades del nou esquema (*JSON)*	JSON	Crea un nou esquema
/ws/schemas/{id}	PUT	Noves dades de l'esquema	JSON	Modifica l'esquema {id} i retorna dades actualitzada de l'esquema
/ws/schemas/{id}	DELETE		JSON	Esborra l'esquema {id}
/ws/solutiontypes	GET		JSON	Retorna el llistat de tots els tipus de solució

Taula 10: Taula d'operacions API REST

6.4. Conclusió de l'estudi

Com a conclusió, l'aplicació ATW de partida és una aplicació amb un disseny i implementació acurats que pot ser fàcilment extensible i modificable. Per tant, per aprofitar al màxim la bona feina realitzada de la primera versió, la segona versió de l'ATW segueix i estén el mateix disseny i segueix utilitzant les mateixes tecnologies, mateixos patrons etc.. Per altra banda, aquest també era el requisit marcat en iniciar el TFG per la directora i usuària de la futura ATW2.

És a dir, l'aplicació client mantindrà l'arquitectura de tres capes, i serà una aplicació desenvolupada en els llenguatges natius del desenvolupament de web (*HTML, Javascript, CSS)*, utilitzarà el *framework AngularJS* a la capa de presentació, beneficiant-nos de la seva implementació de l'*MVC*. En quan a l'aparença i distribució d'elements, utilitzarà la nova especificació de *CSS* anomenada *Flexbox* [22].

L'aplicació servidor utilitzarà el *framework Struts2*, beneficiant-nos de la seva implementació de *Front-Controller*¸ i se seguirà dissenyant com una *REST API*.

La forma en que es farà la comunicació entre l'aplicació servidor i l'aplicació client continuarà sent mitjançant el *JSON.*

7. Anàlisis de requisits

En aquest capítol s'analitzaran els requisits funcionals i no funcionals que s'ha d'incorporar en l'ATW2 per a més endavant fer-ne l'especificació. disseny i implementació.

7.1. Objectius detallats

A continuació es detallaran els objectius que es desitja que satisfaci l'ATW2. Per facilitar fer referència als objectius més endavant, s'identifica cada objectiu amb l'identificador "OB".

OB #1: Desenvolupament de correctors	
Descripció	L'aplicació proporcionarà un llistat de tots els correctors (també anomenats Scorers), i per cada corrector es mostrarà la informació del corrector en detall (URL, descripció, categories, estat del corrector, i qüestions associades), i es proporcionarà la gestió de correctors, és a dir, l'alta, baixa, modificació i execució de les qüestions assignades a un corrector.

OB #2: Generació del fitxer adjunt	
Descripció	L'aplicació permetrà fer la generació automàtica del fitxer adjunt de les qüestions. Per fer-ho l'usuari podrà seleccionar el contingut que el fitxer ha de tenir. Si la qüestió és de tipus JDBC, es generarà el fitxer adjunt de manera automàtica en funció del contingut del zip genèric (OB# 7) i altres atributs de la qüestió. Laplicació permetrà també pujar el fitxer adjunt de la qüestió, en cas que l'usuari desitgi generar aquest fitxer fora de l'aplicació.

OB #3: Desenvolupar la còpia de qüestions a altres bases de dades	
Descripció	L'aplicació permetrà mostrar un formulari de connexió a una altra base de dades, i copiar una qüestió a la base de dades indicada.

OB #4: Desenvolupar la pestanya d'execució de qüestions	
Descripció	L'aplicació permetrà veure per una qüestió els correctors assignats, modificar els correctors assignats, i executar la qüestió en cadascun dels correctors assignats. També es podrà veure el resultat de l'execució de cadascun dels jocs de proves i comprovacions de la qüestió, i el log amb el resultat de les execucions que es van fent. Finalment es permetrà canviar l'ordre de correcció dels jocs de proves i netejar el log d'execucions.

OB #5: Desenvolupar la pestanya d'execució de possibles respostes	
Descripció	L'aplicació permetrà gestionar per una qüestió les possibles respostes a la qüestió. En concret permetrà fer alta, baixa, modificació, i execució de les respostes d'una en una o de totes les respostes en bloc . També s'haurà de mostrar un log d'execucions, i netejar els resultats de les execucions.

OB #6: Gestió de zip genèric	
Descripció	L'aplicació permetrà seleccionar un fitxer local, i carregar-ho per configurar el zip genèric del sistema que es fa servir per la generació automàtica del fitxer adjunt en qüestions JDBC, també permetrà visualitzar un llistat del contingut del fitxer zip genèric.

OB #7: Consultar resultats	
Descripció	L'aplicació permetrà consultar informació per ajudar als professors en la definició d'una qüestió, específicament dels seus jocs de proves i comprovacions. Es permetrà consultar dos tipus de dades. La primera és per cada possible resposta d'una qüestió, els jocs de proves/comprovacions que la qüestió passa favorablement o no passa. La segona és per cada vegada que una qüestió ha estat posada en un qüestionari en un curs de la FIB, per cada grup d'estudiants que van fer el qüestionari, les notes mitjanes que varen treure els estudiants i la quantitat d'estudiants que van donar una resposta del total del curs.

7.2. Requisits funcionals

Els requisits funcionals defineixen uns serveis o funcions que ha de proporcionar el sistema, de manera que aquest ha de reaccionar les entrades particulars, i com s'ha de comportar en situacions particulars.

7.2.1. Llistat de requisits funcionals

A continuació es llista tots els nous requisits funcionals d'aquesta segona versió del sistema basat en els objectius definits en el capítol anterior.

- **RF1. Consulta de correctors:** Els professors podran consultar tots els correctors amb les dades corresponent.
- **RF2. Crear un corrector:** Els professors podran crear un nou corrector.
- **RF3. Modificar el corrector:** Els professors podran modificar les dades d'un corrector.
- **RF4. Eliminar el corrector:** Els professors podran eliminar un corrector un cop seleccionar un corrector per veure les dades.
- **RF5. Executar el corrector:** Els professors podran executar les qüestions assignades del corrector, i el sistema ha de ser capaç de mostrar el resultat d'execució de cada qüestió.
- **RF6. Refrescar el corrector:** Els professors podran refrescar el corrector per obtenir l'estat actual del corrector.
- **RF7. Modificar el fitxer adjunt:** Els professors podran seleccionar un fitxer local per modificar el fitxer adjunt de la qüestió.
- **RF8. Eliminar el fitxer adjunt:** Els professors podran eliminar el fitxer adjunt de la qüestió.
- **RF9. Guardar el fitxer adjunt:** Els professors podran descarregar el fitxer adjunt de la qüestió.
- **RF10. Generar el fitxer adjunt:** Els professors podran seleccionar el contingut del fitxer adjunt, i grabar les dades del fitxer adjunt de les qüestions de tipus JDBC
- **RF11. Generar llistats:** Els professors podran seleccionar el contingut de la informació general de la qüestió per poder descarregar en un fitxer text pla.
- **RF12. Copiar la qüestió a altres bases de dades (CopyBD):** Els professors podran introduir dades per connectar a altres bases de dades i copiar la qüestió.
- **RF13. Executar la qüestió:** Els professors podran executar la qüestió, i el sistema ha de capaç de donar la solució de cada joc de proves de la qüestió, i mostrar el log d'execució.

- **RF14. Canviar l'ordre de joc de proves o comprovació:** Els professors podran ordenar jocs de proves amb el botó de "Pujar" i el botó de "Baixar".

- **RF15. Consulta de solucions:** Els professors podran consultar totes les solucions de la qüestió.

- **RF16. Crear la solució:** Els professors podran crear una nova solució.

- **RF17. Modificar la solució:** Els professors podran modificar les dades d'una solució de la qüestió.

- **RF18. Eliminar la solució:** Els professors podran eliminar d'una solució de la qüestió.

- **RF19. Executar la solució o diverses solucions en bloc:** Els professors podran executar una solució o més d'una solució en bloc i visualitzar el resultat d'execució.

- **RF20. Modificar el zip genèric:** Els professors podran seleccionar un fitxer local en extensió .zip, i carregar-ho per modificar el zip genèric.

- **RF21. Consultar resultats de la qüestió:** Els professors podran visualitzar per cada possible resposta d'una qüestió, els jocs de proves/comprovacions que la qüestió passa favorablement o no passa. I podran consultar informacions de resultats de la qüestió provinent del Moodle-LearnSQL sobre la nota mitjana sense penalitzacions, nota mitjana amb penalitzacions, número d'equips contestats, i número d'equips que podien contestar per ajudar en la definició de preguntes.

- **RF22. Duplicar un joc de proves o una comprovació:** Es dupliquen totes les seves dades i totes les seves comprovacions si es duplica un joc de proves.

A continuació es mostra que tots els objectius tenen un o més requisits que garantiran el seu assoliment.

	OB #1	OB #2	OB #3	OB #4	OB #5	OB #6	OB #7
RF1	X						
RF2	X						
RF3	X						
RF4	X						
RF5	X						
RF6	X						
RF7		X					
RF8		X					
RF9		X					
RF10		X					
RF12			X				
RF13				X			
RF14				X			
RF15					X		
RF16					X		
RF17					X		
RF18					X		
RF19					X		
RF20						X	
RF21							X

7.3. Requisits no funcionals

Els requisits no funcionals defineixen les restriccions dels serveis o funcions ofertes pel sistema. Són condicions que requereixen que el software o la seva execució tingui unes certes propietats. Inclouen les restriccions de percepció, usabilitat, rendiment, manteniment i suport, i seguretat.

La descripció dels requisits no funcionals es representa amb una plantilla que té la forma següent:

Requisit #:	*1	Tipus de requisit:	*2
Descripció: *3			
Justificació: *4			
Criteri de satisfacció: *5			

- ***1:** Identificador del requisit no funcional
- ***2:** Tipus del requisit no funcional
- ***3:** Descripció del requisit no funcional
- ***4:** Motiu pel qual és necessari el requisit
- ***5:** Què ha de succeir per complir aquest el requisit

7.3.1. Requisits de percepció

Requisit #:	1	Tipus de requisit:	Requisit d'aparença
Descripció:	El disseny de la interfície serà atractiu i d'ús senzill per a tots els usuaris.		
Justificació:	Al ser una aplicació enfocada de cara a l'usuari, la seva satisfacció utilitzant-la és vital. S'ha d'aconseguir que els ususaris se sentin atrets per l'aplicació.		
Criteri de satisfacció:	L'aplicació compleix les recomanacions d'estil definides per *Google Material Design* [23]. Aquest són, indicadors d'estat que contribueixen a una bona experiència d'usuari.		

7.3.2. Requisits d'usabilitat i humanitat

Requisit #:	2	Tipus de requisit:	Requisit de facilitat d'ús
Descripció:	L'aplicació compleix els estàndards ISO d'usabilitat.		
Justificació:	És molt important que l'usuari sàpiga de bon principi com s'utilitza l'aplicació, per tant la interfície haurà de ser de ser senzilla i intuïtiva.		
Criteri de satisfacció:	Es comprova que es compleix l'estàndard ISO-9241 [24].		

Requisit #:	3	Tipus de requisit:	Requisit de comprensió i cortesia
Descripció:	El llenguatge que s'usa en la interfície ha de ser correcte, comprensible, concís i respectuós.		
Justificació:	Els usuaris han d'entendre els conceptes i dades que es mostrïn en la interfície de l'aplicació.		
Criteri de satisfacció:	Una persona amb un alt coneixement dels tres idiomes en que es pot veure la interfície verificarà que el llenguatge és correcte i entenedor.		

Requisit #:	4	Tipus de requisit:	Requisit d'accessibilitat
Descripció:	L'aplicació ha de seguir unes pautes fer fer-la accessible, tindrà un entorn accessible per a gent amb discapacitat.		
Justificació:	Facilitat d'accés i d'ús per qualsevol usuari.		
Criteri de satisfacció:	Es comprovarà que es compleixen la ISO-21542 [24].		

7.3.3. Requisits de rendiment

Requisit #:	5	Tipus de requisit:	Requisit de velocitat i latència
Descripció:	L'aplicació ha de ser capaç de donar respostes rapides a les peticions.		
Justificació:	El temps de comunicació entre l'aplicació client i l'aplicació servidor ha de ser més ràpid possible.		
Criteri de satisfacció:	Es farà proves de temps de resposta a les peticions bàsiques.		

Requisit #:	6	Tipus de requisit:	Requisit de disponibilitat
Descripció:	L'aplicació ha der ser capaç d'emmagatzemar una ultima versió de les dades obtingudes.		
Justificació:	L'aplicació client ha de mantenir una copia de les dades que està treballant quan algun moment hi ha una desconnexió del servidor.		
Criteri de satisfacció:	Es farà proves parant i recuperant l'accés a la xarxa duran l'ús de l'aplicació i gravant els canvis posteriorment.		

7.3.4. Requisits de manteniment i suport

Requisit #:	7	Tipus de requisit:	Requisit de manteniment
Descripció:	L'aplicació ha de ser fàcilment mantenible.		
Justificació:	L'aplicació ha de poder evolucionar cap a noves versions amb més funcionalitats i millores.		
Criteri de satisfacció:	Es realitzarà un desenvolupament de l'aplicació en capes, intentant que hi hagi les menys dependències entre capes possibles.		

Requisit #:	8	Tipus de requisit:	Requisit d'adaptabilitat
Descripció:	L'aplicació serà compatible amb els navegadors Chrome i Firefox. Els usuaris han de poder visualitzar l'aplicació sense problemes independentment de la mida de la pantalla amb la qual s'accedeix.		
Justificació:	Aquests dos navegadors són els usats pels professors. La mida de la pantalla varia en cas d'accedir des de portàtils quan els professors estan en les aules d'examen o quan usen l'aplicació des del sobretaula del despatx.		
Criteri de satisfacció:	Es faran proves de l'aplicació en tots els entorns indicats.		

7.3.5. Requisits de seguretat

Requisit #:	9	Tipus de requisit:	Requisit d'accés
Descripció:	Els desenvolupadors tindran accés al codi de l'aplicació i els usuaris tindran accés a l'aplicació només un cop s'hagin identificat i autentificat amb nom d'usuari i contrasenya.		
Justificació:	Evitar que qualsevol usuari pugui modificar les dades sense cap control d'autenticació.		
Criteri de satisfacció:	Auto-explicatiu		

Requisit #:	10	Tipus de requisit:	Requisit d'integritat
Descripció:	El codi font del l'aplicació client i l'aplicació servidor han de tenir una còpia de seguretat.		
Justificació:	Per provenir qualsevol problema amb el codi font i cal tenir la manera de recuperar.		
Criteri de satisfacció:	Es guardarà el codi en un gestor de versions per tal de que es puguin guardar les versions de codi. Aquest gestor de versions assegurarà que no es perdi el codi ni cap dels canvis que s'hi pugui aplicar.		

8. Especificació

L'especificació de software consisteix en una descripció completa del comportament de l'aplicació que s'ha de desenvolupar. A partir dels requisits definits en el capítol anterior, es descriu el comportament de l'aplicació explicant en detall aquests següents punts en aquest capítol:

- Autors: Autors implicats
- Casos d'ús: Són un conjunt d'esdeveniments realitzats entre l'usuari i l'aplicació per produir un resultat observable i valuós.
- Models conceptuals i model de comportament
- Argumentació satisfacció

8.1. Actors

Els actors que interaccionaran amb l'aplicació són els següents:

Professors: Són els usuaris que interactuen amb l'aplicació per realitzar algun procés. L'aplicació que es desenvolupa va adreçada a un tipus d'usuari específic, concretament els professors de les assignatures de bases de dades de la FIB, que són els usuaris registrats per usar l'aplicació.

Servei web corrector: Sistema extern on s'executen les solucions dels professors per a les qüestions, i les respostes alternatives que es guardaran. Aquests correctors generen les sortides i els *feedbacks* corresponents als jocs de proves i comprovacions.

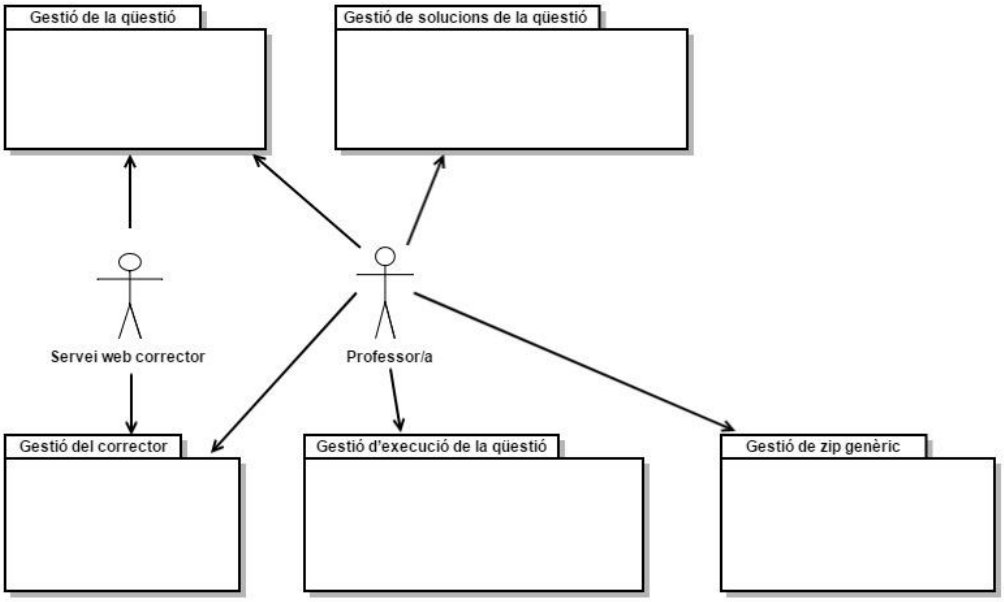
8.2. Diagrama de casos d'ús

Un cas d'ús representa una seqüencia d'esdeveniments que dispara per un actor que haurà de realitzar per dur a terme algun procés mitjançant l'aplicació. Identifiquem cada cas d'ús amb l'identificatiu "CU" per facilitar la referència, cada cas d'ús es correspon amb un requisit funcional que hem definit en el capítol anterior.

A continuació es pot veure el diagrama global del casos d'ús que mostra la relació entre els actors i les diferents agrupacions de casos d'ús. Aquest diagrama de casos d'ús està agrupat per requisits funcionals. Cal tenir en compte de que només l'aplicació client disposarà aquests casos d'ús, ja que és la part que interactua amb els usuaris, en aquest cas, els usuaris són els professors de les assignatures de bases de dades de la FIB.

L'actor professor es qui interactua amb totes les agrupacions, en canvi l'actor servei web corrector interactua únicament amb l'agrupació gestió de qüestions i l'agrupació gestió del corrector, ja que només intervé en l'execució de solució de qüestió.

A continuació, es pot veure una figura que mostra la relació entre els actors i les agrupacions de casos d'ús.

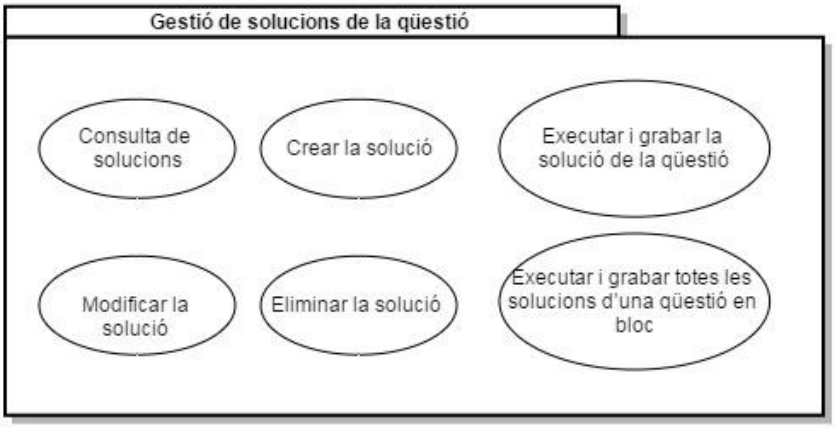


A continuació es mostra per cadascuna de les agrupacions, els casos d'ús que hi inclou.

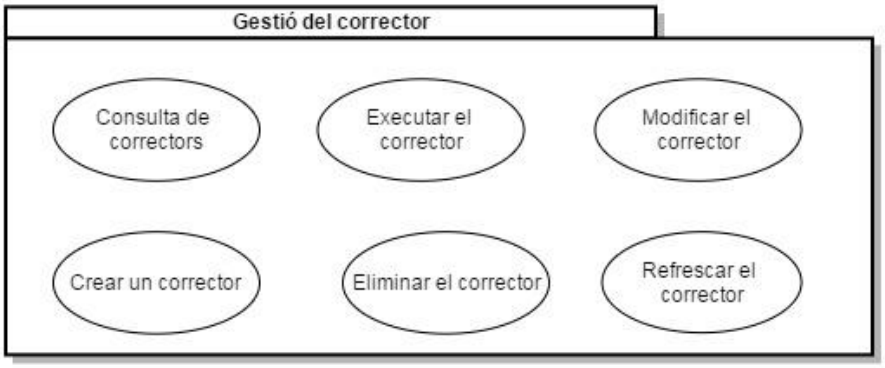
- **Gestió de la qüestió:**
  - CU #01: Modificar el fitxer adjunt
  - CU #02: Eliminar el fitxer adjunt
  - CU #03: Guardar el fitxer adjunt
  - CU #04: Generar el fitxer adjunt
  - CU #05: Generar llistats
  - CU #06: Copiar la qüestió a altres bases de dades (CopyBD)
  - CU #07: Duplicar un joc de proves o una comprovació
  - CU #08: Consultar resultats de la qüestió

- **Gestió d'execució de la qüestió**
  - CU #09: Executar la qüestió
  - CU #10: Canviar l'ordre de joc de proves o comprovació

- **Gestió de solucions de la qüestió**
  - CU #11: Consulta de solucions
  - CU #12: Crear la solució
  - CU #13: Modificar la solució
  - CU #14: Eliminar la solució
  - CU #15: Executar i grabar la solució de la qüestió
  - CU #16: Executar i grabar totes les solucions d'una qüestió en bloc



- **Gestió del corrector**
  - CU #17: Consulta de correctors
  - CU #18: Crear un corrector
  - CU #19: Modificar el corrector
  - CU #20: Eliminar el corrector
  - CU #21: Executar el corrector
  - CU #22: Refrescar el corrector



- **Gestió de zip genèric**
  - CU #23: Modificar el zip genèric

8.3. Descripció dels casos d'ús

La descripció de casos d'ús consisteix en descriure l'escenari d'interaccions que realitza entre l'aplicació i l'usuari per produir un resultat observable i valuós. A continuació per mostrar la descripció dels casos d'ús, es farà servir de la plantilla següent:

Autor principal: *1
Precondició: *2
Disparador: *3
Escenari principal d'èxit: *4
1. ...
2. ...
3. ...

Extensions: *5
1a. ...
1a1. ...
1a2. ...

- ***1:** L'usuari principal que participa en el cas d'ús.
- ***2:** Condició que s'ha de satisfer prèviament en el cas d'ús.
- ***3:** Condició que provoca l'execució del cas d'ús.
- ***4:** Seqüencia d'interaccions realitzats entre l'usuari i sistema del cas d'ús.
- ***5:** Alternatives i tractament en situacions especials.

CU #01: Modificar el fitxer adjunt

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol fer modificar el fitxer.
Escenari principal d'èxit:
1. El professor prem el botó de "seleccionar".
2. S'obre una finestra que conté els fitxers locals per seleccionar.
3. El professor selecciona un fitxer.
4. El professor prem el botó de càrrega manual o prem el botó de càrrega automàtica.
5. El sistema mostra que s'ha fet la modificació amb èxit.

Extensions:
5a. El sistema no s'ha pogut modificar el fitxer.
5a1. El sistema mostra que el fitxer ha de ser amb l'extensió .zip o .class
5a2. Es torna al pas 1.

CU #02: Eliminar el fitxer adjunt

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit, i la qüestió conté un fitxer adjunt.
Disparador: El professor vol eliminar el fitxer adjunt.
Escenari principal d'èxit:

1. El professor prem el botó d'eliminar.
2. El sistema pregunta si realment vol eliminar-ho.
3. El professor confirma que vol eliminar el fitxer.
4. El sistema mostra que el fitxer s'ha eliminat correctament.

Extensions:

3a. El professor cancel·la l'acció d'eliminar el fitxer.
  3a1. El professor prem el botó de cancel·lar.
  3a2. Finalitza el cas d'ús.
4a. El sistema no s'ha pogut eliminar el fitxer.

CU #03: Guardar el fitxer adjunt

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit, i la qüestió conté un fitxer adjunt.
Disparador: El professor vol guardar el fitxer adjunt al local.
Escenari principal d'èxit:

1. El professor prem el botó de descarrega.
2. El sistema es descarrega el fitxer adjunt.

Extensions:

CU #04: Generar el fitxer adjunt

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit, i la qüestió està resolta.
Disparador: El professor vol grabar el fitxer adjunt de la qüestió.
Escenari principal d'èxit:

1. El professor prem el botó de "Generar".
2. El sistema mostra un Pop-up on el professor pugui marcar i desmarcar els *checkbox* per seleccionar el contingut que cal posar en el fitxer adjunt.
3. El professor prem el botó "Veure".
4. El sistema mostra el contingut corresponent de l'adjunt.
5. El professor prem el botó "Guardar".
6. El sistema actualitza el fitxer adjunt de la qüestió amb el contingut que havia configurat el professor.

Extensions:

3a. El professor cancel·la generar el fitxer adjunt.
  3a1. El professor prem el botó de "Cancel·lar".
  3a2. El sistema tanca el Pop-up.
5a. El professor vol tornar a configurar el contingut del fitxer adjunt.
  5a1. El professor prem el botó per tornar al primer Pop-up.
  5a2. Torna al pas 2.

5b. El professor cancel·la generar el fitxer adjunt.
    5b1. El professor prem el botó de “Cancel·lar”.
    5b2. El sistema tanca el Pop-up.

CU #05: Generar llistats

Autor principal: Professor
Precondició: El professor ha fet el logueig de l’aplicació amb èxit.
Disparador: El professor vol generar el fitxer que conté la informació de la qüestió.
Escenari principal d’èxit:

1. El professor prem el botó de generar llistats.
2. El sistema mostra un *pop-up* que conté els *checkbox*.
3. El professor marca els *checkbox* que necessita.
4. El professor prem el botó de “Acceptar”.
5. El sistema es descarrega un fitxer de .txt.

Extensions:

3a. El professor cancel·la l’acció de generar llistats.
    3a1. El professor prem el botó de “Cancel·lar”.
    3a2. El sistema oculta el *pop-up*.
    3a3. Finalitza el cas d’ús.

CU #06: Copiar la qüestió a altra base de dades (CopyBD)

Autor principal: Professor
Precondició: El professor ha fet el logueig de l’aplicació amb èxit.
Disparador: El professor vol copiar la qüestió a altra base de dades.
Escenari principal d’èxit:

1. El professor prem el botó de “Copiar BD”.
2. El sistema mostra un formulari de paràmetres per introduir.
3. El professor introdueix el paràmetres.
4. El professor prem el botó de “Copiar”.
5. El sistema mostra la qüestió s’ha copiat correctament.

Extensions:

4a. El professor cancel·la l’acció de copiar la qüestió a altra base de dades.
    4a1. El professor prem el botó de “Cancel·lar”.
    4a2. El sistema oculta el formulari.
    4a3. Finalitza el cas d’ús.

5a. El sistema no s’ha pogut copiar la qüestió.
    5a1. El sistema notifica que alguns paràmetres introduïts són incorrectes.
    5a2. Es torna al pas 3.

CU #07: Duplicar un joc de proves o una comprovació

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit, i la qüestió té almenys un joc de proves o una comprovació.
Disparador: El professor vol fer la netejar del resultat de la comparació de jocs de proves i comprovacions.
Escenari principal d'èxit:

1. El professor prem el botó de "Duplicar".
2. El sistema duplica totes les dades i totes les seves comprovacions en cas de duplicació d'un joc de proves.

Extensions:

CU #08: Consultar resultats de la qüestió

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol consultar resultats de la qüestió.
Escenari principal d'èxit:

1. El professor prem el botó de "Visualitzar l'historial".
2. El sistema mostra una taula on per cada fila, conté informacions com: nom del curs, grup d'estudiants, nota mitjana sense penalitzacions, nota mitjana amb penalitzacions, número d'equips van contestar, i número d'equips que podien contestar.

Extensions:

2a. El sistema no s'ha pogut mostrar l'historial.
  2a1. L'aplicació avisa al professor que hi hagut l'error a l'hora de consultar l'historial.

CU #09: Executar la qüestió

Autor principal: Professor, servei web corrector
Precondició: El professor ha fet el logueig de l'aplicació amb èxit, i la qüestió està assignada a un o més d'un correctors.
Disparador: El professor vol executar la qüestió.
Escenari principal d'èxit:

1. El professor prem el botó d'executar.
2. El sistema fa la petició d'execució al servei web corrector.
3. El servei web corrector executa la solució de la qüestió, i retorna el resultat al sistema.
4. El sistema mostra que la qüestió s'ha executat.

Extensions:

3a. El servei web corrector no s'ha pogut d'executar la qüestió.
  3a1. El servei web corrector notifica al sistema que s'ha detectat un error.
  3a2. El sistema mostra per la pantalla l'error obtingut.

CU #10: Canviar l'ordre de joc de proves o comprovació

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit, i la qüestió té almenys dos jocs de proves.
Disparador: El professor vol canviar l'ordre de jocs de proves o canviar l'ordre de comprovació dins d'un joc de proves.
Escenari principal d'èxit:

1. El professor prem el botó de "Baixar" o "Pujar".
2. El sistema canvia l'ordre de joc de proves o comprovació.

Extensions:

CU #11: Consulta de solucions

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol consultar la solució.
Escenari principal d'èxit:

1. El professor selecciona la solució que vol consultar.
2. El sistema carrega les dades guardes i mostra la solució.

Extensions:

CU #12: Crear la solució

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol crear una solució típiques d'estudiant.
Escenari principal d'èxit:

1. El professor prem el botó de "Crear".
2. El sistema mostra un formulari perquè el professor pugui introduir el nom de la solució i el text de solució.
3. El professor prem el botó de "Desar".
4. El sistema notifica que s'ha creat la solució amb èxit.

Extensions:

3a. El professor decideix cancel·lar la creació de la solució.
  3a1. El professor prem el botó de "Cancel·lar".
  3a2. El sistema retorna a l'estat inicial.
  3a3. Finalitza el cas d'ús.

4a. No s'ha pogut crear la solució.
  4a1. El sistema notifica que s'ha detectat un error a l'hora de crear la solució.

CU #13: Modificar la solució

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol modificar la solució
Escenari principal d'èxit:

1. El professor modifica el nom de la solució, i/o el text de solució.
2. El sistema mostra el botó de “Desar” i el botó de “Cancel·lar” al mateix temps.
3. El professor prem el botó de “Desar”.
4. El sistema notifica que s’ha modificat la solució amb èxit.

Extensions:

3a. El professor decideix cancel·lar la modificació de la solució.
  3a1. El professor prem el botó de “Cancel·lar”.
  3a2. El sistema retorna a l’estat inicial.
  3a3. Finalitza el cas d’ús.

4a. No s’ha pogut modificat la solució.
  4a1. El sistema notifica que s’ha detectat un error a l’hora de guardar les dades modificades de la solució.

CU #14: Eliminar la solució

Autor principal: Professor
Precondició: El professor ha fet el logueig de l’aplicació amb èxit.
Disparador: El professor vol eliminar la solució
Escenari principal d’èxit:

1. El professor prem el botó de “Eliminar”.
2. El sistema pregunta si realment vol eliminar-ho.
3. El professor prem el botó de “Acceptar”.
4. El sistema notifica que s’ha eliminat la solució amb èxit.

Extensions:

3a. El professor decideix cancel·lar l’acció d’eliminació.
  3a1. El professor prem el botó de “Cancel·lar”.
  3a2. El sistema retorna a l’estat inicial.
  3a3. Finalitza el cas d’ús.

4a. No s’ha pogut eliminat la solució.
  4a1. El sistema notifica que s’ha detectat un error a l’hora d’eliminar la solució.

CU #15: Executar i grabar la solució de la qüestió

Autor principal: Professor, servei web corrector
Precondició: El professor ha fet el logueig de l’aplicació amb èxit.
Disparador: El professor vol executar una solució o més d’una solució.
Escenari principal d’èxit:

1. El professor prem el botó de “Executar”.
2. El sistema passa les dades de solucions al servei web corrector.
3. El servei web corrector s’executa la solució o les solucions i retorna el resultat al sistema.
4. El sistema mostra per la pantalla el resultat obtingut.

Extensions:

3a. El servei web corrector no s’ha pogut d’executar les solucions.
  3a1. El servei web corrector notifica al sistema que s’ha detectat un error.
  3a2. El sistema mostra per la pantalla l’error obtingut.

CU #16: Executar i grabar totes les solucions d'una qüestió en bloc

Autor principal: Professor

Precondició: El professor ha fet el logueig de l'aplicació amb èxit, i almenys hi ha una solució en la qüestió.
Disparador: El professor vol executar totes les solucions i guardar els resultats de la comparació de jocs de proves i comprovacions a la base de dades.
Escenari principal d'èxit:

1. El professor prem el botó de "Executar i guardar en bloc".
2. El sistema passa les dades de solucions al servei web corrector.
3. El servei web corrector s'executa totes les solucions i es guarda els resultats a la base de dades.
4. El sistema mostra per la pantalla els resultats obtingut.

Extensions:

3a. El servei web corrector no s'ha pogut d'executar les solucions.
   3a1. El servei web corrector notifica al sistema que s'ha detectat un error.
   3a2. El sistema mostra per la pantalla l'error obtingut.

CU #17: Consulta de correctors

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol consultar el corrector.
Escenari principal d'èxit:

1. El professor selecciona el corrector que vol consultar.
2. El sistema carrega les dades guardes i mostra el corrector.

Extensions:

CU #18: Crear un corrector

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol crear un nou corrector.
Escenari principal d'èxit:

1. El professor prem el botó de "Crear".
2. El sistema mostra un formulari perquè el professor pugui introduir url, descripció del nou corrector.
3. El professor prem el botó de "Desar".
4. El sistema notifica que s'ha creat el corrector amb èxit.

Extensions:

3a. El professor decideix cancel·lar la creació del corrector.
   3a1. El professor prem el botó de "Cancel·lar".
   3a2. El sistema retorna a l'estat inicial.
   3a3. Finalitza el cas d'ús.
4a. No s'ha pogut crear el corrector.

4a1. El sistema notifica que s'ha detectat un error a l'hora de crear el corrector.

CU #19: Modificar el corrector

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol modificar el corrector existent.
Escenari principal d'èxit:

1. El professor modifica url, descripció, assignar/desassignar preguntes existents.
2. El sistema mostra el botó de "Desar" i el botó de "Cancel·lar" al mateix temps.
3. El professor prem el botó de "Desar".
4. El sistema notifica que s'ha modificat el corrector amb èxit.

Extensions:

3a. El professor decideix cancel·lar la modificació del corrector.
  3a1. El professor prem el botó de "Cancel·lar".
  3a2. El sistema retorna a l'estat inicial.
  3a3. Finalitza el cas d'ús.

4a. El sistema no s'ha pogut modificat el corrector.
  4a1. El sistema notifica que s'ha detectat un error a l'hora de guardar les dades modificades del corrector.

CU #20: Eliminar el corrector

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol eliminar el corrector.
Escenari principal d'èxit:

1. El professor prem el botó de "Eliminar".
2. El sistema pregunta si realment vol eliminar-ho.
3. El professor prem el botó de "Acceptar".
4. El sistema notifica que s'ha eliminat el corrector amb èxit.

Extensions:

3a. El professor decideix cancel·lar l'acció d'eliminació.
  3a1. El professor prem el botó de "Cancel·lar".
  3a2. El sistema retorna a l'estat inicial.
  3a3. Finalitza el cas d'ús.

4a. No s'ha pogut eliminat el corrector.
  4a1. El sistema notifica que s'ha detectat un error a l'hora d'eliminar el corrector.

CU #21: Executar el corrector

Autor principal: Professor, servei web corrector
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol executar el corrector, el corrector té almenys una qüestió assignada.
Escenari principal d'èxit:

1. El professor prem el botó de "Executar".

2. El sistema passa les dades al servei web corrector.
3. El servei web corrector executa totes les qüestions assignades.

Extensions:

3a. El servei web corrector no s'ha pogut d'executar alguna de les qüestions assignades.
   3a1. El servei web corrector notifica al sistema que s'ha detectat un error.
   3a2. El sistema mostra per la pantalla l'error obtingut.

CU #22: Refrescar el corrector

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol refrescar el corrector per obtenir el corrector amb les dades actualitzades.
Escenari principal d'èxit:

1. El professor prem el botó de "Refrescar".
2. El sistema mostra el corrector amb les dades que venen de la base dades.

Extensions:

CU #23: Modificar el zip genèric

Autor principal: Professor
Precondició: El professor ha fet el logueig de l'aplicació amb èxit.
Disparador: El professor vol fer modificar el fitxer.
Escenari principal d'èxit:

1. El professor prem el botó de "seleccionar".
2. S'obre una finestra que conté els fitxers locals per seleccionar.
3. El professor selecciona un fitxer.
4. El professor prem el botó de "Pujar",
5. El sistema mostra que s'ha fet la modificació del zip genèric amb èxit.
6. El sistema mostra un llistat dels arxius del zip genèric.

Extensions:

5a. El sistema no s'ha pogut modificar el fitxer.
   5a1. L'aplicació mostra que el fitxer ha de ser amb l'extensió .zip.
   5a2. Es torna al pas 1.

8.4. Arguments de satisfacció

Els arguments de satisfacció permeten verificar per cada objectiu establert, quines són les premisses que potser un requisit funcional o un cas d'ús per garantir aquest objectiu es compliran.

La descripció del requisits es farà servir la plantilla següent:

Objectiu	*1
Premisses	*2
Justificació	*3

- ***1:** Descripció d'objectiu establert
- ***2:** Conjunt de requisits per justificar l'objectiu
- ***3:** Justificació de l'argument mitjançant el conjunt dels requisits

OB #1: Desenvolupament de correctors

Objectiu	L'aplicació proporcionarà un llistat de tots els correctors existent del sistema, i per cada corrector es mostra la informació detallada com URL, descripció, categories, l'estat del corrector, i les qüestions associades, i proporcionarà la gestió de correctors, és a dir, l'alta, baixa, modificació i execució de les qüestions assignades del corrector.
Premisses	P1 (Requisit funcional) CU #17: Consulta de correctors P2 (Requisit funcional) CU #18: Crear un corrector P3 (Requisit funcional) CU #19: Modificar el corrector P4 (Requisit funcional) CU #20: Eliminar el corrector P5 (Requisit funcional) CU #21: Executar el corrector P6 (Requisit funcional) CU #22: Refrescar el corrector
Justificació	Com que els usuaris podran visualitzar la informació detallada d'un corrector i un llistat de tots els correctors, i un cop refrescat el corrector també es pot visualitzar la informació, per tant s'assoleix aquest objectiu. I podran fer la creació, modificació, eliminació, i execució d'un corrector, tot això forma part de la gestió de correctors, i per tant s'assoleix aquest objectiu.

OB #2: Generació del fitxer adjunt

Objectiu	L'aplicació permetrà fer la generació automàtica del fitxer adjunt de les qüestions, disposa un Pop-up per seleccionar el contingut, l'usuari ha de marcar quines parts es volen incloure al fitxer i quines parts s'han d'obviar. Si la qüestió és de tipus JDBC, es genera el fitxer adjunt en funció del contingut del zip genèric (OB# 7). Permetrà també grabar a les dades del fitxer adjunt de la qüestió.
Premisses	P1 (Requisit funcional) CU #01: Modificar el fitxer adjunt P2 (Requisit funcional) CU #02: Eliminar el fitxer adjunt P3 (Requisit funcional) CU #03: Guardar el fitxer adjunt P4 (Requisit funcional) CU #04: Generar el fitxer adjunt
Justificació	Com que els usuaris podran fer modificació, eliminació, descarrega del fitxer, tot això forma part de la gestió del fitxer adjunt de la qüestió, per tant s'assoleix aquest objectiu.

OB #3: Desenvolupar la còpia de qüestions a altres bases de dades

Objectiu	L'aplicació permetrà mostra un formulari de connexió, i copiar la qüestió a altres bases de dades.
Premisses	P1 (Requisit funcional) CU #06: Copiar la qüestió a altres bases de dades (CopyBD)
Justificació	Els usuaris podran introduir els paràmetres com el tipus de SGBD, el nom de servidor, el host, l'esquema de bases de dades, el número de la port, si és xifrat SSL o no, etc... i connectar a altra base de dades per copiar la qüestió a aquesta, per tant s'assoleix aquest objectiu.

OB #4: Desenvolupar la pestanya d'execució de qüestions

Objectiu	L'aplicació proporcionar un panell que permetrà mostrar els correctors assignats, modificar els correctors assignats, permetrà executar la qüestió per un o més serveis web, i mostrar les solucions de jocs de proves comprovacions, log d'execució un cop executat, permetrà modificar l'ordre de jocs de proves, permetrà netejar log d'execucions.
Premisses	P1 (Requisit funcional) CU #09: Executar la qüestió P2 (Requisit funcional) CU #10: Canviar l'ordre de joc de proves o comprovació
Justificació	Com que els usuaris podran fer l'execució de qüestió i canviar l'ordre de joc de proves i comprovació per posteriorment poder reexecutar la mateixa qüestió amb aquest ordre, per tant s'assoleix aquest objectiu.

OB #5: Desenvolupar la pestanya d'execució de solucions d'estudiants

Objectiu	L'aplicació permetrà fer la gestió de solucions típiques d'alumnes, fer l'alta, baixa, modificació, i execució de la solució, i donar opció d'executar varies solucions en bloc, mostrar log d'execució, també permetrà netejar el resultat de la comparació de solució.
Premisses	P1 (Requisit funcional) CU #11: Consulta de solucions P2 (Requisit funcional) CU #12: Crear la solució P3 (Requisit funcional) CU #13: Modificar la solució P4 (Requisit funcional) CU #14: Eliminar la solució P5 (Requisit funcional) CU #15: Executar i grabar la solució de la qüestió P6 (Requisit funcional) CU #16: Executar i grabar totes les solucions d'una qüestió en bloc
Justificació	Com que els usuaris podran consultar la solució d'una qüestió, i fer creació, modificació, eliminació, executar d'una solució o diverses solucions en bloc, fer gestió del resultat d'execució, totes aquestes funcionalitats forma part de la gestió de la solució, per tant s'assoleix l'objectiu.

OB #6: Gestió de zip genèric

Objectiu	L'aplicació permetrà seleccionar el fitxer local, i carregar-ho per configurar el zip genèric del sistema que es fa servir per la generació automàtica del fitxer adjunt, també permetrà visualitzar un llistat del arxius del zip genèric.
Premisses	P1 (Requisit funcional) CU #23: Modificar el zip genèric
Justificació	Com que els usuaris podran seleccionar un fitxer en extensió .zip des de local i carrega-ho per configurar el zip genèric del sistema, per tant s'assoleix l'objectiu.

OB #7: Consultar resultats

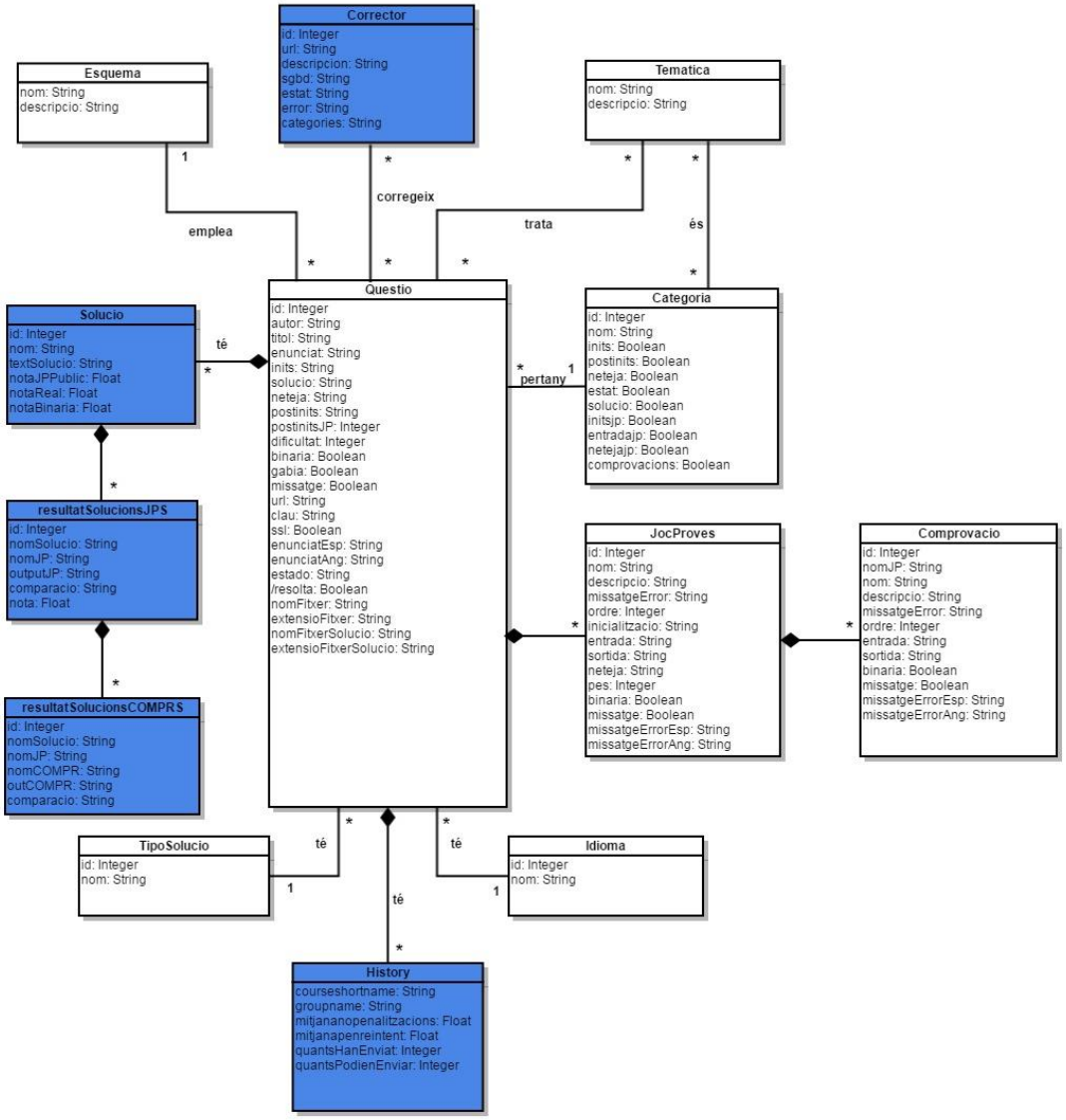
Objectiu	L'aplicació permetrà consultar informació d'una qüestió. La primera mostra una taula on indica per cada possible resposta d'una qüestió, els seus jocs de proves i comprovacions si passa o no. La segona mostra una taula on indica per cada grup d'estudiants que van fer el qüestionari, les notes mitjanes que varen treure els estudiants i la quantitat d'estudiants que van donar una resposta del total del curs.
Premisses	P1 (Requisit funcional) CU #08: Consultar resultats de la qüestió
Justificació	Com que els usuaris podran visualitzar informació dels resultats de la qüestió, per tant s'assoleix l'objectiu.

8.5. Model conceptual

El model conceptual es composa de diferents classes, d'objectes amb els seus atributs i associacions. També inclou les restriccions d'integritat que han de satisfer. En aquest capítol explicarem l'esquema conceptual complementat amb les restriccions textuals d'integritat que s'identifiquin.

8.5.1. Esquema conceptual

A continuació es mostra l'esquema conceptual. En aquest esquema es marca en blau les classes que ha estat necessari afegir en l'ATW2 respecte a l'ATW. Les associacions entre aquestes classes i les existents en la versió de l'aplicació de partida també s'ha hagut de tenir-les en compte en el nou desenvolupament. També s'indiquen les restriccions que cal tenir en compte en el desenvolupament de l'aplicació.



Il·lustració 7: Esquema conceptual

8.5.2. Restriccions d'integritat

Les restriccions d'integritat textuals són aquelles que no es poden representar gràficament dins l'esquema conceptual. A continuació podem veure el llistat de restriccions d'integritat textuals. Identifiquem cada restriccions d'integritat amb l'identificador "RI" per facilitar la referència més endavant.

- **RI #01:** No pot haver dos qüestions amb el mateix identificador.
- **RI #02:** No pot haver dos esquemes amb el mateix nom.
- **RI #03:** No pot haver dos temàtiques amb el mateix nom.
- **RI #04:** No pot haver dos categories amb el mateix identificador.
- **RI #05:** No pot haver dos solucions amb el mateix identificador i el mateix nom.
- **RI #06:** No pot haver dos resultatSolucionsJPS amb el mateix identificador, el mateix nom de la solució i el mateix nom de joc de proves.
- **RI #07:** No pot haver dos resultatSolucionsCOMPRS amb el mateix identificador, el mateix nom de la solució, el mateix nom de joc de proves, i el mateix nom de comprovació.
- **RI #08:** No pot haver dos jocs de proves amb el mateix nom assignats a la mateixa qüestió.
- **RI #09:** No pot haver dos comprovacions amb el mateix nom assignats al mateix joc de proves.
- **RI #10:** No pot haver dos correctors amb la mateixa URL.
- **RI #11:** No pot haver dos tipus de solució amb el mateix identificador.
- **RI #12:** No pot haver dos idiomes amb el mateix identificador.
- **RI #13:** La dificultat de les qüestions només pot prendre valors entre 0 i 1.
- **RI #14:** El pes dels jocs de proves només pot prendre valors entre 0 i 1.
- **RI #15:** La suma dels pesos dels jocs de proves assignats a una qüestió ha de sumar 1.
- **RI #16:** Extensió del fitxer de la qüestió només pot prendre valors "txt" o "sql" o "zip".
- **RI #17:** L'atribut derivat "/resolt" s'obté de la següent manera:
  - Una qüestió es considera com resolta sempre quan tingui jocs de proves i cap d'aquests ni tampoc cap de les seves comprovacions, disposi d'una sortida buida.
  - En qualsevol altre cas la qüestió es considera com no resolta.
- **RI #18:** L'error dels correctors només pot prendre valor entre 1 i 4.

- **RI #19:** La nota públic, la nota real, i la nota binaria de la solució només poden prendre valor entre 0.0 i 1.0.

- **RI #20:** La nota públic, la nota real, i la nota binaria del resultatSolucionsJPS només poden prendre valor entre 0.0 i 1.0.

- **RI #21:** La nota públic, la nota real, i la nota binaria del resultatSolucionsCOMPRS només poden prendre valor entre 0.0 i 1.0.

- **RI #24:** La nota mitjana sense i amb penalitzacions només poden prendre valor entre 0.0 i 1.0.

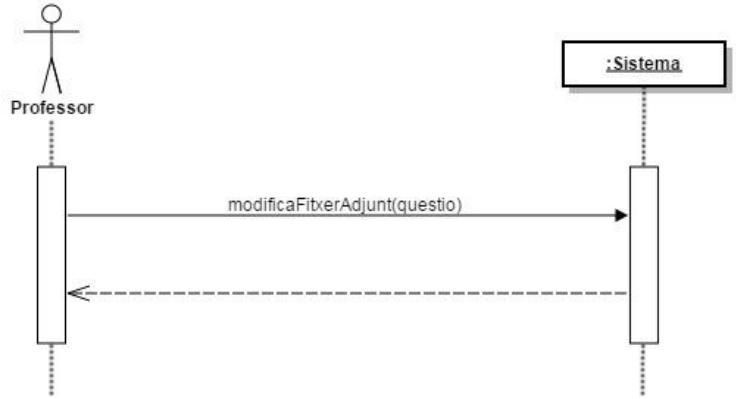
8.6. Model de comportament

El model de comportament defineix les operacions que es duen a terme per realitzar els casos d'ús identificats en el capítol anterior.

8.6.1. Diagrames de seqüencia

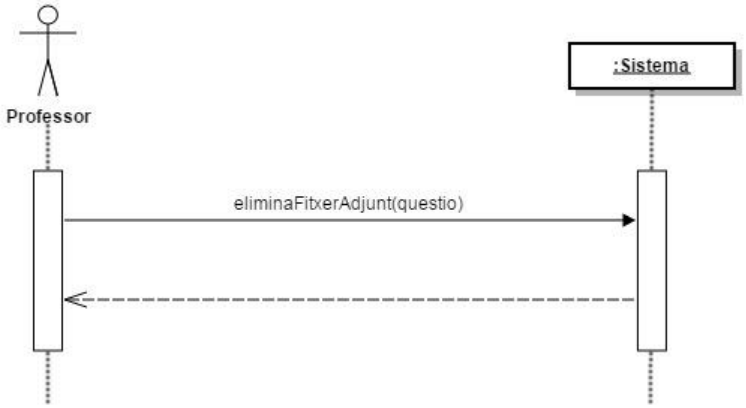
El digrama de seqüencia mostra la interacció entre els actors i el sistema (aplicació que es desenvolupa). Per cada cas d'ús es mostra l'escenari particular del cas d'ús que conté els actors principals, i les operacions que es criden entre ells. A continuació, es mostren els casos d'ús més destacats.

CU #01: Modificar el fitxer adjunt



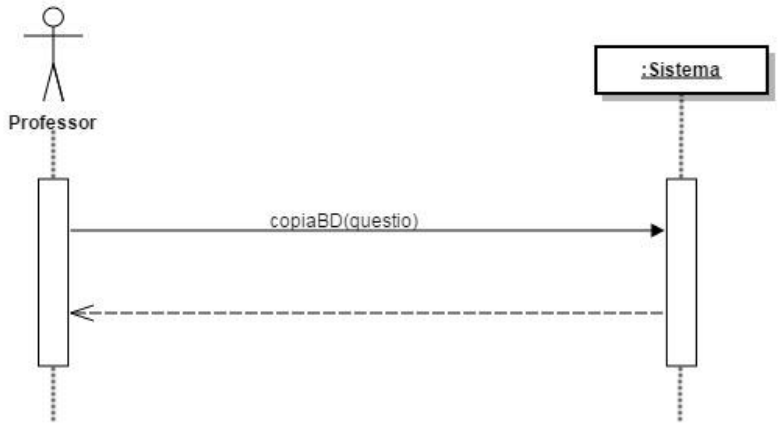
Context Sistema::modificaFitxerAdjunt (questio: Questio)
Pre La questio no és buida.
Post El sistema actualitza el fitxer adjunt antic de la qüestió pel nou.

CU #02: Eliminar el fitxer adjunt



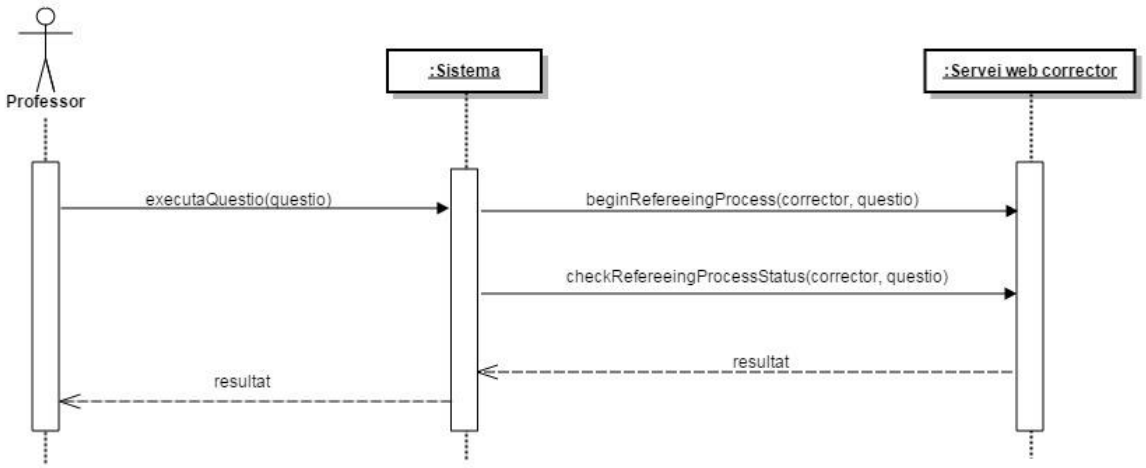
Context Sistema::eliminaFitxerAdjunt (questio: Questio)
Pre La questio no és buida.
La questio té valor en l'atribut fitxeradjunt
Post El sistema elimina el fitxer adjunt de la qüestió.

CU #06: Copiar la qüestió a altres bases de dades (CopyBD)



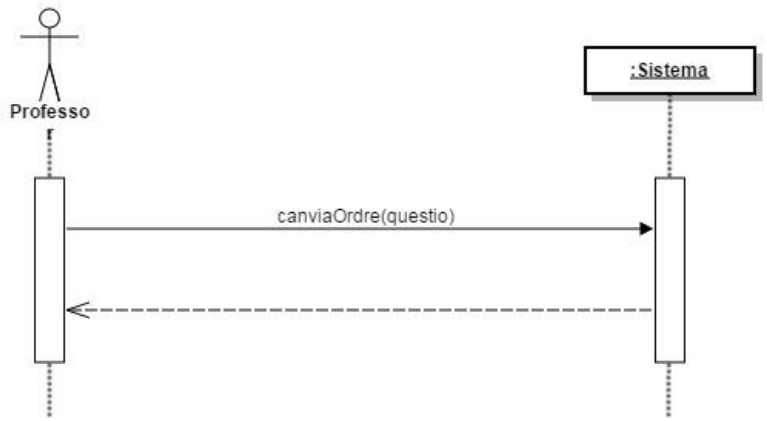
Context Sistema::copiaBD (questio: Questio)
Pre La questio no és buida.
Post El sistema estableix connexió segons el paràmetres de base de dades, i copia la qüestió a aquesta base de dades.

CU #09: Executar la qüestió



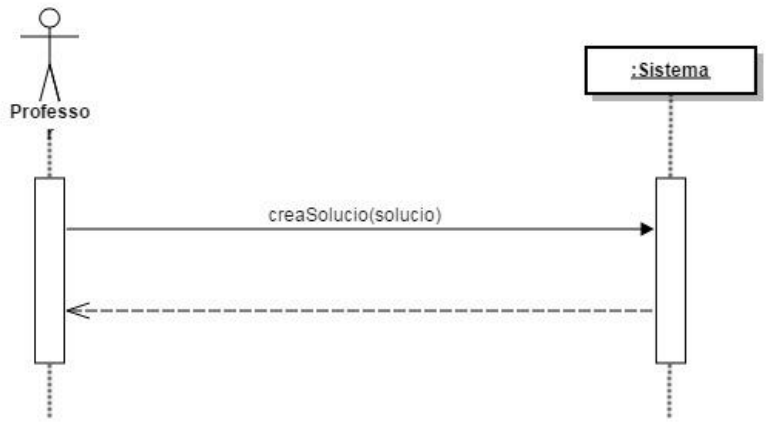
Context Sistema::executaQuestio (questio: Questio)
Pre La questio no és buida amb jocs de proves, i la qüestió té assignada almenys a un corrector.
Post El servei web corrector retorna el resultat al sistema, i el sistema mostra el resultat per la pantalla.

CU #10: Canviar l'ordre de joc de proves o comprovació



Context Sistema:: canviaOrdre (questio: Questio)
Pre La questio no és buida amb jocs de proves.
Post El sistema canvia el ordre del joc de proves o comprovació.

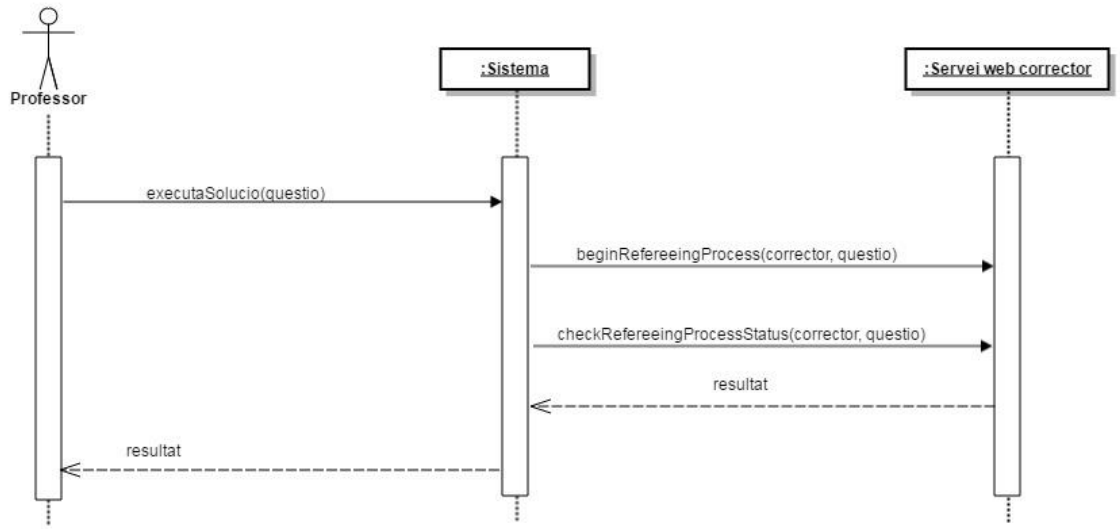
CU #12: Crear la solució



Context Sistema:: creaSolucio (solució: Solucio)
Pre La solució no és buida.

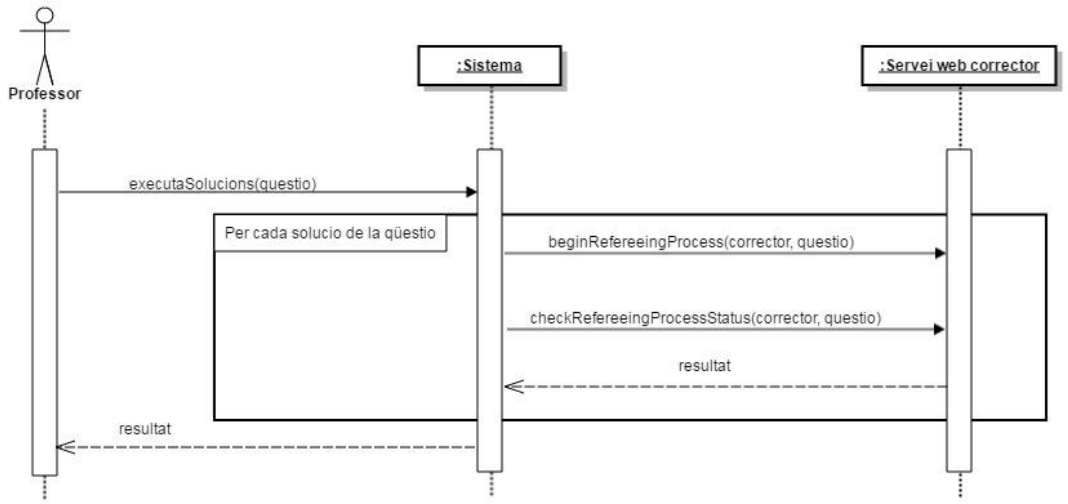
Post El sistema es dona l'alta d'una instancia de la nova solució amb el nom, i el text de solució.

CU #15: Executar i grabar la solució de la qüestió



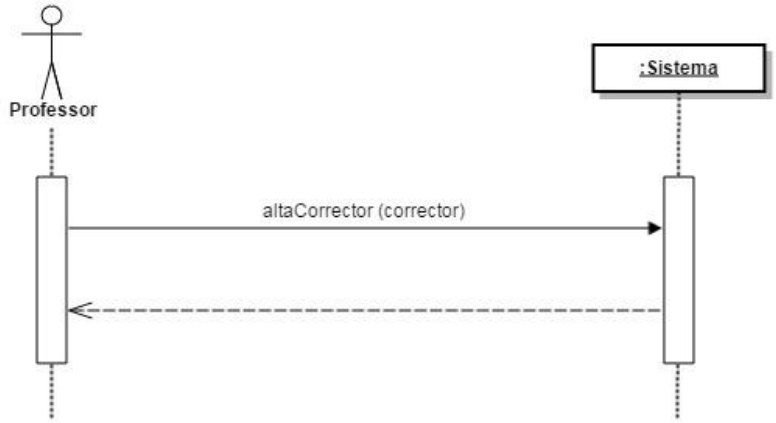
Context Sistema:: executaSolucio (questio: Questio)
Pre La qüestió té almenys una possible solució.
Post El servei web corrector executa la solució, i retorna el resultat al sistema, i el sistema mostra per la pantalla el resultat de la comparació.

CU #16: Executar i grabar totes les solucions d'una qüestió en bloc



Context Sistema:: guardaResultatsJPCOMPR (questio: Questio)
Pre La qüestió té almenys una possible solució.
Post Per cada solució de la qüestió, el servei web corrector ho executa, i retorna el resultat al sistema, i el sistema mostra per la pantalla el resultat de la comparació.

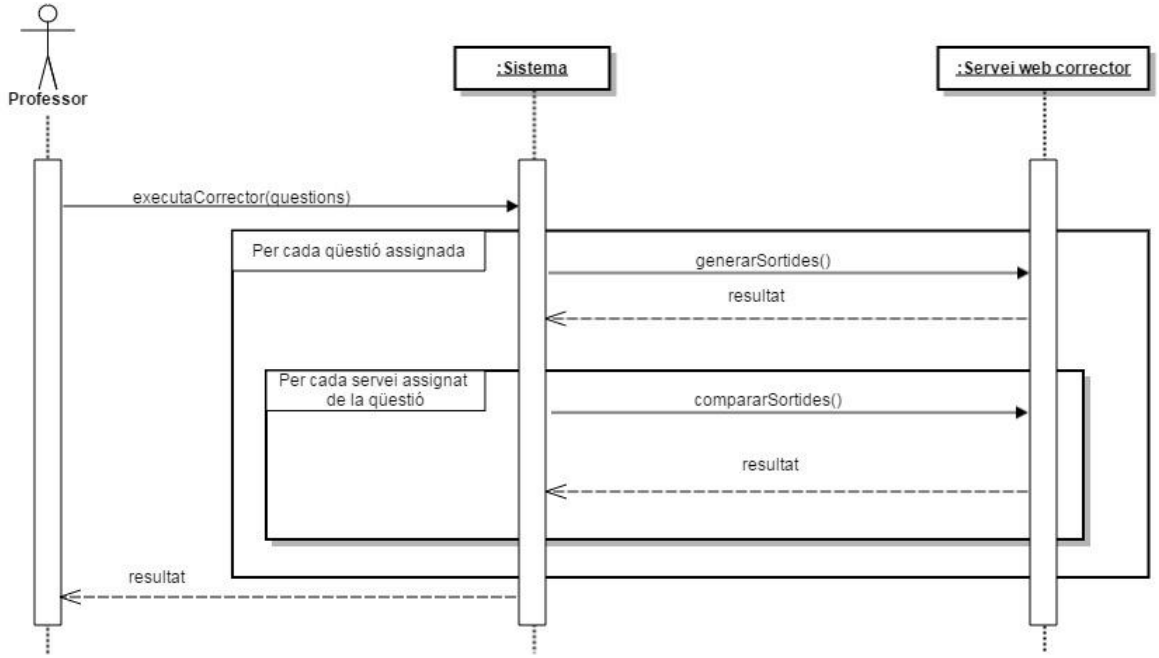
CU #18: Crear un corrector



Context Sistema:: altaCorrector (corrector:Corrector)
Pre L'URL de corrector no és buit.

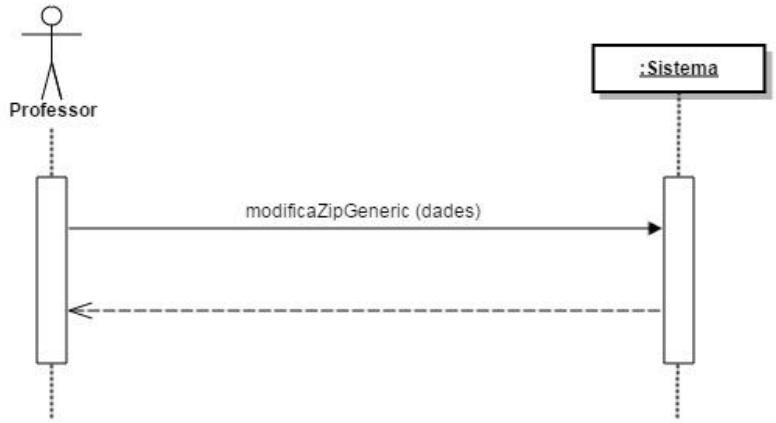
Post El sistema es elimina el text de solució, el resultat de la comparació de jocs de proves de la solució en la base de dades.

CU #21: Executar el corrector



Context Sistema:: executaCorrector(corrector:Corrector)
Pre El corrector té qüestions assignades, i l'estat del corrector és accessible.
Post El servei web corrector genera les sortides de les qüestions que estan assignades a un corrector, retorna correcte (per cada qüestió) que ha generat solucions correctes i error en cas contrari, a més retorna (en cas de haver-hi més d'un corrector assignat) correcte si les sortides igual, i error en cas contrari.

CU #23: Modificar el zip genèric

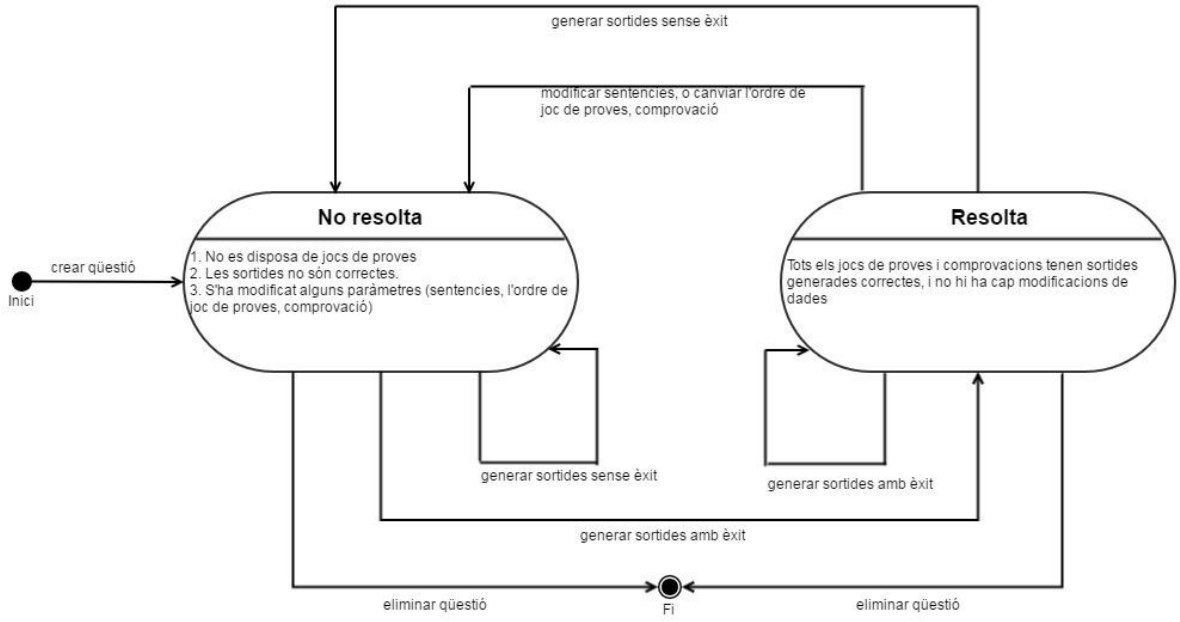


Context Sistema:: modificaZipGeneric (dades)
Pre Les dades de zip genèric no és buida.
Post El sistema actualitza les dades de zip genèric del sistema.

8.7. Model d'estats

Un model d'estats representa els diferents estats que poden estar els atributs d'una classe. En aquest cas, l'atribut "resolta" de la classe qüestió es comporta de manera diferent segons les acciona que es facin, es pot estar resolta, o sense resoldre.

A continuació podem veure un flux del diagrama d'estats, com es pot observar una qüestió estarà resolta sempre quan s'hagi generat les sortides de jocs de proves i comprovacions, i els resultats siguin correctes. Mentre que, qualsevol canvi que faci sobre les sentencies, l'ordre de jocs de proves, comprovacions de la qüestió etc... Tot això fa que la qüestió estigui en l'estat no resolta.



Il·lustració 8: Model d'estats

9. Disseny

El disseny d'un sistema representa com està construït internament per resoldre les funcionalitats que se li demanen. En aquest capítol es detallarà el disseny de l'aplicació que es desenvolupa. En concret es mostra el disseny de l'aplicació servidor, i el disseny de l'aplicació client en que es descomposa.

9.1. Disseny del sistema

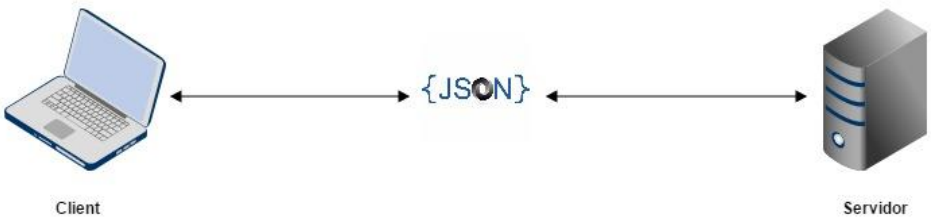
L'ATW2 segueix el mateix tipus de disseny de l'aplicació ATW, és a dir, *Single Page Application.* Aquest disseny ofereix una experiència d'usuari més fluida, i usable que altres dissenys alternatius per aplicacions web. L'aplicació global es divideixen en dues aplicacions que comuniquen entre si:

- **Aplicació servidor:** S'encarrega de la lògica de negoci del sistema, disposa de rebre peticions, i les dades que venen de l'aplicació client, processar-les, i tornar a enviar processades a l'aplicació client, també s'encarrega d'emmagatzemar les dades a la base de dades, i de comunicar-se amb els serveis web correctors.
- **Aplicació client:** S'encarrega de mostrar la informació a l'usuari, d'interactuar amb l'usuari, i de fer peticions quan sigui necessari a l'aplicació servidor.

La idea és que en la primera petició que es fa de l'aplicació client a l'aplicació servidor, es carrega pràcticament totes les dades necessàries al client. Així doncs, l'usuari podrà treballar amb aquestes dades a nivell de client sense demanar recursos a l'aplicació servidor. Només es fa peticions a l'aplicació servidor quan hagi d'actualitzar les dades, accedir a dades puntuals no obtingudes en la primera petició, o fer peticions als serveis web correctors. D'aquesta manera, el client actualitza les dades de la interfície selectivament sense necessitat de perdre la resta.

La manera de comunicar entre el client i el servidor s'ha mantingut, és a dir, utilitza el forma d'intercanvi JSON (Javascript Object Notation), ja que té avantatge de la lleugeresa, la facilitat de maneig i la compatibilitat.

A continuació podem observar la figura d'arquitectura del sistema:



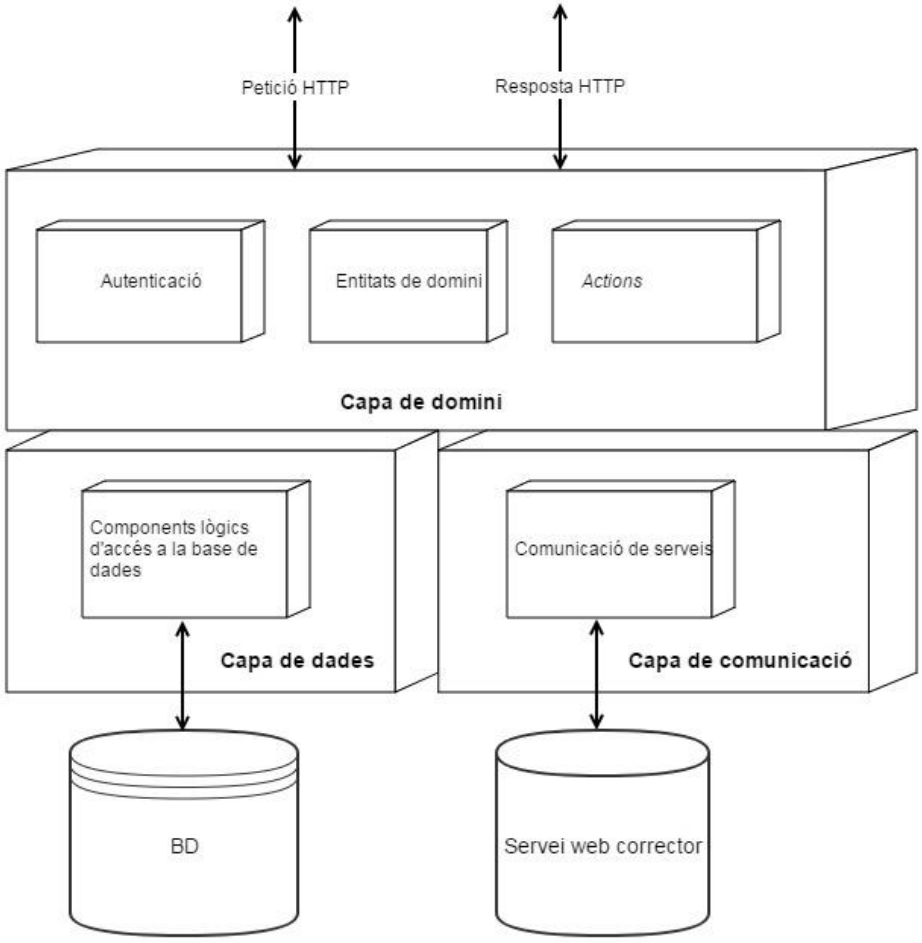
Il·lustració 9: Arquitectura del sistema

9.2. Disseny de l'aplicació servidor

En aquest apartat es detallarà el disseny de l'aplicació servidor, des d'un punt de vista d'arquitectura: el patró usat, peticions de *HTTP*, la autenticació del sistema, la comunicació als servei web, i accés a las base de dades. S'ha treballat sobre el disseny, l'estructura de paquets, i el codi de l'aplicació existent.

9.2.1. Arquitectura

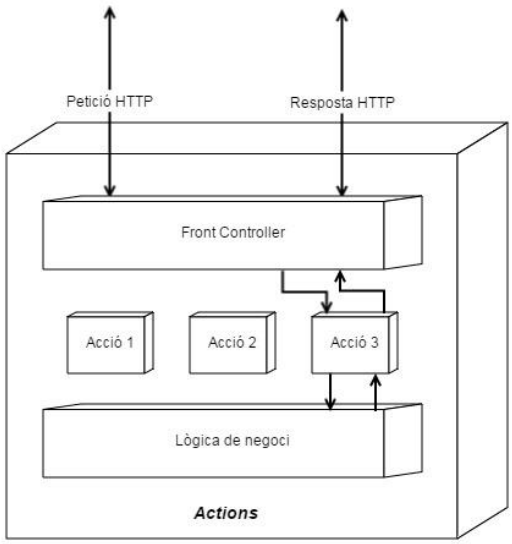
L'aplicació servidor es descompon en tres capes. La capa de domini és la que respon a peticions en forma de servei web, i s'encarrega d'implementar tota la lògica de negoci. La capa de dades és la que s'encarrega de consultar, emmagatzemar, i actualitzar la base de dades relacional. La capa de comunicació és la que comunica amb el servei web corrector. La següent figura il·lustra aquesta arquitectura.



Il·lustració 10: Disseny de l'aplicació servidor

9.2.2. *Patró Front-Controller*

En el desenvolupament de l'aplicació servidor s'usa el patró de disseny *Front-Controller,* la idea d'aquest patró és bàsicament imposar un únic punt d'entrada per a totes les peticions que rep l'aplicació, d'aquesta manera, permet unificar totes les peticions, i per cada petició pot delegar-la a l'Acció corresponent.



Il·lustració 11: Patró Front-Controller

9.2.3. REST API

La REST API abreviatura de *Representational State Transfer* és un estil d'arquitectura per dissenyar aplicacions en xarxa. És molt més efectiva gràcies a HTTP (*Hypertext Transfer Protocol*) [25]. El motiu de que això sigui així és que aquest protocol permet compartir informació entre un client (portàtil, telèfon mòbil, tauleta, etc.) i un servidor. Les característiques d'aquesta arquitectura són següents:

- **Client / Servidor:** La separació entre el client i el servidor ajuda a millor la portabilitat de la interfície a l'altre tipus de plataforma.
- **Sense estat:** S'utilitza HTTP com protocol sense estat de peticions entre client i servidor. Cada petició HTTP és totalment independent, això vol dir que cada petició ha de contenir tota la informació necessària per ser realitzada.
- **Interfície uniforme:** Utilitza els mètodes que indiquen accions que es faran sobre un recurs, els mètodes més habituals per la REST API són:
    - **GET:** Per consultar, llegir i en definitiva accedir a un recurs.
    - **POST:** Per enviar dades per crear un recurs. Com en qualsevol petició POST, les dades no s'envien a la URI, sinó que han d'anar inclosos en el cos de la petició.
    - **PUT:** Per modificar un recurs. Igual que el POST, les dades han d'anar en el cos de la petició.
    - **DELETE:** Per eliminar un recurs.
    - **PATCH:** Per modificar parcialment un recurs.

A continuació podem observar la taula de punt d'entrada que conté les noves URLs afegides en l'ATW2 (en l'apartat 6 s'ha llistat les que hi havia previament en l'ATW).

URL	Mètode *HTTP*	Petició	Resposta	Descripció
/ws/questions/{id}/copyBD	POST	Dades de la qüestió (*JSON)*	JSON	Copia la qüestió {id} a altra base de dades.
/ws/questions/{id}/execute_question	POST	Dades de la qüestió (*JSON)*	JSON	Executa qüestió {id}
/ws/questions/{id}/compare_question	POST	Dades de la qüestió (*JSON)*	JSON	Executa les solucions de la qüestió {id}
/ws/questions/{id}/commit_question	POST	Dades de la qüestió (*JSON)*	JSON	Guarda el resultat de l'execució de la solució de la qüestió {id}
/ws/questions/{id}/clear_question	POST	Dades de la qüestió (*JSON)*	JSON	Elimina el resultat de l'execució de la solució de la qüestió {id}
/correctors*	GET		HTML	Retorna tots els correctors
/ws/correctors	GET		JSON	Retorna un JSON que conté el llistat de tots els correctors.
/ws/correctors	POST	Noves dades del corrector	JSON	Crea un nou corrector i el torna.
/ws/correctors/{id}	PUT	Noves dades del corrector	JSON	Modifica el corrector {id} i retorna dades actualitzada del corrector.
/ws/correctors/{id}	DELETE		JSON	Esborra el corrector {id}
/ws/correctors/{id}/execute	POST		JSON	Executa les qüestions assignades del corrector {id}
/ws/simplequestions	GET		JSON	Retorna un JSON que conté el llistat de totes les qüestions simples
/ws/simplecorrectors	GET		JSON	Retorna un JSON que conté el llistat de totes les qüestions simples
/ws/configuration	POST	Noves dades del zip genèric	JSON	Retorna el zip genèric actualitzat.

Taula 11: Taula de punt d'entrada

9.2.4. *Actions*

Les *Actions* són les classes que s'encarreguen de rebre les peticions *HTTPs* que arriben a l'aplicació servidor, i en funció de la URL de la petició, preparar les dades necessàries per poder executar la petició.

Les *Actions* estan agrupades segons la necessitat d'autenticació. La configuració de com estan agrupades les *Actions* es pot trobar al fitxer struts.xml, i per tant és fàcilment modificable. Els grups que hi ha definits són els que s'indica a continuació:

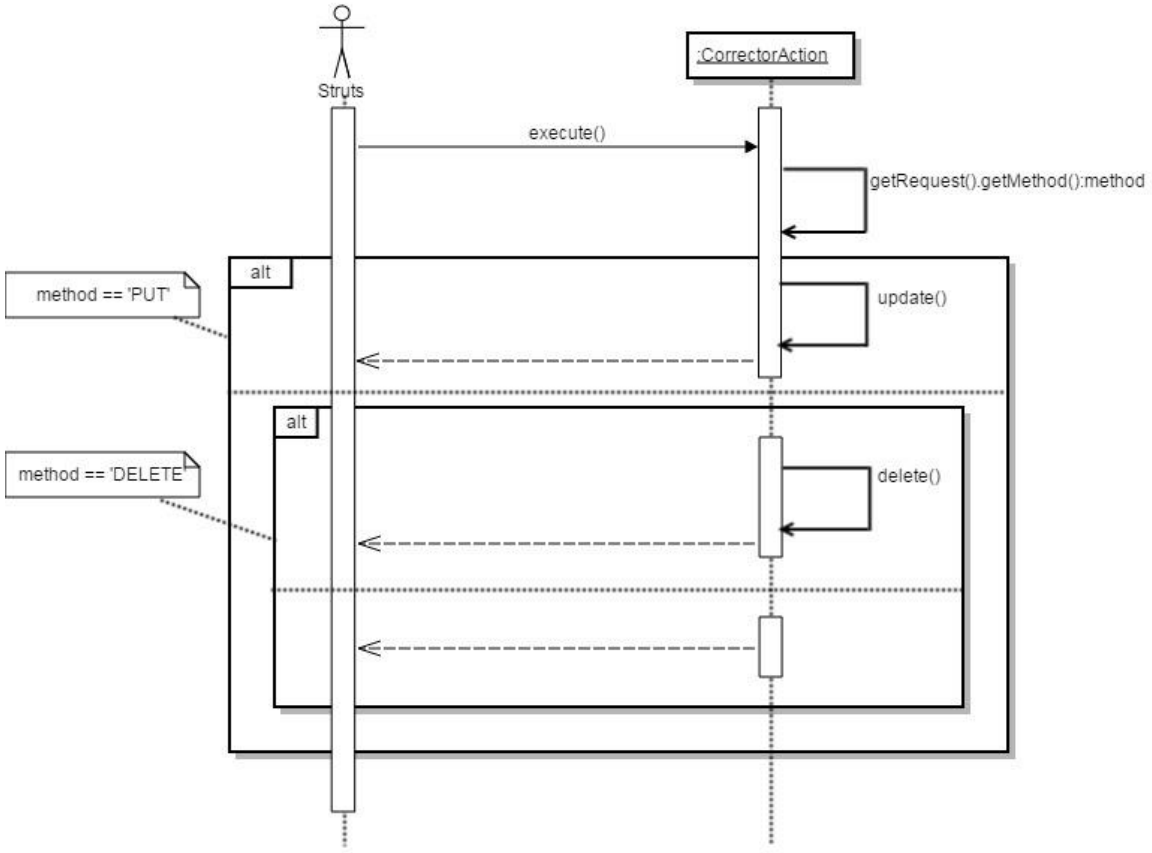
- `public`: Agrupa aquelles accions públiques que no requereix l'autenticació de l'usuari.
- `secured`: Agrupa aquelles accions que requereixen autenticació de l'usuari, però un cop autenticat, ja que fa petició a l'aplicació client per comprovar si l'usuari estigui autenticat.
- `secured-ws`: Agrupa aquelles accions que requereixen autenticació.

A continuació es pot veure la relació entre les peticions URLs i les *Actions* del sistema:

URL	***Action***	**Package**
`/correctors*`	-	secured
`/ws/questions/{id}/copy`	`QuestionCopyAction`	secured-ws
`/ws/questions/{id}/lock`	`QuestionLockAction`	secured-ws
`/ws/questions/{id}/copyBD`	`QuestionCopyDBAction`	secured-ws
`/ws/questions/{id}/execute_question`	`QuestionExecuteAction`	secured-ws
`/ws/questions/{id}/compare_question`	`QuestionCompareAction`	secured-ws
`/ws/correctors`	`CorrectorsAction`	secured-ws
`/ws/correctors/{id}`	`CorrectorAction`	secured-ws
`/ws/correctors/{id}/execute`	`CorrectorExecuteAction`	secured-ws
`/ws/simplequestions`	`SimplequestionAction`	secured-ws
`/ws/simplecorrectors`	`SimplecorrectorAction`	secured-ws
`/ws/configuration`	`ConfigurationAction`	secured-ws

Taula 12: Taula de relació entre URLs i les Actions

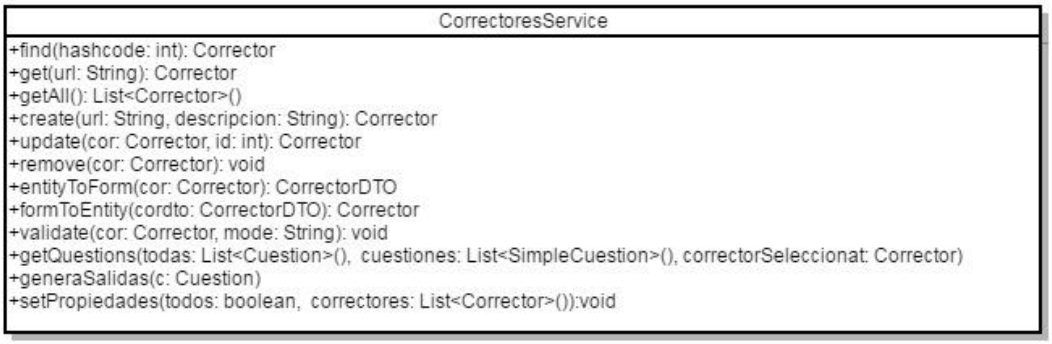
A continuació es pot veure un diagrama seqüencia de l'operació d'esborrar un corrector de l'acció `CorrectorAction`.



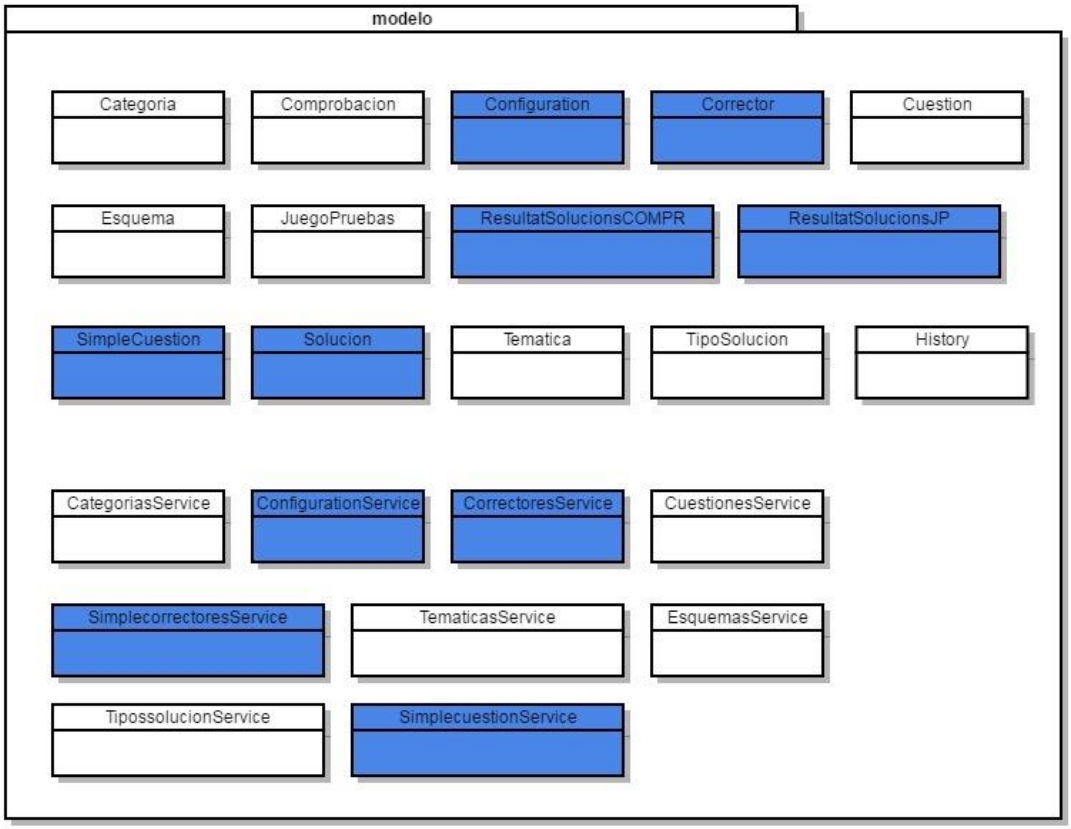
Il·lustració 12: Diagrama de l'operació d'esborrar un corrector

9.2.5. *Service*

Les classes *Service* són les classes que s'encarreguen de fer d'intermediari entre el consumidor i les operacions d'accés a dades, assegurar la validesa de les entitats, serialitzar / des-serialitzar l'entitat *DTO*. Aquestes classes són al paquet *modelo*, a continuació podem observar un exemple d'una classe d'aquest tipus: `CorrectoresService`.

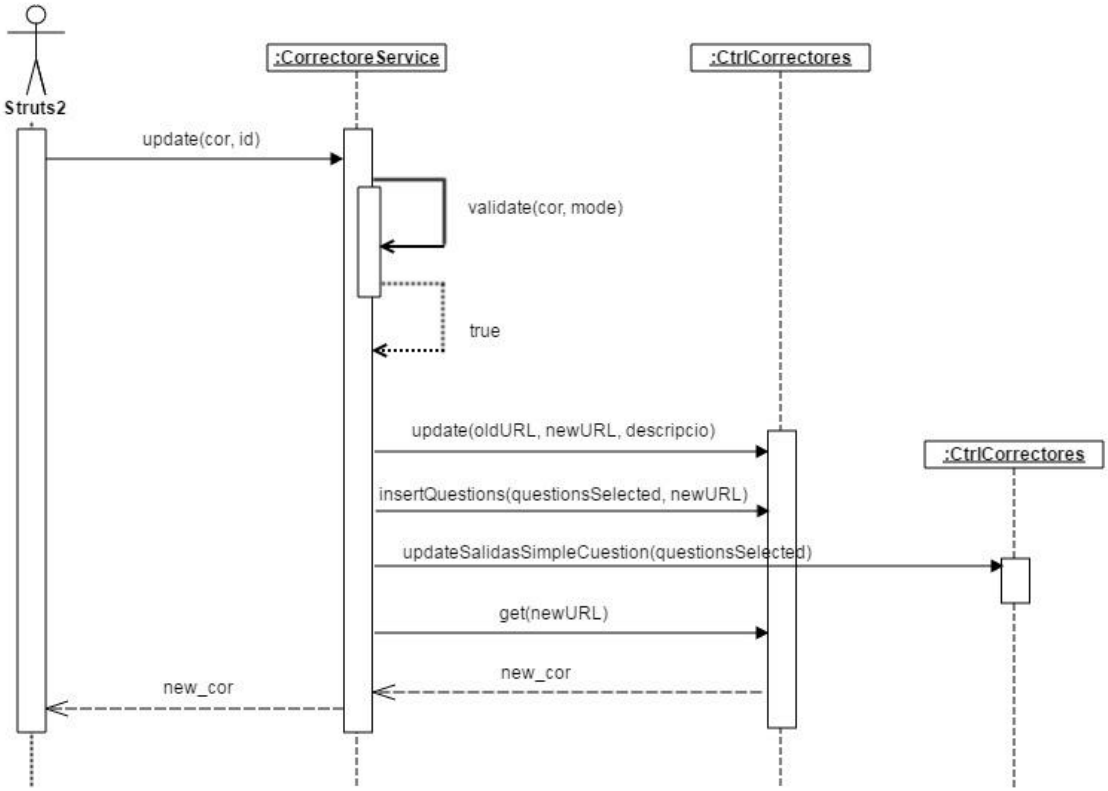


En l'ATW2 s'ha mantingut les entitats del paquet *modelo* tal com es va començar a fer al dissenyar l'ATW, i s'ha treballat sobre aquest. El nou paquest modelo queda organitzat de la manera següent. Les classes marcades de color blau són les noves classes afegides en el desenvolupament de l'ATW2.



Il·lustració 13: Paquet modelo

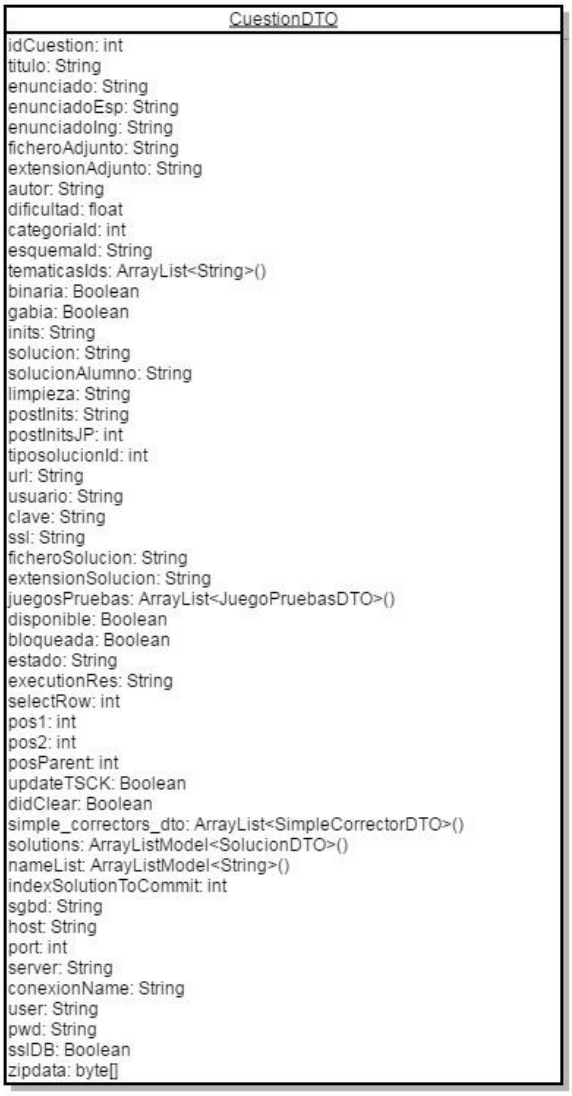
A continuació es pot veure un diagrama de seqüencia de l'operació de modificació de les dades d'un corrector: `CorrectorService`.



Il·lustració 14: Diagrama de l'operació modificació d'un corrector

9.2.6. *DTO*

Per poder convertir les dades que l'aplicació client envia a l'aplicació servidor apareix una classe *DTO (Data Transfer Object).* Aquesta classe té per objectiu permetre l'intercanvi d'informació entre client i servidor. Per cada entitat bàsica del paquet *modelo* s'ha de crear la seva classe *DTO* corresponent, a continuació veiem un exemple del *DTO* per l'entitat `Cuestion`.



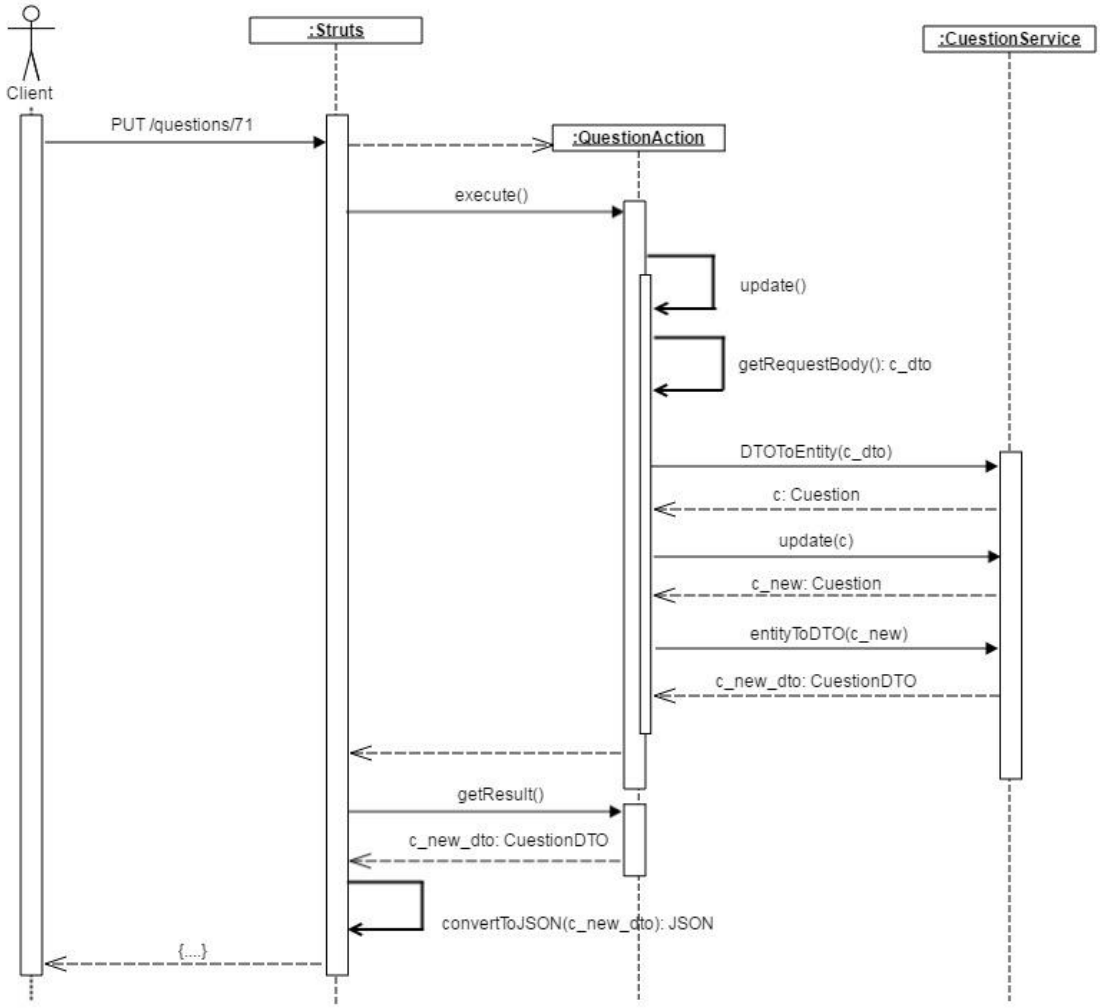
Quan l'aplicació client fa una petició a l'aplicació servidor s'assigna la responsabilitat a una de les classes *Action* gracies al patró *Front-Controller*. Posteriorment, la classe `JSONInterceptor` s'encarrega de convertir aquest cós de la petició en l'objecte *DTO* corresponent, la classe *Action* crida a la classe *Service* corresponent que s'encarrega de normalitzar el *DTO* corresponent a la classe entitat, un cop tingui la classe entitat actualitzada o creada, la classe *Service* es pot convertir-la a *DTO* altre cop, i finalment s'envia a l'aplicació client convertint *DTO* a *JSON*.

A continuació es pot veure el contingut del *JSON* corresponent a una qüestió.

```
{SGBD: "POSTGRESQL", SSLDB: true, autor: "carme.quer", binaria: null, bloqueada: false, categoriaId: 6,…}
  SGBD: "POSTGRESQL"
  SSLDB: true
  autor: "carme.quer"
  binaria: null
  bloqueada: false
  categoriaId: 6
  clave: ""
  conexionName: ""
  didClear: false
  dificultad: 1
  disponible: true
  enunciado: "<b>PASSOS PREVIS: </b> LLegir el contingut del fitxer PassosASeguir.txt (dins del zip del fit
  enunciadoEsp: null
  enunciadoIng: null
  esquemaId: "Professors"
  estado: null
  executionRes: "An error has occurred during the compilation: CtrlDadesPublic.java:5: error: cannot access
  extensionAdjunto: "zip"
  extensionSolucion: ""
  ficheroAdjunto: null
  ficheroSolucion: null
  gabia: null
  host: ""
  id: 71
  indexSolutionToCommit: 0
  inits: "create table professors↵(dni char(50),↵nomProf char(50) unique,↵telefon char(15),↵primary key (dn
▶ juegosPruebas: [{binaria: true, comprobaciones: [], consumeix: null,…},…]
  limpieza: "drop table  assignacions;↵drop table  professors;↵drop table  despatxos;"
  nameList: null
  port: 0
  pos1: -1
  pos2: -1
  posParent: -1
  postInits: ""
  postInitsJP: 0
  pwd: ""
  selectRow: -1
  server: ""
▶ simpleCorrectores: [{,…}, {,…}]
  solucion: "/* Imports de la classe */↵import java.sql.*; ↵import java.io.*;↵↵/* Capa de Control de Dades
  solucionAlumno: ""
▶ solutions: [{id: 71, isNew: "0", mostrar: "", nombreSolucion: "Nova solució - bona", notaBinaria: 1,…}]
  ssl: false
▶ tematicasIds: ["JDBC - consultes"]
  tiposolucionId: 1
  titulo: "BD-JDBC- Professors0 - Obtenir professors"
  updateTSCK: false
  url: ""
  user: ""
  usuario: ""
  zipdata: null
```


Il·lustració 15: JSON de les qüestions

A continuació, es pot veure el diagrama de petició de modificació de dades d'una qüestió per il·lustra aquest funcionament: `CuestionService`.



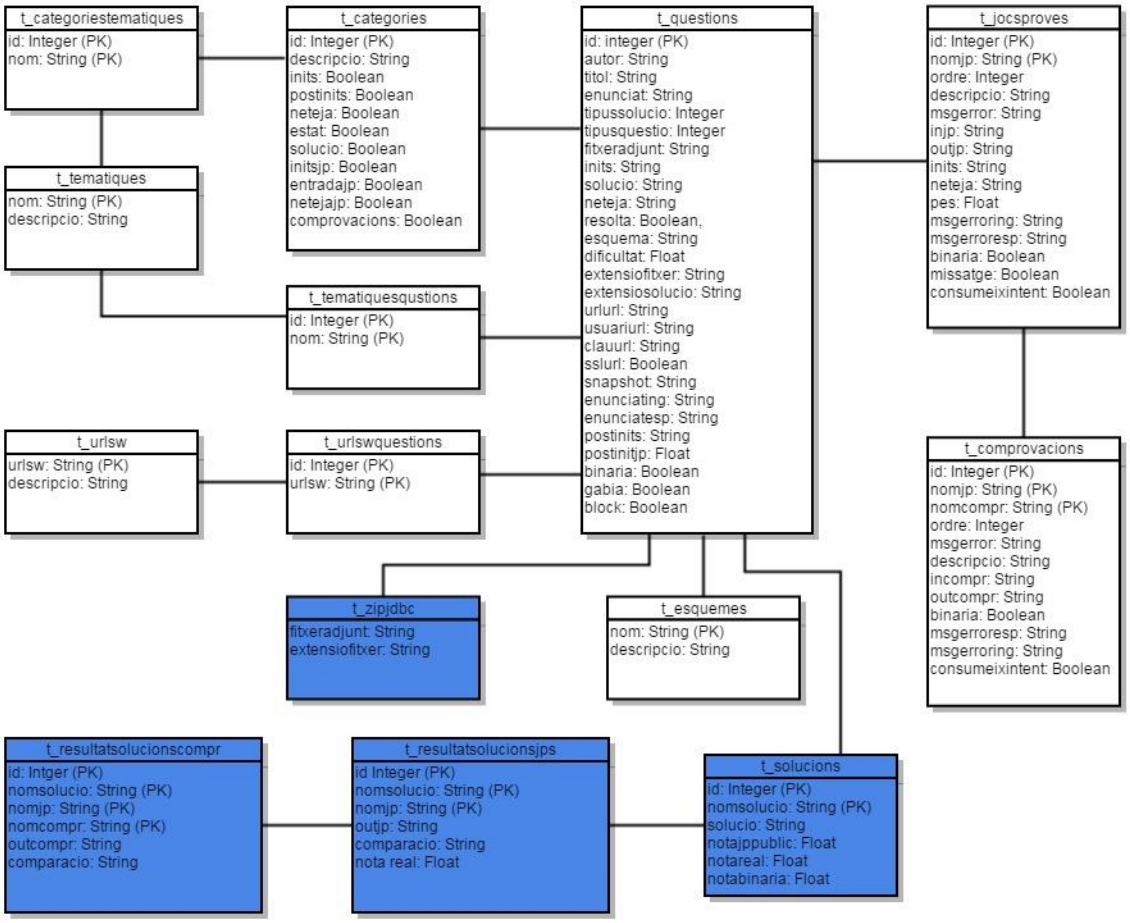
Il·lustració 16: Diagrama de petició de modificació d'una qüestió

9.2.7. Capa de dades

Aquesta capa està dissenyada per tal que la gestió de la base de dades faci d'una manera més eficient, a continuació es detallarà l'especificació de l'esquema lògic i físic d'aquesta capa.

9.2.7.1. Esquema lògic de la base de dades

Cal tenir en compte de que en l'ATW2 s'han afegit taules a la base de dades corresponents a les noves classes del model conceptual necessàries per donar suport a les noves funcionalitats de l'aplicació (veure apartat 7). Cal tenir en compte que les taules marcades en color són les noves taules afegides.



Il·lustració 17: Esquema lògic de la base de dades

9.2.7.2. Esquema físic de la base de dades

L'ATW2 té la base de dades en el sistema PostgreSQL, a l'igual que l'ATW de partida. PostgreSQL és un software lliure. A continuació s'indica les taules, atributs, claus primàries, claus foràneses i altres restriccions dels atributs de les taules.

t_questions
id integer NOT NULL DEFAULT nextval('t_questions_id_seq'::regclass) autor text NOT NULL titol text NOT NULL enunciat text NOT NULL tipussolucio integer NOT NULL tipusquestio integer NOT NULL fitxeradjunt bytea inits bytea solucio bytea NOT NULL neteja bytea resolta boolean NOT NULL DEFAULT false esquema text NOT NULL dificultat real extensiofitxer text extensiosolucio text urlurl bytea usuariurl bytea clauurl bytea sslurl boolean snapshot text enunciating text enunciatesp text postinits text postinitjp integer binaria boolean gabia boolean block boolean
t_questions_pkey PRIMARY KEY (id) t_questions_categories_fkey FOREIGN KEY (tipusquestio) REFERENCES esque.t_categories (id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE INITIALLY IMMEDIATE t_questions_esquema_fkey FOREIGN KEY (esquema) REFERENCES esque.t_esquemes (nom) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE INITIALLY IMMEDIATE

t_categories
id integer NOT NULL descripcio text NOT NULL inits boolean postinits boolean neteja boolean estat boolean solucio boolean initsjp boolean entradajp boolean netejajp boolean comprovacions boolean
t_categories_pkey PRIMARY KEY (id)

t_esquemes
nom text NOT NULL descripcio text NOT NULL
t_esquemes_pkey PRIMARY KEY (nom)

t_tematiques
nom text NOT NULL descripcio text NOT NULL,
t_tematiques_pkey PRIMARY KEY (nom)

t_urlsw
urlsw text NOT NULL DEFAULT ''::text descripcio text NOT NULL
t_urlsw_pkey PRIMARY KEY (urlsw)

t_zipjdbc
fitxeradjunt bytea, extensiofitxer text

t_solucions
id integer NOT NULL nomsolucio text NOT NULL solucio bytea NOT NULL notajppublic real notareal real notabinaria real
t_solucions_pkey PRIMARY KEY (id, nomsolucio) t_solucions_id_fkey FOREIGN KEY (id) REFERENCES esque.t_questions (id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

t_jocsproves
id integer NOT NULL ordre integer NOT NULL nomjp text NOT NULL descripcio text NOT NULL msgerror text NOT NULL injp bytea outjp bytea inits bytea neteja bytea pes real NOT NULL msgerroring text msgerroresp text binaria boolean missatge boolean consumeixintent boolean
t_jocsproves_pkey PRIMARY KEY (id, nomjp) t_jocsproves_id_fkey FOREIGN KEY (id) REFERENCES esque.t_questions (id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_jocsproves_id_ordre_unique UNIQUE (id, ordre)

t_comprovacions
id integer NOT NULL nomjp text NOT NULL nomcompr text NOT NULL ordre integer NOT NULL msgerror text NOT NULL descripcio text NOT NULL incompr bytea NOT NULL outcompr bytea binaria boolean msgerroresp text msgerroring text consumeixintent boolean
t_comprovacions_pkey PRIMARY KEY (id, nomjp, nomcompr) t_comprovacions_id_nomjp_fkey FOREIGN KEY (id, nomjp) REFERENCES esque.t_jocsproves (id, nomjp) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE INITIALLY IMMEDIATE t_comprovacions_id_nomjp_ordre_unique UNIQUE (id, nomjp, ordre)

t_resultatsolucionsjps
id integer NOT NULL nomsolucio text NOT NULL nomjp text NOT NULL outjp bytea NOT NULL comparacio text NOT NULL nota real NOT NULL
t_resultatsolucionsjps_pkey PRIMARY KEY (id, nomsolucio, nomjp) t_resultatsolucionsjps_id_fkey FOREIGN KEY (id, nomsolucio) REFERENCES esque.t_solucions (id, nomsolucio) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_resultatsolucionsjps_id_fkey1 FOREIGN KEY (id, nomjp) REFERENCES esque.t_jocsproves (id, nomjp) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_resultatsolucionsjps_comparacio_check CHECK (comparacio = ANY (ARRAY['correct'::text, 'error'::text]))

t_resultatsolucionscompr
id integer NOT NULL nomsolucio text NOT NULL nomjp text NOT NULL nomcompr text NOT NULL outcompr bytea NOT NULL comparacio text NOT NULL
t_resultatsolucionscompr_pkey PRIMARY KEY (id, nomsolucio, nomjp, nomcompr) t_resultatsolucionscompr_id_fkey FOREIGN KEY (id, nomsolucio) REFERENCES esque.t_solucions (id, nomsolucio) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_resultatsolucionscompr_id_fkey1 FOREIGN KEY (id, nomjp, nomcompr) REFERENCES esque.t_comprovacions (id, nomjp, nomcompr) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_resultatsolucionscompr_comparacio_check CHECK (comparacio = ANY (ARRAY['correct'::text, 'error'::text]))

t_categoriestematiques
id integer NOT NULL nom text NOT NULL
t_categoriestematiques_pkey PRIMARY KEY (id, nom) t_categoriestematiques_id_fkey FOREIGN KEY (id) REFERENCES esque.t_categories (id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_categoriestematiques_nom_fkey FOREIGN KEY (nom) REFERENCES esque.t_tematiques (nom) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE INITIALLY IMMEDIATE

t_tematiquesquestions
id integer NOT NULL nom text NOT NULL

t_tematiquesquestions_pkey PRIMARY KEY (id, nom) t_tematiquesquestions_id_fkey FOREIGN KEY (id) REFERENCES esque.t_questions (id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_tematiquesquestions_nom_fkey FOREIGN KEY (nom) REFERENCES esque.t_tematiques (nom) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE INITIALLY IMMEDIATE

t_urlswquestions
id bigint NOT NULL urlsw text NOT NULL
t_urlswquestions_pkey PRIMARY KEY (id, urlsw) t_urlswquestions_id_fkey FOREIGN KEY (id) REFERENCES esque.t_questions (id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION t_urlswquestions_urlsw_fkey FOREIGN KEY (urlsw) REFERENCES esque.t_urlsw (urlsw) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION DEFERRABLE INITIALLY IMMEDIATE

9.2.7.3. Accés a la base de dades

La configuració de la connexió a la base de dades es troba al fitxer `conexion.properties` que es pot donar valor al paràmetre *sgbd, server, host, port, ssl*, com podem observar la figura següent. Això fa que es pugui modificar les dades de la base de dades on hi ha les qüestions sense necessitat de recompilar el codi de l'aplicació.

```
sgbd:postgresql
server:postgresfib.fib.upc.edu
host:postgresfib.fib.upc.edu
port:5433
ssl:true
```

Il·lustració 18: Connexió a la base de dades

Pel que fa a la funcionalitat de mostrar resultats de les execucions d'una qüestió (OB:#7), per tal d'implementar-la l'aplicació servidor caldrà que es connecti a la base de dades del subsistema Remote Test Module: Plugin de Moodle [8] per a LearnSQL (veure secció 1.3.1). Per donar mantenibilitat, s'ha posat en un fitxer de configuració les dades de la base de dades de Moodle necessàries per poder fer la consulta de resultats de la qüestió es troba a `moodelConexion.properties`. Així doncs, en cas que les dades d'aquesta base de dades canviïn, només caldrà modificar aquest fitxer i no serà necessari recompilar l'aplicació.

```
sgbd:postgresql
server:postgresfib.fib.upc.edu
host:postgresfib.fib.upc.edu
port:5433
ssl:true
dbname:DBtestmoodle
```

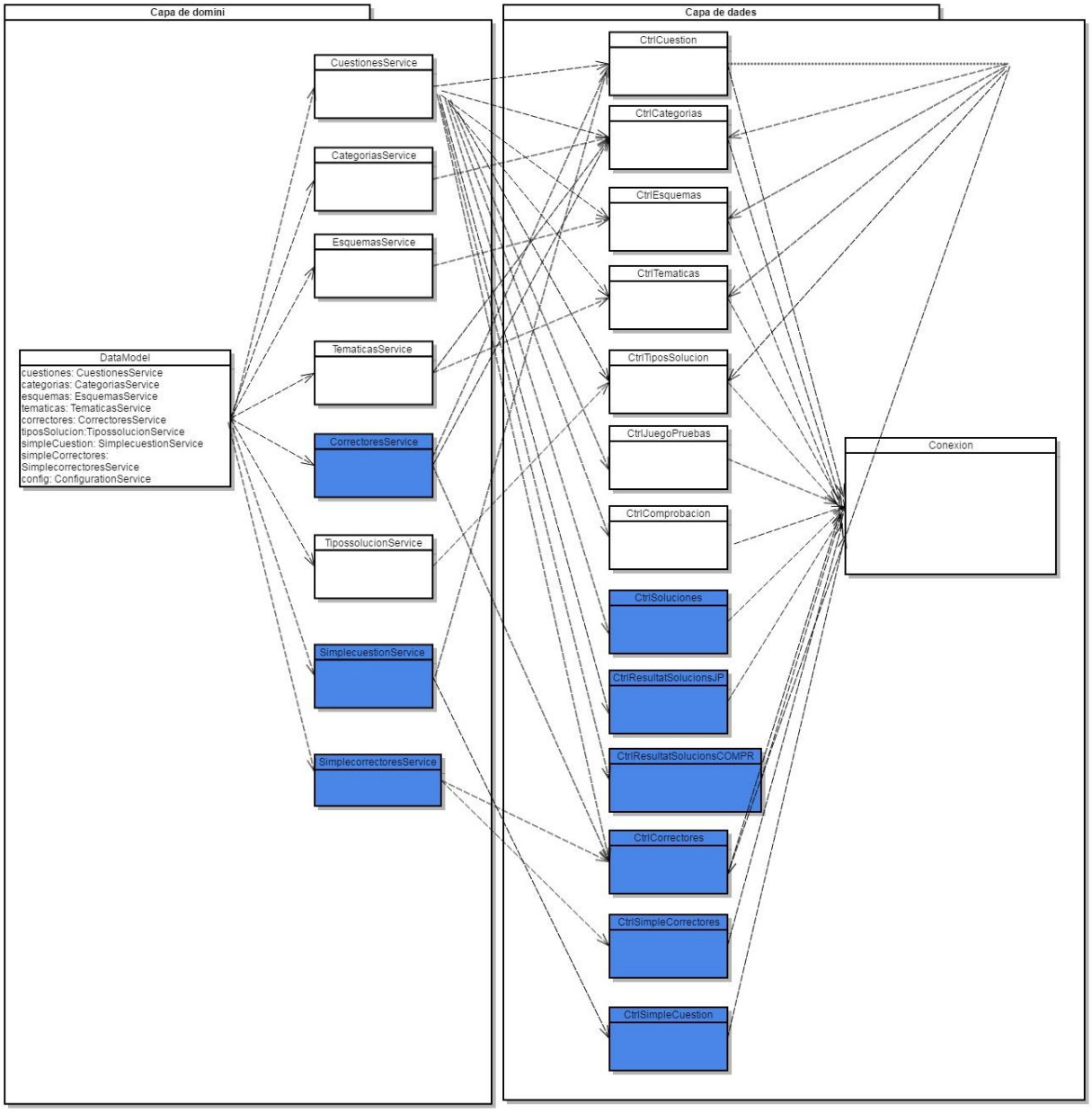
Il·lustració 19: Connexió per fer consulta de resultats

9.2.8. Autenticació

Per poder connectar a la base de dades, prèviament l'usuari necessita donar les seves credencials que les recollim a través de la pantalla d'accés, i l'acció `auth.LoginAction` és qui s'encarrega d'aquest control d'accés, un cop loguetjat l'usuari, les dades de credencials de l'usuari es guarda en sessió per a que les properes peticions es puguin executar sense demanar accés.

9.2.9. Relació entre capa de domini i capa de dades

Per tenir disponible les *Actions* de la capa de domini en qualsevol moment, es crea la classe `DataModel` que conté tots els *Services* per poder executar les operacions del domini, i cada classe *Service* es relaciona una o més classe *Controller* per poder accedir a la base de dades, a continuació podem observar la figura de com estan relacionat els objectes de dues capes. Cal tenir en compte que les classes marcades en color són les noves d'aquesta segona versió.



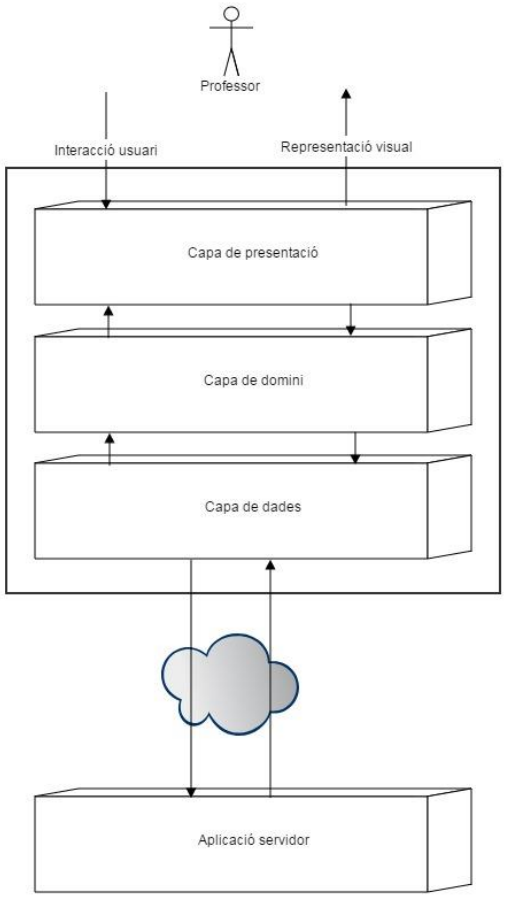
Il·lustració 20: Capa de domini i capa de dades

9.3. Disseny de l'aplicació client

En aquest apartat es detallarà el disseny de l'aplicació client, des de punt de vista d'arquitectura, la capa de dades, la capa de domini, i la capa de presentació. El disseny de la segona versió treballa sobre el de la primera, segueix utilitzant el *framework AngularJS* amb l'objectiu de desenvolupar una aplicació més àgil, reactiva i interactiva,

9.3.1. Arquitectura

La arquitectura de l'aplicació client està dissenyat en una arquitectura de 3 capes, tots els components d'una mateixa capa han de treballar al mateix nivell d'abstracció, els servei que proporciona una capa utilitzen serveis proporcionats per la capa inferior i ofereixen serveis a les capes superiors, l'avantatge principal d'aquesta arquitectura es permet separar la lògica del disseny de la lògica de negoci, i millorar la mantenibilitat, canviabilitat, i reusabilitat. En aquesta segona versió segueix la mateixa arquitectura de 3 capes que la primera versió, a continuació observem una figura que il·lustra aquesta arquitectura.



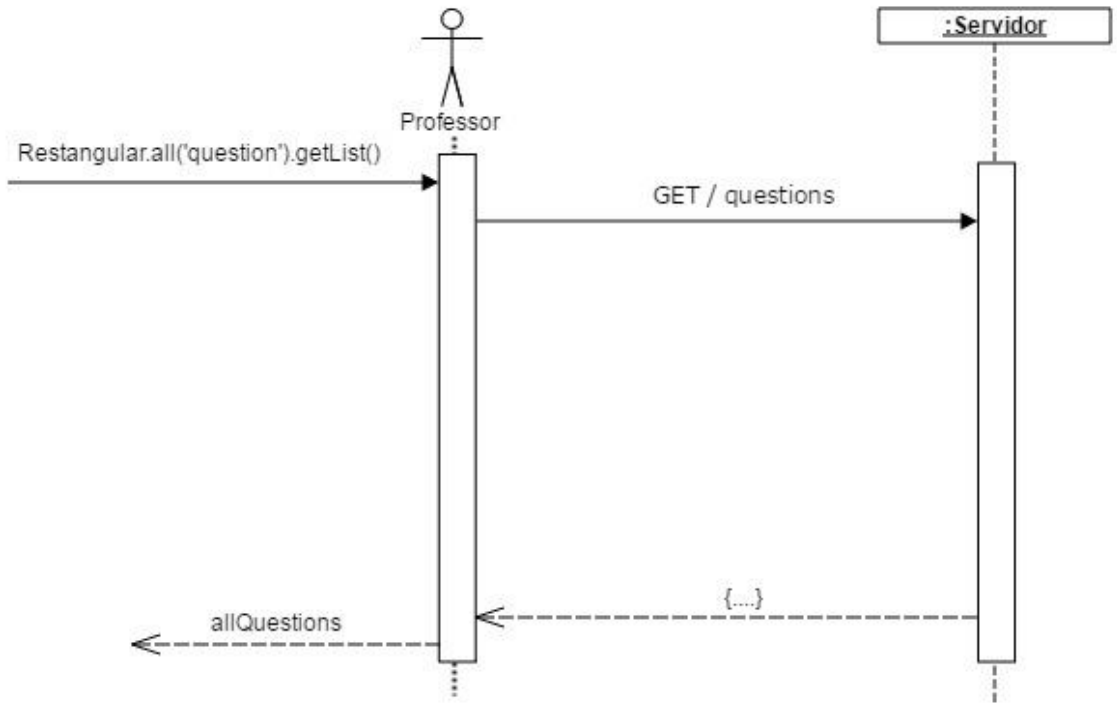
Il·lustració 21: Arquitectura de l'aplicació client

- **Capa de dades**: aquesta capa és responsable d'interactuar amb el SGBD (sistema de gestor de bases de dades), fer consultes, modificacions sobre aquest, i rebre peticions de la capa de domini,.
- **Capa de domini**: aquesta capa és responsable d'implementar tota la lògica de negoci, rebre peticions de la capa presentació, i mantenir l'estat de les dades de l'aplicació.
- **Capa de presentació**: aquesta capa és responsable d'interactuar amb l'usuari, i li proporciona una representació visual de l'aplicació.

9.3.2. Capa de dades

La capa de dades ha de comunicar-se amb l'aplicació servidor mitjançant el protocol *HTTP(s)* per a que l'aplicació client puguin servir, en concret permet obtenir totes les qüestions, categories, esquemes, temàtiques, correctors etc.., i enviar peticions de modificacions.

Per dur a terme la comunicació asíncrona amb l'aplicació servidor, utilitza la llibreria *Restangular* [26] que proporciona una capa d'abstracció per facilitar, i simplificar les peticions *HTTP(s)*, a continuació observem un exemple de petició GET de qüestions.



Il·lustració 22: Diagrama de l'operació de Restangular

9.3.3. Capa de domini

La capa de domini té finalitat de mantenir l'estat de les dades de negoci, i l'estat global de cert components específics de la interfície, fer el seguiment de canvis de les dades, comunicar amb la capa de dades. En aquest apartat detallarem les funcionalitats aplicades com Diccionaris, *Services*, *UiStateManager*¸ *WIP*.

9.3.3.1. Diccionaris i *Services*

Un Diccionari és una factoria que gestiona les dades de la seva entitat de negoci, les funcionalitat principals són:

1. Obtenir les dades de les entitats de negoci usant la capa de dades per a comunicar-se amb l'aplicació servidor.
2. Mantenir l'estat de les dades de negoci.
3. Fer el seguiment de canvis de les dades de negoci (nou elements, modificació d'elements, eliminació d'elements).
4. Permetre recuperar les modificacions de les dades.
5. Persistir les dades de les entitat.

Mentre que *Services* naixen amb la necessitat d'agrupar funcionalitats relacionades amb alguna entitat. Com el cas dels jocs de proves, les comprovacions, les solucions d'una qüestió. A continuació proposem un exemple de `SolutionsService`:

SolutionsService

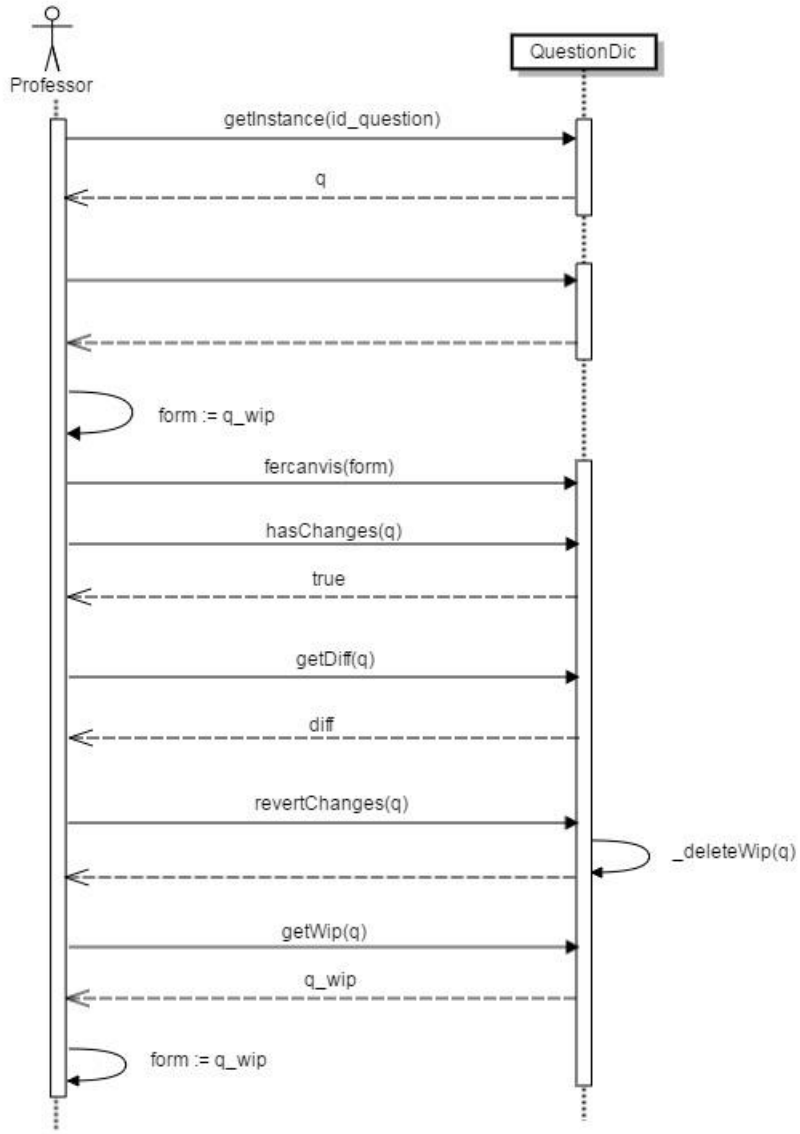
`findIndexByName(name, solutions)`
`findByIndex(index, solutions)`
`deleteByIndex(index, solutions)`
`addNew(name, id, solutions)`
`has(solutions)`

9.3.3.2. *UiStateManager*

Amb la necessitat de desar informació que els usuaris hagin obtingut, i recuperar-la, va nàixer l'objecte *UiStateManger*, bàsicament actua com un diccionari que emmagatzema un *key* i el seu *value* corresponent, té el mètode de get i set que es poden utilitzar en qualsevol lloc de l'aplicació.

9.3.3.3. *Work In Process (WIP)*

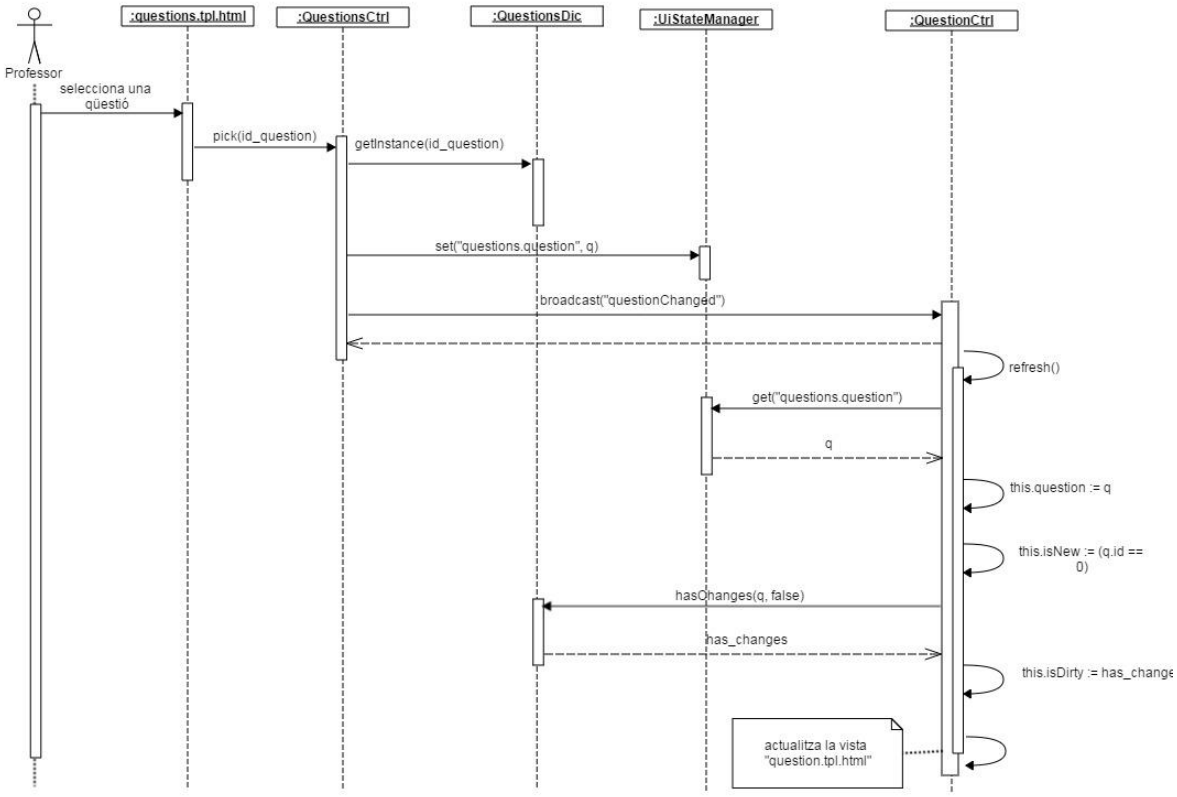
La idea de *WIP* és bàsicament fer una "copia se seguretat" d'entitat emmagatzemada en el Diccionari, per tant pot treballar sobre la copia sense preocupacions, sempre té la opció de recuperar la versió original en el moment que vulgui. A continuació veiem un exemple d'obtenir una qüestió a poder modificar-la, i recuperar la versió original.



Il·lustració 23: Diagrama de WIP

9.3.3.4. Diagrama global i visió general

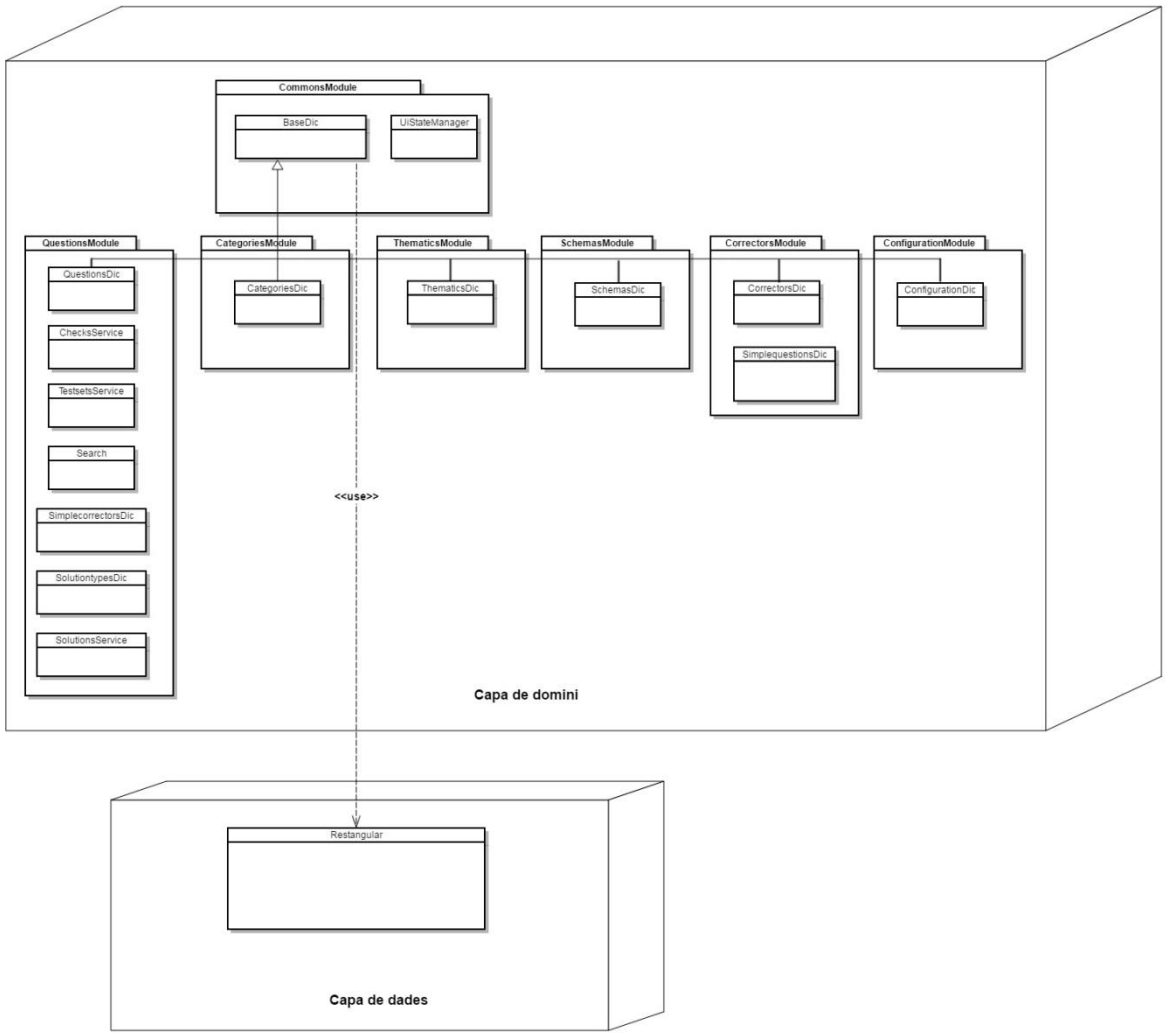
A continuació veiem un diagrama per il·lustrar el funcionament de Dicccionari, *UiStateManager*¸ i WIP. I una visió general de com està estructurat internament la capa de domini.



Il·lustració 24: Diagrama global

Com pot veure QuestionsCtrl s'encarrega d'indicar quina qüestió està seleccionada, i es comunica mitjançant senyals *d'AngularJS amb* QuestionCtrl cada cop hagi un canvi de la selecció, posteriorment *UiStateManager* s'encarrega de guardar l'entitat de qüestió, i la pot consultar en el moment que vulgui, quan es detecta un canvi de la vista, QuestionCtrl avisa al QuestionsDic, i aquest es capaç de saber quins canvi hagi fet fent servir *WIP*, si confirma que ha fet canvis, s'actualitza la vista degudament.

A continuació veiem com està estructurat els mòduls de la capa de domini, i la relació amb la capa de dades.



Il·lustració 25: Estructura dels mòduls

Com es pot veure, al mòdul Common es troba la classe *BaseDic* que proporcionar a tots els diccionaris totes aquestes funcions comunes, i fa servir les funcions de la llibreia *Restangular*.

9.3.4. Capa de presentació

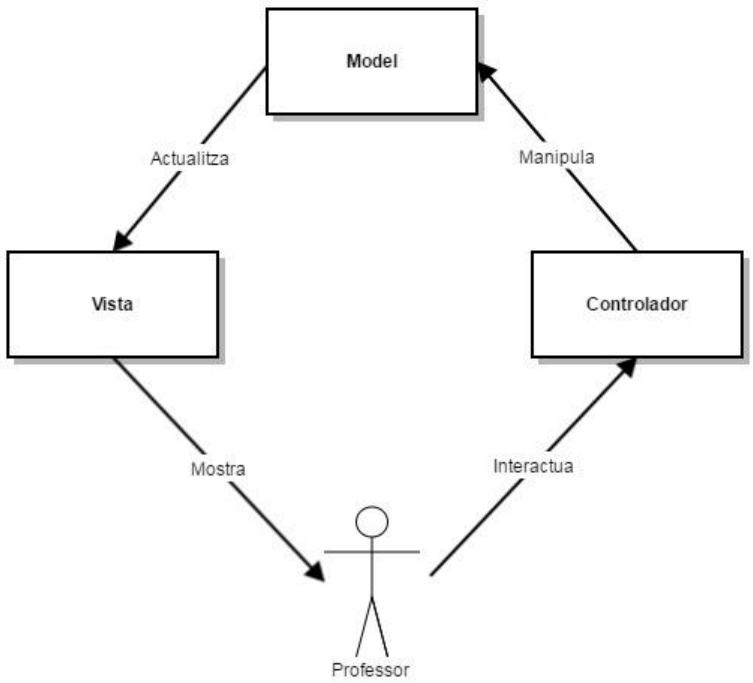
La capa de presentació és l'encarregada de gestionar la interacció amb l'usuari, en aquest apartat detallarem el patró d'arquitectura, la organització, esquema de components, i el disseny de les pantalles. En aquesta segona versió, afegiran dos mòduls més respecte la primera versió, que són:

- **`Correctors:`** agrupa tots els components relacionats amb els casos d'ús de Corrector.
- **`Configuration:`** agrupa tos els components relacionats amb els casos d'ús de Configuration.

9.3.4.1. *MVC*

La capa de presentació de l'aplicació client ha de contenir un mar de components, a més a més, quan hi ha un sol canvi en el model d'un component, s'ha de ser capaç de modificar un gran quantitat de components, el patró *MVC (Model-View-Controller)* encaixa bé per aquesta paradigma, ja que els tres tenen el següent comportament:

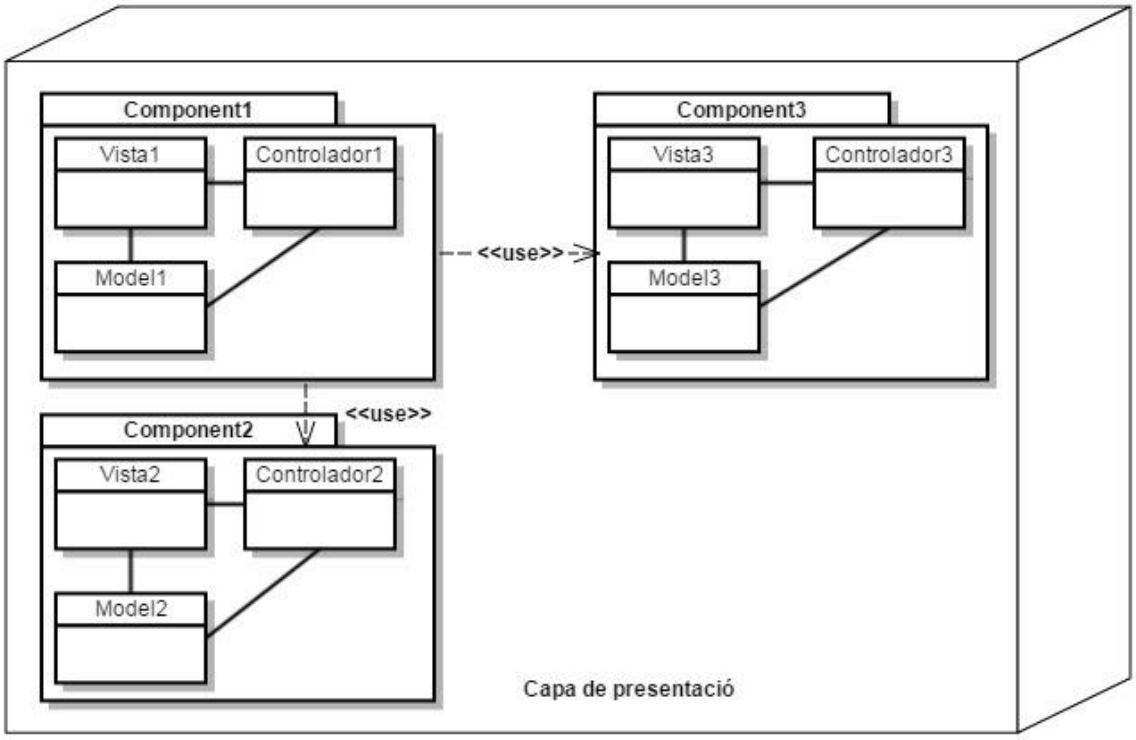
- **Model**: s'encarrega de mantenir l'estat de la vista, ha de passar la nova dada del model a la vista quan hi ha un canvi de l'estat.
- **Vista**: s'encarrega de representar visualment les dades del model mitjançant la interfície gràfica, i poder interactuar amb els usuaris.
- **Controlador**: s'encarrega de rebre accions de l'usuari, i interactuar amb el model mitjançant les peticions.



Il·lustració 26: Model-Vista-Controlador

9.3.4.2. Components

Per poder organitzar la capa de presentació d'una manera àgil, es segueix l'estratègia de la primera versió que utilitza Components, que és equivalent una unitat lògica que té una finalitat determinada, cada component està compost per un *MVC*, a continuació observem una figura que representar la idea de component.

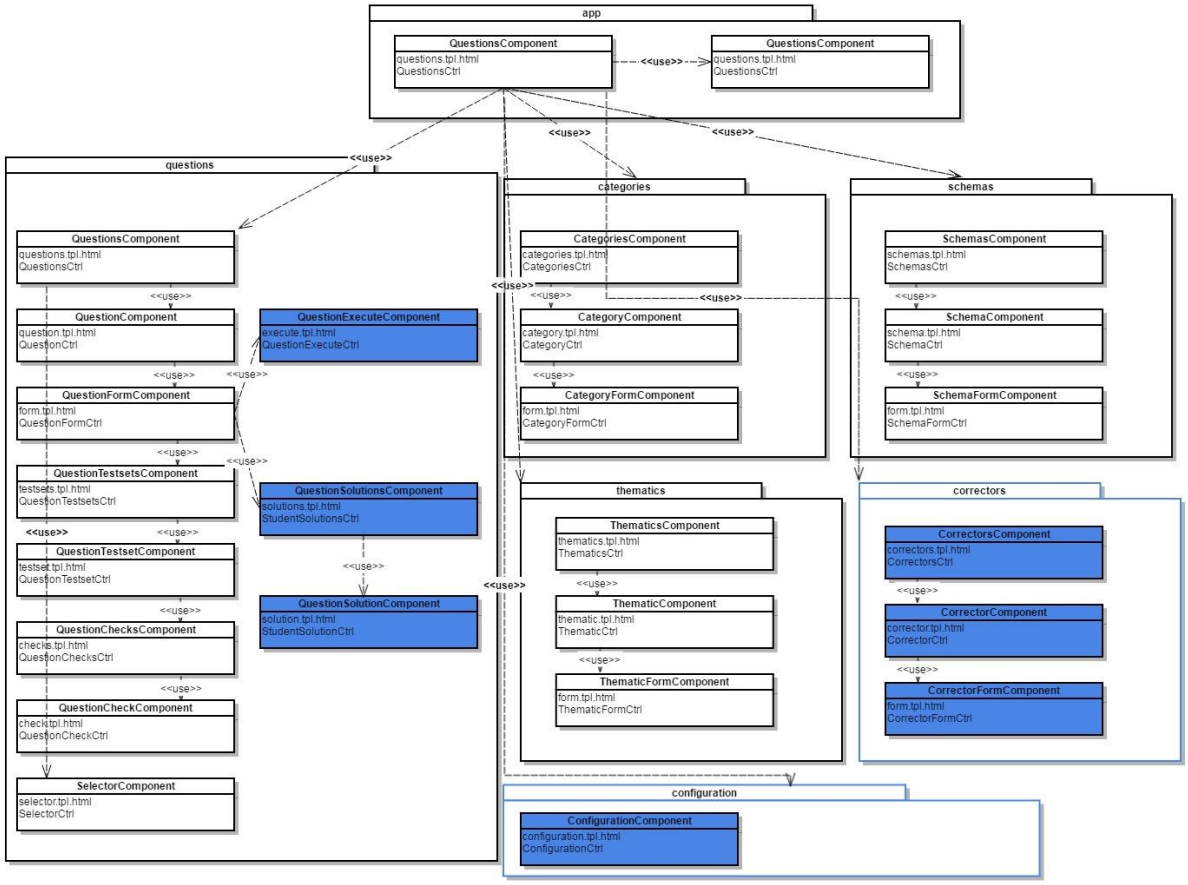


Il·lustració 27: Capa de presentació

9.3.4.3. Esquema de Components

En aquesta segona versió, s'ha afegit més components relacionats amb els nous objectius, a continuació veiem una visió global d'esquema de components que esta format per diferents mòduls relacionats entre ells, el mòdul *App* conté tots els altres mòduls, ja que s'ha de carregar tots els components essencials a la primera arrancada, els altres mòduls contenen tots els components relacionat amb el tema d'aquest mòdul. Cal tenir en compte que els components amb color són els nous d'aquesta segona versió de l'ATW.

Com podem veure, cada vista (*HTML* [27]) té el seu propi controlador, que s'encarrega d'exposar un model de dades a la vista.



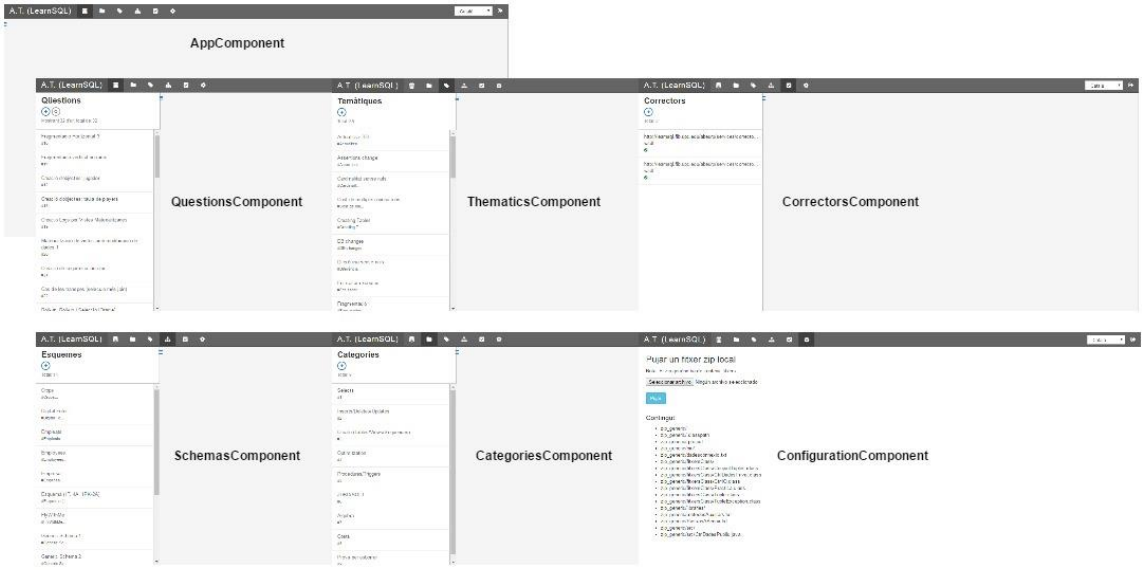
Il·lustració 28: Esquema de Components

9.3.4.4. Disseny de les pantalles

En aquest apartat es mostra el disseny de les pantalles de l'aplicació que són destacades en la primera i la segona versió juntes.

- **Pantalles globals**

A la figura següent es poden observar les seccions principals de l'aplicació, cada component es correspon amb un mòdul, i una pantalla, el *QuestionsComponent, ThematicsComponent, SchemasComponent, CateogoriesComponent* i *CorrectorsComponent* es disposen un menú lateral on es poden veure totes les unitats relacionades, i cadascun conté els seus propis subcomponents. El component principal *AppComponent* disposa una barra de navegació on permet a l'usuari anar a la secció que vulgui.



Il·lustració 29: Pantalles globals

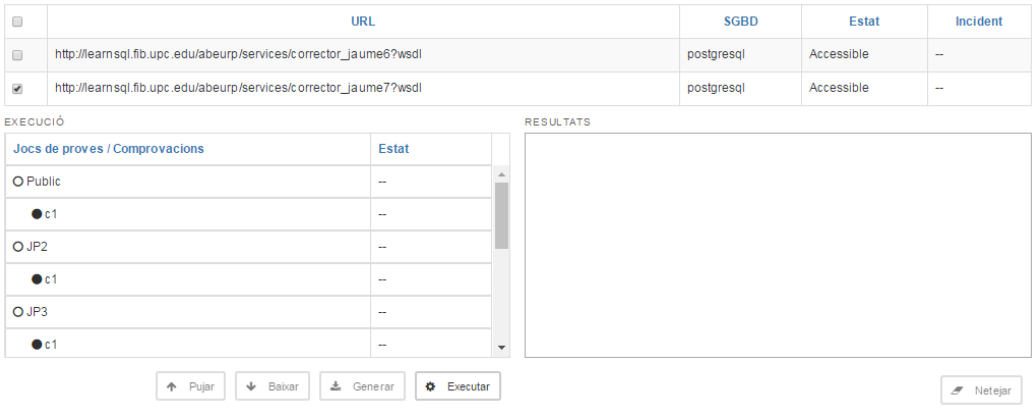
- ***QuestionsComponent***

Podríem dir que el component *QuestionsComponent* és un llistat de *QuestionComponent*¸i *QuestionComponent* conté el component fill que és *QuestionFormComponent*, que conté diverses seccions (components) com "Dades generals", "Sentències SQL", "Jocs de proves", "Pesos", "Estat BD", "Panell d'execució", "Solucions d'estudiants", i "Resultats". Algun d'aquests components contenen més subcomponents, com el cas de "Jocs de proves" i "Solucions d'estudiants". D'aquesta manera, cada qüestió es pot gestionar de manera independent. A continuació observem un arbre jeràrquic de components de *QuestionsComponent.*

Il·lustració 30: Conjunt de les pantalles de QuestionsComponent

- **Panell d'execució**

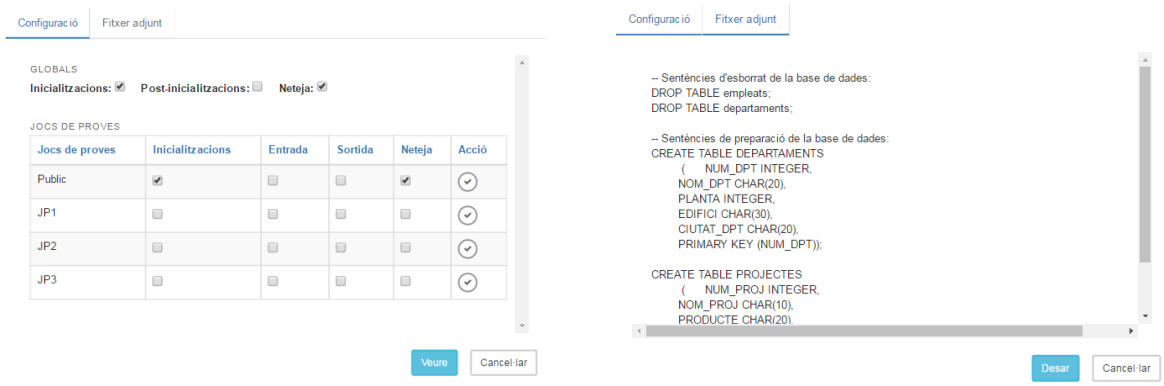
El panell d'execució és una secció de *QuestionExecuteComponent* esta dins de QuestionFormComponent, que esta compost en diferents elements. A la part superior es troba una taula de tots correctors, els correctors que estan marcats són els que estan relacionats amb la qüestió, es mostra la informació de cada corrector com URL, SGBD, estat, i incident si hi ha. A la part inferior esquerra es troba una taula de jocs de proves i comprovacions, més avall es troben 4 botons, el boto "Pujar" i "Baixar" s'encarreguen d'ordenar els jocs de proves i comprovacions, s'activen quan l'usuari doni clic al joc de proves o la comprovació, al clicar el botó "Generar", s'activa un Pop-up per generar el fitxer adjunt, al clicar el botó d'executar, s'executa la qüestió amb els correctors assignats. A la part inferior dreta es troba els logs acumulatius cada que s'executa la qüestió, i al clicar el botó "Netejar" es poden netejar els logs.



Il·lustració 31: Pantalla de panell d'execució

- **Pop-up fitxer adjunt**

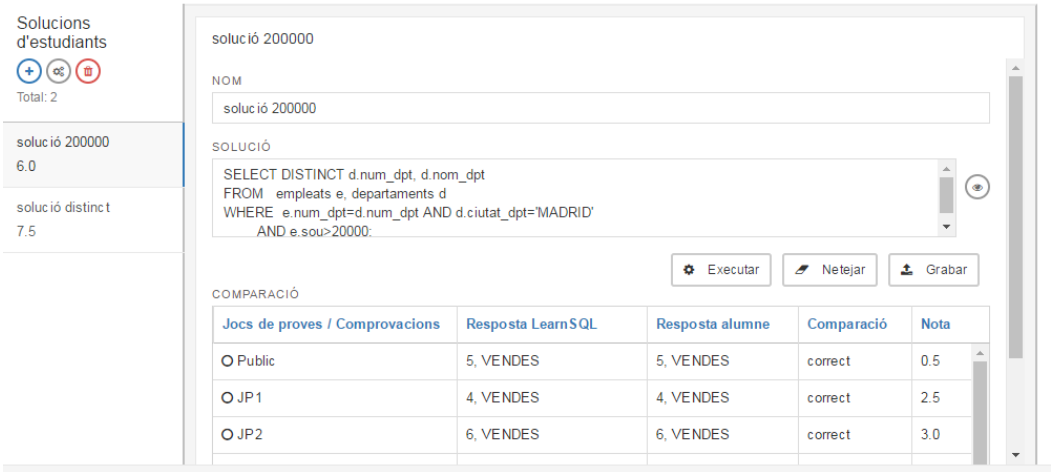
En aquest Pop-up podem veure dos subseccions, la secció "Configuració" es troben els 3 *checkbox* i una taula de jocs de proves, al clicar la fila d'un joc de proves, es sortirà una altra taula més avall que conté les comprovacions d'aquest joc de proves, el boto de *chebox* es pot marcar o desmarcar tots els *checkbox* de la mateixa fila, a continuació es troben els dos botons de la part inferior, el botó "Veure" es mostra el contingut del fitxer adjunt en funció dels *checkbox*, i el botó "Cancel·lar" es tanca el Pop-up immediatament. El botó "Desar" permet guardar el contingut generat a la base de dades.



Il·lustració 32: Pantalles dels Pop-up

- **Solucions d'estudiants**

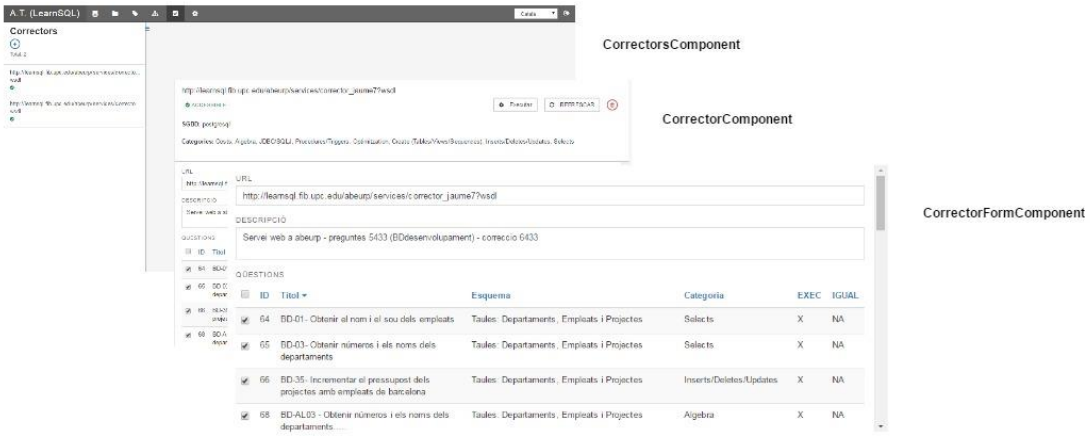
La pantalla de solucions d'estudiants una secció de Question*SolutionsComponent,* que disposa un menú lateral on es pot trobar totes les solucions de la qüestió, i els botons per afegir, eliminar la solució, i executar les solucions en bloc. Cada solució es representa una secció de Question*SolutionComponent* que conté el nom de la solució, el text de la solució, i la taula de la comparació de jocs de proves i comprovacions. Al clicar el boto "Executar", s'executa la solució, i es genera una taula de la comparació, la nota *Public*, la nota binaria, i la nota real. Al clicar el boto "Netejar", s'esborra les dades parcials de la taula, el text de la solució, i les tres notes. Al clicar el boto "Grabar", es guarda l'estat actual de la solució a la base de dades.



Il·lustració 33: Pantalla de solucions d'estudiants

- **Correctors**

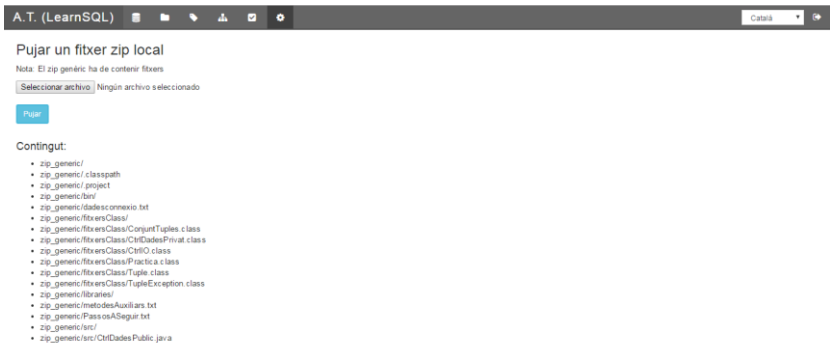
A la pantalla de correctors representa la secció de *CorrectorsComponent* que disposa un menú lateral on es mostra tots els correctors, i cada corrector que representa la secció de *CorrectorFormComponent* es troba la informació com l'URL, el SGBD, les categories, la descripció, i les qüestions assignades o no assignades amb la seva informació.



Il·lustració 34: Conjunt de pantalles de CorrectorsComponent

- **Panell zip genèric**

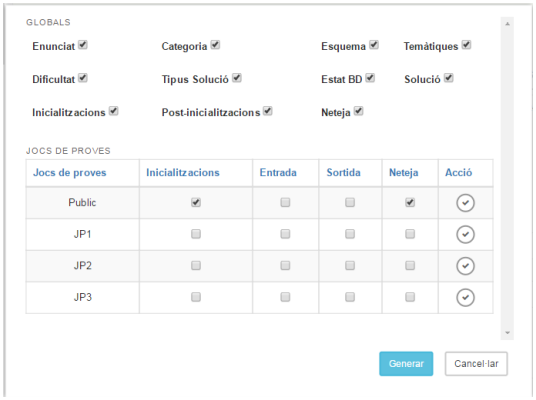
A la part superior de la pantalla del panell zip genèric es pot observar un boto per seleccionar el fitxer en extensió .zip de local, i un boto "Pujar" per carregar aquest mateix fitxer zip. A la part inferior es troba un llista de tots els arxius i directoris d'aquest zip genèric per a què els professors sàpiguen el contingut del zip genèric.



Il·lustració 35: Pantalla de panell zip genèric

- **Generar llistats**

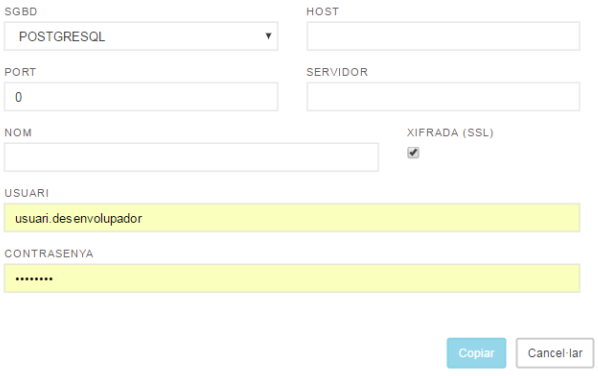
A aquest Pop-up es troba la part de globals on contenen els *checkbox* per especificar, i la part de jocs de proves es troba una taula de jocs de proves on per cada joc de proves hi ha 4 *checkbox* i un boto per marca i desmarcar tots els 4 *checkbox*. Al clicar la fila de joc de proves, es sortirà una altra taula de comprovacions si aquest joc de proves té comprovacions relacionades. Al donar clic al botó "Generar", es baixarà un fitxer txt amb totes les informacions personalitzades de la qüestió.



Il·lustració 36: Pantalla de Pop-up de generar llistat

- **Copiar la qüestió**

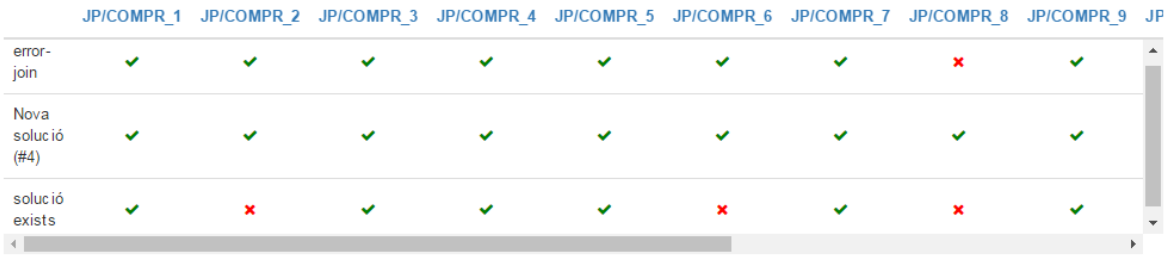
Aquest Pop-up serveix per connectar a l'altra base de dades, i copiar la qüestió a aquesta base de dades, per poder fer-ho, es necessita introduir els paràmetres com: el nom de SGBD, el número de host, el número de post, i nom d'esquema de base de dades, xifrat o no, i l'usuari, contrasenya.



Il·lustració 37: Pantalla de Pop-up de copiar la qüestió

- **Matriu de solucions i jocs de proves, comprovacions**

Aquesta matriu es pot veure globalment el resultat de cada joc de proves o comprovació de cada solució, el *checkbox* en color verd vol dir que el resultat que dona joc de proves o comprovació de la solució d'estudiant és equivalent que el resultat correcte. En canvi, si no és equivalent, mostra una creu de color vermella.



Il·lustració 38: Matriu de solucions i jocs de proves, comprovacions

- **Taula de l'historial**

Aquesta taula mostra per cada execució que hi ha en un curs de Moodle-LearnSQL, les notes mitjanes dels estudiants, i el número d'estudiants que l'han fet.

Nom del curs	Grup d'estudiants	Nota mitjana sense penalitzacions	Nota mitjana amb penalitzacions	Nº equips contestats	Nº equips que podien contestar
RDB - Trials	G10	10	10	1	2
RDB - Trials	G20	8.5	7.55	2	2

Il·lustració 39: Taula de l'historial

A continuació mostra una descripció de les responsabilitats de Components més destacats de l'aplicació.

Component	Responsabilitat
AppComponent	Controla la distribució general de l'aplicació, el menú de seccions principals, i el selector d'idioma.
QuestionsComponent	Controla el llistat de qüestions del domini, amb els seus indicadors, el filtre i la poder de seleccionar, afegir, eliminar una qüestió.
QuestionComponent	Controla la informació de la qüestió que està seleccionada, i els seus indicadors d'estat, i els botons d'accions disponibles.
QuestionFormComponent	Controla el formulari de la qüestió, amb totes les diferents pestanyes, camps habilitats, deshabilitats.
CorrectorsComponent	Controla el llistat de correctors del domini, amb els seus indicadors.
CorrectorComponent	Controla la informació del corrector que està seleccionat, i els seus indicador d'estat, i els botons d'accions disponibles.
CorrectorFormComponent	Controla el formulari del corrector amb totes les qüestions relacionades.
ConfigurationComponent	Controla la informació de zip genèric.
QuestionExecuteComponent	Controla la informació d'execució de la qüestions, i els botons disponibles.
QuestionSolutionsComponent	Controla el llistat de correctors del domini, amb els seus indicadors, i la poder de seleccionar, afegir, eliminar un corrector.
QuestionSolutionComponent	Controla el formulari del corrector seleccionat.

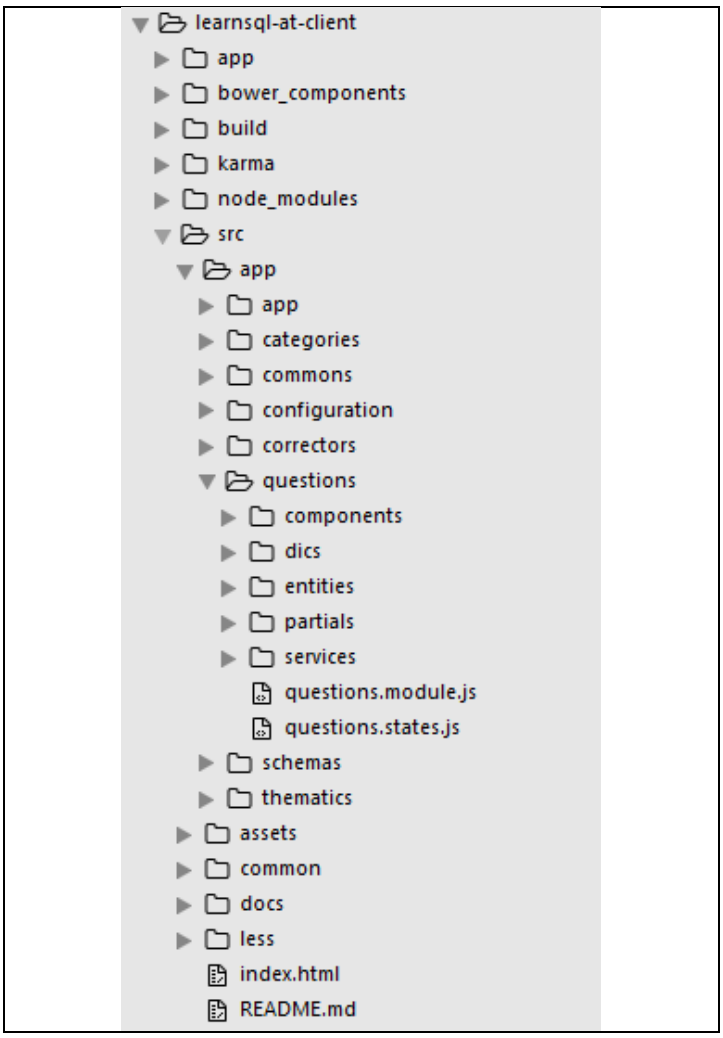
10. Implementació

La segona versió segueix utilitzant les mateixes tecnologies de la primera. Per la part del *Frontend*, s'ha utilitzat en l'entorn de desenvolupament *Sublime Text*, per la part del *Backend*, s'ha realitzat en l'entorn de desenvolupament *Eclipse*. A partir de les eines establertes pel desenvolupament, en aquest capítol, detallarà els detalls de la implementació, i les tecnologies utilitzades.

10.1. Detalls de la implementació

A continuació es mostra la implementació destacades de la part del *Frontend* i *Backend*, i com estructurat del codi del projectes.

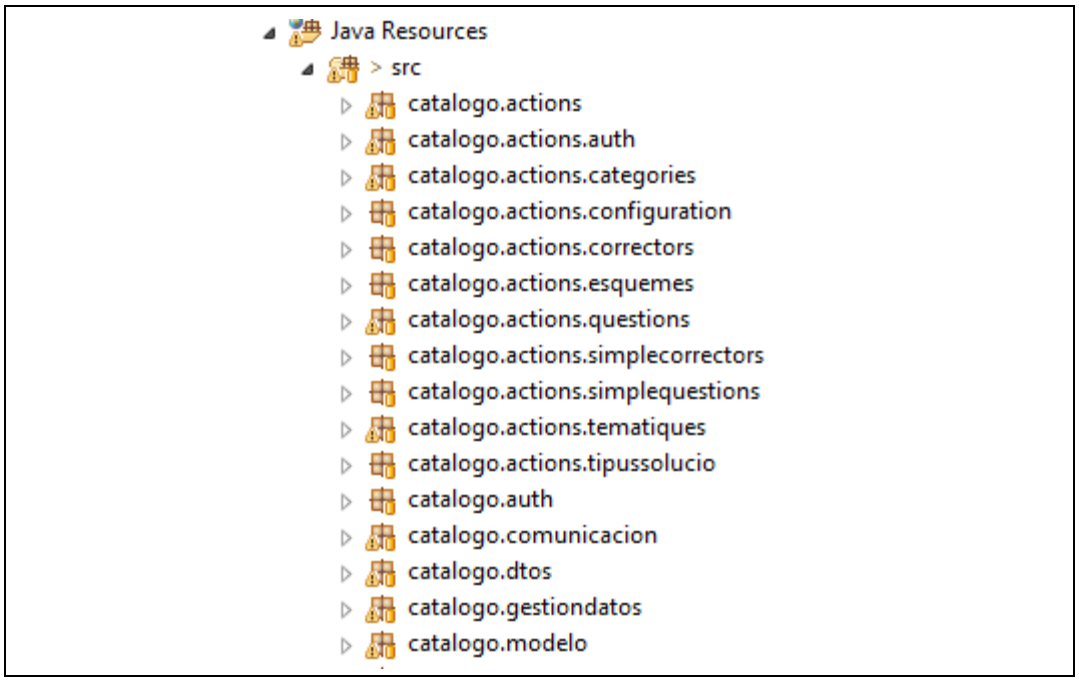
- **Estructura del codi del projecte *Frontend***



Il·lustració 40: Estructura de codi del repositori client

Tal com es pot observat, per cada mòdul de l'aplicació client s'ha creat la seva paquet corresponent, en cada paquet mòdul al menys hi ha un sub-paquet "component" i un sub-paquet "dics".

- **Estructura del codi del projecte *Backend***



Il·lustració 41: Estructura de codi del repositori servidor

Tal com es pot observar, s'han creat un seguit de sub-paquets segons la seva temàtica, per tenir una estructura ben senzilla, comprensible i fàcil d'organitzar a l'hora d'afegir classes o paquets.

- **Accés a la base de dades**

```
try {
    WrapperConexion wcon = new WrapperConexion(
            cuestion.getSGBD().trim(),
            cuestion.getHost().trim(),
            String.valueOf(cuestion.getPort()),
            cuestion.getServer().trim(),
            cuestion.getConexionName().trim(),
            cuestion.getUser(),
            cuestion.getPwd(),
            cuestion.getSSLDB()
        );
    wcon.conectar();
    con = wcon.getConexion();
```

Per poder connectar a la base de dades, es necessiten els paràmetres com nom de SGBD, el número de host, port, l'esquema etc..

- **Obtenir categories**

```
ResultSet rs = null;
Statement s = null;

String sql = "SELECT * FROM t_categories";
String where = "";
for (int id : ids) {
    where += " OR id = '"+id+"'";
}
sql += " WHERE "+where.substring(4)+" ORDER BY id";

try {
    s = con.createStatement();
    rs = s.executeQuery(sql);
    while (rs.next()) {
        categorias.add(new Categoria(
            Integer.parseInt(rs.getString("id")),
            rs.getString("descripcio"),
            rs.getBoolean("inits"),
            rs.getBoolean("postinits"),
            rs.getBoolean("neteja"),
            rs.getBoolean("estat"),
            rs.getBoolean("solucio"),
            rs.getBoolean("initsjp"),
            rs.getBoolean("entradajp"),
            rs.getBoolean("netejajp"),
            rs.getBoolean("comprovacions")
        ));
    }
    con.commit();
```

Per poder obtenir totes les categories, s'executa el *SELECT* corresponent, i per cada *ResultatSet* que retorna, s'omple el objecte amb les dades.

- **Update un temàtic**

```
@Override
protected String update() throws Exception
{
    TematicaDTO tem_form = (TematicaDTO) this.getRequestBody();
    if (!tem_form.getId().equals(this.getId())) {
        this.getResponse().setStatus(400);
        return this.respondAsJson("ids don't match");
    }
    Tematica tem = this.model.tematicas.formToEntity(tem_form);

    Tematica tem_new =
    this.model.tematicas.update(tem, getId());

    tem_form = this.model.tematicas.entityToForm(tem_new);
    return this.respondAsJson(tem_form);
}
```

Per fer un *Update* d'una temàtica, primer agafa el *JSON* provinent del client, transforma-ho en una entitat *Tematica*, posteriorment transforma la nova temàtica en *DTO*, finalment fa una conversió de *DTO* a *JSON*, i retorna-ho al client.

- **Restangular**

```
/**
 * Populates the dictionary with the data in the server.
 */
loadAll(successCallback, errorCallback) {
    var all,
        _this = this;
    if (this.__isLocked()) {
        $timeout(function(){
            _this.loadAll(successCallback, errorCallback);
        }, 0);
        return;
    }

    if (this.__isAlreadyCached()) {
        all = this.registry;
        successCallback(all);
    }
    else {
        this.__lock();
        Restangular
        .all(this.resourceName)
        .getList()
        .then(
            function onSuccess(response) {
                all = response.data;
                _this.__registerAll(Restangular.stripRestangular(all));
                if (successCallback) { successCallback(all); }
            },
            function onError(response) {
                _this.__registerAll([]);
                if (errorCallback) { errorCallback(response); }
            }
        ).finally(function() {
            _this.__unlock();
        });
    }
}
```

Mitjançant la funció de Restangular¸fa una petició de *GET* per obtenir la llistat d'entitat d'un *resource* que potser una qüestió, una categoria, una temàtica etc..

- **SolutionService**

```
findByIndex(index, solutions) {
    if (typeof solutions[index] != "undefined") {
        return solutions[index];
    }

    return null;
}

deleteByIndex(index, solutions) {
    var sol = !ng.isUndefined(solutions[index]) ? solutions[index] : null;

    solutions.splice(index, 1);

    return sol;
}

addNew(name, id, solutions) {
    counter_new++;
    var solution = {
        id: id,
        isNew: "1",
        nombreSolucion: name,
        mostrar: "",
        solucion: "",
        notaJPPublica: null,
        notaReal: null,
        notaBinaria: null,
        resSolutionJPS: [],
        shouldExecute: "0",
    };
    solutions.push(solution);

    solution.nombreSolucion = solution.nombreSolucion.replace("{{counter}}", solutions.length);

    return solution;
}

has(solutions) {
    return Array.isArray(solutions) && solutions.length > 0;
}
```

Es mostren les operacions bàsiques del *SolutionService*.

10.2. Les tecnologies utilitzades

A continuació veiem breu descripció de les tecnologies, llenguatges de programació que he utilitzat per desenvolupar el projecte.

- **Apache Struts2**

Apache Struts 2 és un framework d'aplicacions web de codi obert per a desenvolupar aplicacions web Java EE. S'utilitza i estén l'API Java Servlet per animar als desenvolupadors a adoptar un model-vista-controlador (MVC) de l'arquitectura

- **AngularJS**

AngularJS és un framework MVC de JavaScript per al desenvolupament *Web FrontEnd* que permet crear aplicacions SPA *Single Page Applications*.

- ***SPA***

És una aplicació web o un lloc web que cap en una sola pàgina web amb l'objectiu de proporcionar una experiència d'usuari similar a la d'una aplicació d'escriptori.

- **SQL**

SQL l'abreviatura de *Structured Query Language* és un llenguatge estàndard de comunicació amb bases de dades relacionals.

- **PostgreSQL**

PostgreSQL és un software lliure que implementa un sistema de gestió de bases de dades relacional.

- **Java**

Java és un llenguatge de programació de propòsit general, concurrent, orientat a objectes que va ser dissenyat específicament per tenir tan poques dependències d'implementació com fos possible.

- **JavaScript**

JavaScript és un llenguatge de programació interpretat, dialecte de l'estàndard *ECMAScript*. Es defineix com orientat a objectes, 3 basat en prototips, imperatiu, dèbilment tipat i dinàmic.

- **HTML**

És un estàndard que serveix de referència del software que connecta amb l'elaboració de pàgines web en els seus diferents versions, defineix una estructura bàsica i un codi (anomenat codi HTML) per a la definició de contingut d'una pàgina web.

- **CSS**

CSS l'abreviatura de *Cascading Stylesheets* es un llenguatge de disseny gràfic per definir i crear la presentació d'un document estructurat escrit en un llenguatge de *markup*.

- **JSON**

JSON l'abreviatura de *JavaScript Object Notation* un estàndard obert basat en text dissenyat per a intercanvi de dades llegible per humans.

- **DTO**

DTO l'abreviatura de *Data Transfer Object* un patró de disseny fet servir per a transferir dades entre subsistemes d'aplicacions de software.

- **Flexbox**

Flexbox és una manera de disseny que permet col·locar els elements d'una pàgina perquè es comportin de forma predictible quan el disseny de la pàgina ha d'adaptar a diferents mides de pantalla i diferents dispositius.

- **Restangular**

Restangular és un servei que simplifica *AngularJS GET, POST, DELETE i UPDATE* sol·licituds amb un mínim de codi de client. És un ajustament perfecte per a qualsevol aplicació web que consumeix dades d'una API REST.

11. Proves

Les proves tenen com objectiu de garantir que l'aplicació es complexin tots els requisits demanats, per tant ens caldrà fer un pla de proves per validar tots els requisits funcionals i no funcionals que definits. A continuació, es mostra taules per cadascun dels requisits funcionals i no funcionals.

11.1. Proves del requisits funcionals

Per descriure les proves que s'han fet dels requisits funcionals (veure secció 7.2.1) es farà servir la plantilla següent:

*1	
Escenari	*2
Resultat	*3
Observacions	*4

***1:** Nom del requisit funcional
***2:** Descripció del la prova realitzada
***3:** Resultat de la prova
***4**: Observacions a l'hora de realització de la prova.

RF1. Consulta de correctors	
Escenari	Es comprova que es veuen la llista de corrector, es visualitzen les dades d'un corrector.
Resultat	100%
Observacions	Funcionalitat correcta.

RF2. Crear un corrector	
Escenari	Es comprova que es pot afegir un corrector, s'ha de donar error si aquest corrector té el mateix nom d'un altre que ja existeix.
Resultat	100%
Observacions	L'aplicació mostra el missatge d'error si afegeix un corrector que ja existeix.

RF3. Modificar el corrector	
Escenari	Es comprova que qualsevol modificació del corrector queda guardada un cop premut el botó "Desar".
Resultat	100%
Observacions	La modificació es queda guardada.

RF4. Eliminar el corrector	
Escenari	Es comprova que en prémer el botó “Eliminar”, s’elimina el corrector indicat.
Resultat	100%
Observacions	L’aplicació avisa que el corrector s’ha eliminat correctament.

RF5. Executar el corrector	
Escenari	Executa totes les qüestions que estan assignades al corrector seleccionat una darrera l’altre. Si s’executa correctament, la columna Resultat de l’execució posarà OK. Si s’executa i l’execució dóna error posarà ERROR. S’indica si s’ha executat totes les qüestions amb un comptar que surt al botó “Execució”.
Resultat	100%
Observacions	Funcionalitat correcta

RF6. Refrescar el corrector	
Escenari	Es comprova que en prémer el botó “Refrescar”, es refresca el corrector amb les dades actualitzades.
Resultat	100%
Observacions	Funcionalitat correcta.

RF7. Modificar el fitxer adjunt	
Escenari	Es comprova que en prémer el botó “Pujar” modifica el que hi havia, i ha de poder veure el contingut del fitxer que acaba de pujar.
Resultat	100%
Observacions	Ha mostrat el contingut corresponent correctament.

RF8. Eliminar el fitxer adjunt	
Escenari	Es comprova que en prémer el botó “Eliminar”, la qüestió ha de quedar sense fitxer adjunt.
Resultat	100%
Observacions	La qüestió ha quedat sense fitxer adjunt, i el botó de “Veure” el fitxer adjunt s’ha quedat desactivat.

RF9. Guardar el fitxer adjunt	
Escenari	Es comprova que en prémer el botó de “Veure”, es descarrega el fitxer adjunt i es pot veure el seu contingut.

Resultat	100%
Observacions	El contingut es mostra correctament.

RF10. Generar el fitxer adjunt	
Escenari	Es comprova que en prémer el botó “Generació automàtica”, es genera el fitxer adjunt per a les qüestions tipus JDBC.
Resultat	100%
Observacions	La generació s’ha realitzat correctament.

RF11. Generar llistats	
Escenari	Es comprova que es permet seleccionar els camps a llistar, si un joc de proves està seleccionat, es mostraran les seves comprovacions, un cop premut el botó “Generar”, es descarrega un fitxer amb contingut que correspon les opcions que ha seleccionat.
Resultat	100%
Observacions	El fitxer s’ha descarrega correctament.

RF12. Copiar la qüestió a altres bases de dades (CopyBD)	
Escenari	Es comprova que es permet connectar a altra base de dades un cop tinguin les dades introduït correctament, i verifica que en aquesta base de dades tingui la qüestió copiada.
Resultat	100%
Observacions	La qüestió s’ha copiat a altra base de dades.

RF13. Executar la qüestió	
Escenari	Es comprova que es pot executar la qüestió, i el sistema ha de capaç de donar la solució de cada joc de proves de la qüestió, i mostrar el log d’execució.
Resultat	100%
Observacions	S’ha mostrat el log d’execució un cop executat la qüestió, i la qüestió s’ha passat a l’estat resolta.

RF14. Canviar l’ordre de joc de proves o comprovació	
Escenari	Es comprova que es pot ordenar el joc de proves o comprovació, activa o desactiva el botó “Pujar”, “Baixar” segon el joc de proves o comprovació que selecciona.
Resultat	100%
Observacions	Els botons d’ordenar han funcionat correctament.

RF15. Consulta de solucions	
Escenari	Es comprova que es visualitzen les solucions de la qüestió.
Resultat	100%
Observacions	Funcionalitat correcta

RF16. Crear la solució	
Escenari	Es comprova que es pot crear una nova solució, en canvi no ho pot si ja existeix una solució amb el mateix nom .
Resultat	100%
Observacions	Es dóna l'error quan crea una solució amb el mateix nom que l'altra.

RF17. Modificar la solució	
Escenari	Es comprova que qualsevol modificació queda guardada un cop premut el botó "Desar".
Resultat	100%
Observacions	Les modificacions de la solució s'han guardat correctament.

RF18. Eliminar la solució	
Escenari	Es comprova que en prémer el botó "Eliminar", s'elimina la solució indicada.
Resultat	100%
Observacions	L'aplicació avisa que la solució s'ha quedat eliminada correctament.

RF19. Executar la solució o diverses solucions en bloc	
Escenari	Es comprova que es prémer el botó "Executar" o "Executar tot", s'executa la solució indicada o totes les solució de la qüestió, i per cada solució es guarden els seus resultats de l'execució a la base de dades.
Resultat	100%
Observacions	Per cada solució l'aplicació es mostra el seu log d'execució, i els seus resultats de l'execució s'han guardat.

RF20. Modificar el zip genèric	
Escenari	Es comprova que es pot pujar un fitxer d'extensió ".zip", i es mostren el contingut d'aquest fitxer.
Resultat	100%
Observacions	Funcionalitat correcta.

RF21. Consultar resultats de la qüestió	
Escenari	Es comprova que per cada possible resposta d'una qüestió, els jocs de proves/comprovacions que la qüestió passa favorablement o no passa, i podran consultar informacions de resultats de la qüestió provinent del Moodle-LearnSQL.
Resultat	100%
Observacions	Funcionalitat correcta.

RF22. Duplicar un joc de proves o una comprovació	
Escenari	Es comprova que en duplicar el joc de proves o comprovació, cal que es dupliquin totes les seves dades i totes les seves comprovacions si es duplica un joc de proves.
Resultat	100%
Observacions	S'ha duplicat correctament.

11.2. Proves del requisits no funcionals

Per descriure les proves que s'han fet de satisfacció dels requisits no funcionals (veure secció 7.3.1) es fa servir la plantilla següent:

Descripció	*1
Criteri de satisfacció	*2
Resultat	*3
Observacions	*4

***1:** Descripció del requisit no funcional
***2:** Descripció del la prova de satisfacció
***3:** Resultat de la prova
***4:** Observacions durant o després de realització de la prova

Descripció	**RNF1:** El disseny serà atractiu i d'ús senzill per a tots els usuaris.
Criteri de satisfacció	El sistema compleix les recomanacions d'estil definides per *Google Material Design*.
Resultat	100%
Observacions	S'utilitza menú lateral per llistat tots els entitats (qüestió, categoria, temàtica etc..), i disposa una barra de navegació.

Descripció	**RNF2:** L'aplicació compleix els estàndards ISO d'usabilitat.
Criteri de satisfacció	Es comprova que es compleix l'estàndard ISO-9241
Resultat	100%
Observacions	L'usuari pot utilitzar de forma correcte sense coneixements previs.

Descripció	**RNF3:** El llenguatge ha de ser correcte, comprensible, concís i respectuós.
Criteri de satisfacció	Una persona amb un alt coneixement en l'idioma que mostra l'aplicació verificarà que el llenguatge és correcte o no.
Resultat	100%
Observacions	El llenguatge utilitzat en la informació s'entén de forma correcte.

Descripció	**RNF5:** L'aplicació ha de ser capaç de donar respostes rapides a les peticions.
Criteri de satisfacció	El sistema ha de respondre a les peticions bàsiques ràpidament, i si la petició requereix més temps, s'ha de bloquejar la pantalla de l'aplicació.
Resultat	50%
Observacions	El temps mitja de respondre peticions com crear, modificar, eliminar l'entitat s'ha tardat un màxim de 2-3s, les peticions com executar la qüestió s'ha tardat un màxim de 4s. Sobretot el temps que triga bastant per tal poder carregar totes qüestions en l'entorn de producció. En resum, l'eficiència de l'aplicació no del està tot assolit.

Descripció	**RNF6:** L'aplicació ha der ser capaç d'emmagatzemar una ultima versió de les dades obtingudes.
Criteri de satisfacció	L'aplicació client ha de mantenir una copia de les dades que estava treballant quan algun moment hi ha una desconnexió del servidor.
Resultat	100%
Observacions	Permet treballar l'aplicació de forma *off-line*.

Descripció	**RNF7:** El sistema ha de ser fàcilment mantenible.
Criteri de satisfacció	Per reduir el cost de manteniment, el desenvolupament del sistema es farà seguint el disseny en capes per tenir menys dependències possibles.
Resultat	100%
Observacions	S'ha desenvolupat l'aplicació client i l'aplicació servidor mitjançant capes.

Descripció	**RNF8:** L'aplicació serà compatible amb navegador com Chrome, Firefox, IE etc..
Criteri de satisfacció	L'aplicació ha de funcionar en qualsevol dispositius que tinguin un navegador amb últimes actualitzacions.
Resultat	100%
Observacions	L'aplicació es pot entrar mitjançant navegador com Chrome, Firefox i IE, i dispositius com Smartphone.

Descripció	**RNF9:** Els desenvolupadors tindran accés al codi del sistema i els usuaris tindran accés a l'aplicació un cop hagi fet logueig del sistema.
Criteri de satisfacció	Només permet entrar els usuaris que hagin introduït l'usuari, la contrasenya, i el nom de base de dades de forma correcta.
Resultat	100%
Observacions	No s'ha pogut entrar a l'aplicació si no ha fer l'autenticació correcta.

Descripció	**RNF10:** El codi font del l'aplicació client i l'aplicació servidor han de tenir una còpia de seguretat.
Criteri de satisfacció	El codi font està emmagatzemat en un repositori remot.
Resultat	100%
Observacions	Els dos projectes estan en repositoris de *Bitbucket.*

11.3. Sumari de les proves

En la taula següent es mostra un sumari dels resultats de les proves de satisfacció obtinguts.

Requisits	**Resultat proves**
RF1	100%
RF2	100%
RF3	100%
RF4	100%
RF5	100%
RF6	100%
RF7	100%
RF8	100%
RF9	100%
RF10	100%
RF11	100%
RF12	100%
RF13	100%
RF14	100%
RF15	100%
RF16	100%
RF17	100%
RF18	100%
RF19	100%
RF20	100%
RF21	100%
RF22	100%
RNF1	100%
RNF2	100%
RNF3	100%
RNF5	50%
RNF6	100%
RNF7	100%
RNF8	100%
RNF9	100%
RNF10	100%

12. Resultats

En aquest capítol es reflexiona sobre els resultats obtinguts, les principals assignatures de l'especialitat que han servit per aquest projecte, les competències tècniques, les conclusions que s'han extret després d'un llarg termini del desenvolupament i el possible treball futur a realitzar.

12.1. Coneixements de les assignatures de l'especialitat

Per poder realitzar el projecte, els coneixements de les assignatures que he fet servir són els següents:

- **Disseny base de dades (DBD)**

Encara que el projecte és una extensió d'una aplicació ja existent, entre les noves funcionalitats que s'ha desenvolupat en l'ATW2 hi ha funcionalitats que han requerit afegir noves taules a la base de dades de qüestions. Per afegir les noves taules ha estat necessari pensar en com era més adequat de dissenyar les noves taules.

- **Aplicacions i Servei Web (ASW)**

L'ATW2 és una eina de plataforma web, per tant els coneixements que he après en assignatures de l'especialitat com protocols, llenguatge web, i sobretot sobre disseny arquitectònic d'aplicacions web, serveis web, usabilitat, han estat necessaris a l'hora de desenvolupar el projecte.

- **Enginyeria de Requisits (ER)**

En l'assignatura d'ER es va aprendre a identificar els objectius d'un nou projecte a desenvolupar, i per cada objectiu veure com es podria assolir mitjançant requisits funcionals i no funcionals. Això és el que també s'ha fet en el plantejament del projecte per desenvolupar l'ATW2 i garantir l'èxit del projecte.

- **Gestió de Projecte de Software (GPS)**

En l'assignatura GPS es va obtenir coneixements sobre metodologies de desenvolupament de projectes, sobre planificació de projectes, i sobre identificació de riscos. Aquests coneixements han estat útils per a gestionar el projecte durant tot el seu desenvolupament.

Així doncs aquest projecte s'adequa a les característiques de l'especialitat d'Enginyeria del Software, ja que en ell s'ha desenvolupat un sistema software, en aquest cas, un sistema software de plataforma web. Per desenvolupar aquest sistema software s'ha seguit un procés en el que

s'han fet activitats com identificació de requisits, especificació, disseny, implementació, i proves d'acceptació. El sistema software resultant té per objectiu el millorar la gestió i la qualitat de treball dels *stakeholders* o persones interessades en el sistema LearnSQL. Tot l'anterior fa que el projecte encaixi en les competències de l'especialitat d'Enginyeria del Software.

12.2. Competències tècniques

CES1.1: Desenvolupar, mantenir i avaluar sistemes i serveis software complexos i/o crítics. **[En profunditat]**

En quan a desenvolupament, aquest és un projecte de desenvolupament d'un sistema software, dins del projecte es desenvoluparan serveis web que encapsularan la banda del servidor del sistema software. En quan a manteniment, aquest projecte va partir d'una primera versió d'aplicació per la que hi ha parts que cal modificar i parts que cal estendre, per tant es pot entendre que es fa manteniment de l'ATW per a convertir-la en la nova versió ATW2. En quan a avaluació, la intenció és en acabar el projecte avaluar el sistema software resultant per veure que compleix els requisits esperats passant les proves d'acceptació en l'entorn de producció.

CES1.3: Identificar, avaluar i gestionar els riscos potencials associats a la construcció de software que es poguessin presentar. **[Una mica]**

Els riscos s'han identificat en el document resultant de cursar l'assignatura GEP. Durant el projecte ha estat necessari pensar en com gestionar els riscs potencials que puguin aparèixer.

CES1.7: Controlar la qualitat i dissenyar proves en la producció de software. **[Una mica]**

A cada iteració es faran proves unitàries a nivell individual per part del desenvolupador, i es va fer una reunió per mostrar funcionalitats que hagin desenvolupat amb la directora, així la directora s'ha pogut fer les proves d'acceptació i fins tot s'ha pogut donar recomanacions per millorar el funcionament.

CES2.1: Definir i gestionar els requisits d'un sistema software. **[Una mica]**

Tal com s'ha dit abans, mentre s'ha cursat l'assignatura de GEP s'ha hagut de identificar els objectius de les parts interessades en el projecte, i per cada objectiu s'ha vist quins requisits funcionals i no funcionals que ha de satisfer l'ATW2 un cop acabada per garantir l'èxit del projecte. Durant el projecte ha estat necessari pensar en com gestionar aquests requisits per tal d'assegurar el seu compliment. En acabar el projecte s'ha hagut de validar la satisfacció de tots els requisits identificats, tant funcionals com no funcionals.

12.3. Conclusions

Al llarg del desenvolupament d'aquest projecte, tot i que he trobat molt problemes que han anat sorgint, he pogut superat reptes com la configuració de l'entorn de desenvolupament, el desconeixement de les noves tecnologies necessàries pel projecte, el descobriment i comprensió de la implementació de l'AT i ATW de partida, el desenvolupament d'una aplicació partint d'un disseny i implementació existents. He fet un gran esforç, he dedicat molt temps per superar aquest reptes, però ha sigut una gran experiència.

Gràcies a aquest projecte, he pogut aprendre molts coneixements que resulten importants i que crec que em serviran tant professionalment com personalment. Gràcies a la base sòlida construïda per l'Albert Boada en desenvolupar l'ATW, he pogut desenvolupat un projecte que servirà als professors de les assignatures de base de dades per millorar a l'hora de gestionar i analitzar les qüestions, espero que sigui una eina que pugui ser útil. Per altra banda, el fet de partir d'una aplicació ATW construïda pensant per a facilitar la seva mantenibilitat i extensibilitat, he aprés també com ho hauria de fer jo per fer el mateix en un futur.

Finalment volia donar les gracies a totes les persones que m'han ajudat durant el desenvolupament del projecte, sobre tot la directora del projecte, Carme Quer, que ha estat al meu costat per ajudar i resoldre els problemes que he tingut.

12.4. Treball futur

Aquest projecte té una data límit especifica, sempre queden coses per fer i si en un futur. Bàsicament consisteix en dues principals millores, la primera millora és optimitzar el fetch inicial, ja que quan l'aplicació es connecta a la base de dades de la producció, es pot arribar fins i tot més de 1000 registres de qüestions, el rendiment del fetch inicial es veu deteriorat a mesura que augmenti el nombre de registres.

Una possible solució És dissenyar un fetch inicial de forma *lazy loading*, és a dir, en compte de carregar totes les dades de qüestions, només carreguen les dades crítiques, com l'identificador, el nom, l'estat de la qüestió, ja que un cop entrat a l'aplicació, primerament, només mostra el menú lateral amb les dades crítiques de qüestions, posteriorment si l'usuari vol gestionar una qüestió, es farà una petició GET amb identificador d'aquesta qüestió al servidor per carregar les dades necessàries. L'altra solució és plantejar migrar l'aplicació client a *AngularJS 2*, que s'ha optimitzat tot el procés intern de detecció de canvis, van dissenyar *ultra fast change detection* y estructures immutables, per poder treballar amb quantitats grans de dades de manera més eficient i ràpida. De cara al futur, a mesura que van utilitzant l'aplicació, si necessiten noves funcionalitat, es caldrà desenvolupar una tercera versió de l'aplicació.

13. Referències

[1] Memòria de desenvolupament d'una interfície web per a una aplicació per a la gestió de preguntes de LearnSQL
<https://upcommons.upc.edu/handle/2117/90580>
[Consulta: 21 de febrer de 2017]

[2] Memòria de desenvolupament de noves funcionalitats i millores de l'aplicació de gestió de qüestions de LearnSQL
<https://upcommons.upc.edu/handle/2099.1/9753>
[Consulta: 21 de febrer de 2017]

[3] Learning Environment for Automatic Rating Notions of SQL (LearnSQL)
<http://www.upc.edu/learn-sql/en/>
[Consulta: 21 de febrer de 2017]

[4] AngularJS
<https://angularjs.org/>
[Consulta: 1 de març de 2017]

[5] Apache Struts 2
<http://struts.apache.org/>
[Consulta: 7 de març de 2017]

[6] Enginyeria de Serveis i Sistemes d'Informació (ESSI)
<https://www.essi.upc.edu/ca>
[Consulta: 8 de març de 2017]

[7] Bases de Dades Relacionals
<https://ioc.xtec.cat/materials/FP/Materials/2252_DAM/DAM_2252_M02/web/html/WebContent/u3/a1/continguts.html>
[Consulta: 14 de març de 2017]

[8] Moodle
<https://moodle.org/?lang=ca>
[Consulta: 15 de març de 2017]

[9] Sistema de Gestió de Bases de Dades (SGBD)
<http://www.alegsa.com.ar/Dic/sgbd.php>
[Consulta: 15 de març de 2017]

[10] Alberto Abelló, Xavier Burgués, M. José Casany, Carme Martín, Carme Quer, M. Elena Rodríguez, Oscar Romero, Toni Urpí: A Software Tool for E-Assessment of Relational Database Skills. International Journal of Engineering Education. Vol 32, No. 3(A), 2016.
[Consulta: 20 de març de 2017]

[11] Bitbucket
<https://bitbucket.org/>
[Consulta: 21 de març de 2017]

[12] JDBC
<https://es.wikipedia.org/wiki/Java_Database_Connectivity>
[Consulta: 21 de març de 2017]

[13] Estudi de remuneració de *PagePersonnel* de l'any 2016
<http://www.pagepersonnel.es/sites/pagepersonnel.es/files/er_tecnologia16.pdf>
[Consulta: 25 de març de 2017]

[14] Informe de Sostenibilitat del TFG
<http://www.fib.upc.edu/fib/estudiar-enginyeria-informatica/treball-final-grau/mainColumnParagraphs/013/document/2016_SyCS_GEP_v3_catala.pdf >
[Consulta: 26 de març de 2017]

[15] Single Page Application
<https://www.codeschool.com/beginners-guide-to-web-development/single-page-applications>
[Consulta: 1 d'abril de 2017]

[16] JSON
<http://www.json.org/>
[Consulta: 2 d'abril de 2017]

[17] Model-Vista-Controlador
<https://msdn.microsoft.com/en-us/library/ff649643.aspx>
[Consulta: 3 d'abril de 2017]

[18] Apache Struts 2
<https://struts.apache.org/>
[Consulta: 3 de maig de 2017]

[19] Patró Front Controller
<https://en.wikipedia.org/wiki/Front_controller>
[Consulta: 3 de maig de 2017]

[20] DTO
<https://msdn.microsoft.com/en-us/library/ff649585.aspx>
[Consulta: 4 de maig de 2017]

[21] REST API
<https://stackoverflow.com/questions/671118/what-exactly-is-restful-programming>
[Consulta: 7 de maig de 2017]

[22] Flexbox
<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
[Consulta: 8 de maig de 2017]

[23] Material Desgin
<https://material.io/guidelines/usability/accessibility.html#>
[Consulta: 15 de maig de 2017]

[24] International Organization for Standardization
<https://en.wikipedia.org/wiki/List_of_International_Organization_for_Standardization_standards>
[Consulta: 21 de maig de 2017]

[25] HTTP
<https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol>
[Consulta: 2 de juny de 2017]

[26] Restangular
<https://github.com/mgonto/restangular>
[Consulta: 2 de juny de 2017]

[27] HTML

<https://www.w3schools.com/html/>

[Consulta: 15 de juny de 2017]

[28] AngularJS 2
<https://angular.io/>
[Consulta: 17 de juny de 2017]