



InfoJOR

Carles Canellas Crusellas

Grau d'enginyeria informàtica

Enginyeria del programari

Consultor: Oriol Martí Girona

Responsable: Santi Caballé Llobet

Lliurament: 01/2017



Aquesta obra està subjecta a una llicència de
[Reconeixement-NoComercial-SenseObraDerivada
3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Fitxa del treball final

Títol del treball:	<i>InfoJOR</i>
Nom de l'autor:	<i>Carles Canellas Crusellas</i>
Nom del consultor/a:	<i>Oriol Martí Girona</i>
Nom del PRA:	<i>Santi Caballé Llobet</i>
Data de lliurament (mm/aaaa):	<i>01/2017</i>
Titulació o programa:	<i>Grau d'enginyeria informàtica</i>
Àrea del Treball Final:	<i>Enginyeria del programari</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Web, PHP, Informes</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>Aquest projecte és una aplicació Web per gestionar d'una forma integral els informes trimestrals i finals que es fan a l'alumnat d'educació infantil i primària. Està programada en PHP a la banda del servidor i Javascript a la del client, i opera sobre una base de dades MySQL.</p> <p>El curs 2016-2017, l'escola Josep Orriols i Roca de Moià, va decidir adaptar les valoracions trimestrals i finals dels alumnes a les proposades per la Generalitat de Catalunya.</p> <p>Aprofitant aquest canvi, es va decidir modificar la forma d'entrar les valoracions per part dels mestres. L'anterior sistema presentava alguns inconvenients importants, com ara: poca flexibilitat per entrar les dades, obligació de tenir instal·lat MSAccess, problemes durant el procés de sincronització, problemes d'impressió a doble pàgina, format de sortida poc elegant i intel·ligible, etc.</p> <p>Com que no es va trobar cap alternativa existent, es va decidir programar un nou sistema que complís aquests requisits: entrada de les valoracions des de qualsevol lloc i des de qualsevol dispositiu, barat de mantenir, pràctic per imprimir, segur, flexible i fàcil d'administrar.</p> <p>Per la consecució d'aquest projecte s'ha seguit una metodologia àgil.</p>	
Abstract (in English, 250 words or less):	

This project is a Web application for managing the trimester and final reports of infant and primary education students. It is programmed in PHP on the server side and in JavaScript on the client side, and operates on a MySQL database.

During the academic year 2016-2017, the school Josep Oriols i Roca, in Moià, decided to adapt the trimester and course ratings to the ones proposed by the Catalan Government.

Taking advantage of this change, it was decided to improve the system that the teachers used to enter the ratings. The previous application had some significant drawbacks, such as: lack of flexibility to enter the data, installation of MSAccess required, problems during synchronization, problems for duplex printing, inelegant and little understandable output, etc.

Because there was no alternative found, it was decided to program a new system that met this requirements: input ratings from anywhere and on any device, inexpensive to maintain, practical for duplex printing, secure, flexible and easy to manage.

This project has been developed following an agile and iterative approach.

Índex de continguts

Fitxa del treball final.....	iii
1.Introducció.....	8
1.1.Context i justificació del Treball.....	8
1.2.Objectius del Treball.....	10
1.3.Enfocament i mètode seguit.....	10
1.4.Planificació del Treball.....	12
1.5.Breu sumari de productes obtinguts.....	13
1.6.Breu descripció dels altres capítols de la memòria.....	14
2.Tasques prèvies.....	16
2.1.Recollida inicial de requisits.....	16
2.2.Alternatives.....	17
2.3.Avaluació dels riscos.....	17
2.4.Pressupost.....	18
3.Anàlisi.....	19
3.1.Recollida de requisits.....	19
Stakeholders.....	19
Entrevistes amb els Stakeholders.....	20
Qüestionari.....	23
3.2.Requisits.....	25
Requisits no funcionals.....	25
Casos d'ús (nivell alt).....	25
3.3.Casos d'ús detallats.....	28
3.4.Model del domini.....	41
Estructura bàsica de l'escola.....	41
Personal associat a l'escola.....	41
Estructura detallada de l'escola.....	42
Informes.....	43

Avaluacions.....	44
3.5.Model de casos d'ús.....	45
4.Arquitectura.....	46
4.1.Arquitectura física.....	46
4.2.Arquitectura lògica.....	47
Presentació.....	47
Serveis.....	48
DAO.....	48
5.Disseny.....	51
5.1.Interfície d'usuari.....	51
Navegació.....	51
Disseny.....	51
5.2.Operacions de sistema.....	53
5.3.Diagrames d'interacció.....	57
Registrar-se al sistema (login).....	57
Capçalera.....	59
Pàgina principal (main).....	60
Pàgina d'entrada d'avaluacions (evaluate).....	61
5.4.Diagrames de classes.....	64
Diagrama d'entitats.....	64
Models de vista.....	64
Serveis.....	66
5.5.Patrons de disseny utilitzats.....	67
5.6.Disseny de la base de dades.....	67
Disseny lògic.....	67
6.Implementació.....	71
6.1.Estàndards utilitzats.....	71
6.2.Entorn de treball.....	72
Infraestructura utilitzada.....	72

Eines externes.....	72
Instal·lació i configuració del programari.....	72
6.3.Estructura de directoris.....	73
6.4.Consideracions.....	74
Espais de noms.....	74
Impressió dels informes.....	75
7.Instal·lació.....	76
7.1.Requisits del sistema.....	76
Maquinari.....	76
Programari.....	76
7.2.Instal·lació i execució.....	77
8.Conclusions.....	78
8.1.Tasques pendents.....	78
8.2.Objectius assolits.....	79
8.3.Principals dificultats.....	79
8.4.Millores i ampliacions.....	80
8.5.Visió de futur.....	81
9.Fonts d'informació.....	82
9.1.Bibliografia.....	82
9.2.Webs visitades:.....	83
10.Glossari.....	85
11.Annexos.....	86

1. Introducció

1.1. Context i justificació del Treball

Aquest projecte està justificat per dues raons bàsiques:

- Realització del projecte de final de grau d'enginyeria informàtica.
- Canvi de sistema d'entrada de valoracions trimestrals de l'escola.

Des de fa uns quants anys treballo com a responsable de manteniment informàtic de l'escola Josep Orriols i Roca de Moià. Bàsicament la meva feina ha estat la de solucionar els diferents problemes que sorgeixen el dia a dia, assessorar en la compra i configuració d'equips informàtics, instal·lació de programari, configuració de xarxes...

Cada trimestre i a final de curs, els mestres de l'escola han de fer una valoració dels seus alumnes en les diferents àrees: llengua, matemàtiques, música... Aquests informes s'han d'imprimir i enviar-los als pares. Els darrers anys, s'utilitzava una base de dades MSAccess que havia pro-

gramat el coordinador d'informàtica de l'escola. Si bé era un sistema que permetia sortir del pas, presentava una sèrie de problemes tant a nivell d'entrada de dades, administració i impressió:

- Cada classe tenia la seva pròpia base de dades, i només era vàlida per un curs. Això provocava que cada curs l'administrador hagués de copiar la base de dades de cada classe del curs anterior i buidar-ne les dades per poder-les entrar des de zero. A més, no es podia tenir una visió general de l'escola ni dels diferents cursos pels que havien passat els alumnes.
- Els mestres necessitaven accés directe a la base de dades. Per tant, si volien fer els informes des de casa, havien de copiar-la en un *pendrive*, entrar les valoracions, i sincronitzar la base de dades que s'havien emportat amb la de l'escola. Aquest procés, a banda de ser farragós, generava alguns problemes en la sincronització. D'altra banda, l'intercanvi de llapis de memòria entre ordinadors implicava problemes a causa dels virus.
- La impressió a doble cara de tota una classe era poc pràctica, ja que en el cas que l'informe d'un alumne acabés en una pàgina imparell, el del següent alumne començava en pàgina parell, per tant la primera pàgina s'imprimia darrera la mateixa pàgina que l'alumne anterior.
- El format d'impressió era estèticament poc agradable, mal maquetat i poc clar d'interpretar.

El curs 2016-2017, l'escola va decidir adoptar el sistema de valoracions que havia suggerit la Generalitat en els currículums d'educació infantil i primària el 2015, i que encara estava molt poc implementat en les escoles. Aprofitant aquest canvi, es va decidir conjuntament amb l'equip directiu d'implementar un nou sistema de valoracions que s'adeqüés al nou currículum i que solventés els problemes de l'anterior aplicació.

Aquest projecte es va engegar l'octubre de 2016. Les valoracions del primer trimestre calia començar a entrar-les a finals de novembre del mateix any. Degut a la manca de temps per tenir tota l'aplicació acabada, es va decidir fer una implementació per fases.

1.2. Objectius del Treball

Els objectius del present treball són:

- Crear una aplicació web seguint la metodologia de l'enginyeria del programari, utilitzant al màxim les eines proposades per aquesta disciplina.
- Oferir a l'escola una eina pràctica, flexible, robusta, segura, actualitzable i barata de mantenir, que serveixi als mestres per entrar les valoracions trimestrals i finals dels alumnes, i alhora uns informes adaptats als nous requeriments proposats per la Generalitat i que siguin fàcils de comprendre pels pares.
- A nivell personal, aprofundir en els processos de creació d'aplicacions en el camp de l'enginyeria del programari, així com consolidar i aprofundir els meus coneixements en les tecnologies web: HTML, CSS, Javascript, PHP i MySQL.

Si bé és cert que PHP no ofereix les mateixes prestacions que Java a l'hora de programar aplicacions orientades a objectes, cal tenir en compte que és molt més barat llogar un servidor per allotjar una aplicació PHP + MySQL que una Java. Això obre una ventall de possibilitats en la implementació de projectes que requereixin un baix cost de manteniment, com és el present cas. D'altra banda, és una tecnologia en constant evolució que en cada versió presenta millores notables.

1.3. Enfocament i mètode seguit

Degut als requisits de temps per poder lliurar la primera fase del projecte, he optat per un mètode àgil. A més a més, personalment crec que és millor utilitzar aquest mètode que no pas la programació en cascada. Tenint en compte que és el primer projecte que enfoco des d'aquesta perspectiva, vaig valorar la seva execució com una forma d'iniciació i aprenentatge d'aquesta metodologia.

A part de les diferents assignatures, treballs i materials dels que dispo durant la meua etapa com a estudiant de GEI a la UOC, ha estat molt útil el llibre de Craig Larman *Applying UML and Patterns*, que afronta la part d'anàlisi i disseny de projectes des d'un punt de vista iteratiu.

Els diferents passos seguits durant el desenvolupament han estat:

- Entrevistes i qüestionaris amb l'equip directiu i el coordinador d'informàtica de l'escola per definir els requisits de l'aplicació.
- Presentació del pressupost a l'escola per la seva aprovació.
- Anàlisi.
- Disseny de la fase inicial.
- Desenvolupament de la fase inicial.
- Implementació i proves de la fase inicial.
- Disseny de la resta de l'aplicació.
- Desenvolupament de la resta de l'aplicació.
- Implementació i proves de la resta de l'aplicació.

Deixant de banda les fases de recollida de requisits i anàlisi, que es van fer gairebé per complet a l'inici del projecte, la resta de fases s'han anat intercalant. Així doncs, la programació no es va fer després de finalitzar tot el disseny, sinó que, en el moment que una part del disseny es donava per bo (o gairebé), es començava la programació d'aquella part (seguint el mètode àgil).

Durant tot el procés he hagut d'intercalar altres feines de recerca que em permetessin enfocar de la millor manera el projecte, com: buscar i provar un framework ORM per PHP similar a Hibernate, buscar un sistema de plantilles, etc.

Al principi vaig estar buscant frameworks de programació en PHP que em poguessin facilitar la feina: [Symfony](#), [Laravel](#)..., però els vaig acabar descartant perquè vaig creure més convenient dur a terme tot el projecte sense dependre d'una plataforma específica i controlant personalment la interacció entre les diferents capes. A més a més, normalment els controladors d'aquests frameworks tenen assignades moltes responsabilitats, que no crec que sigui una bona pràctica.

Voldria comentar que, després de fer algunes proves, vaig decidir utilitzar el framework [Doctrine](#) per fer el mapatge entre objectes i relacions. Com a avantatge principal, ha estat utilitzat ja fa temps per *Symfony*, la qual cosa li suposa una bona fiabilitat. També ofereix unes prestacions interessants, com facilitar en el mapatge, possibilitat d'herència, canvis automàtics en

la base de dades a partir dels canvis en les relacions...

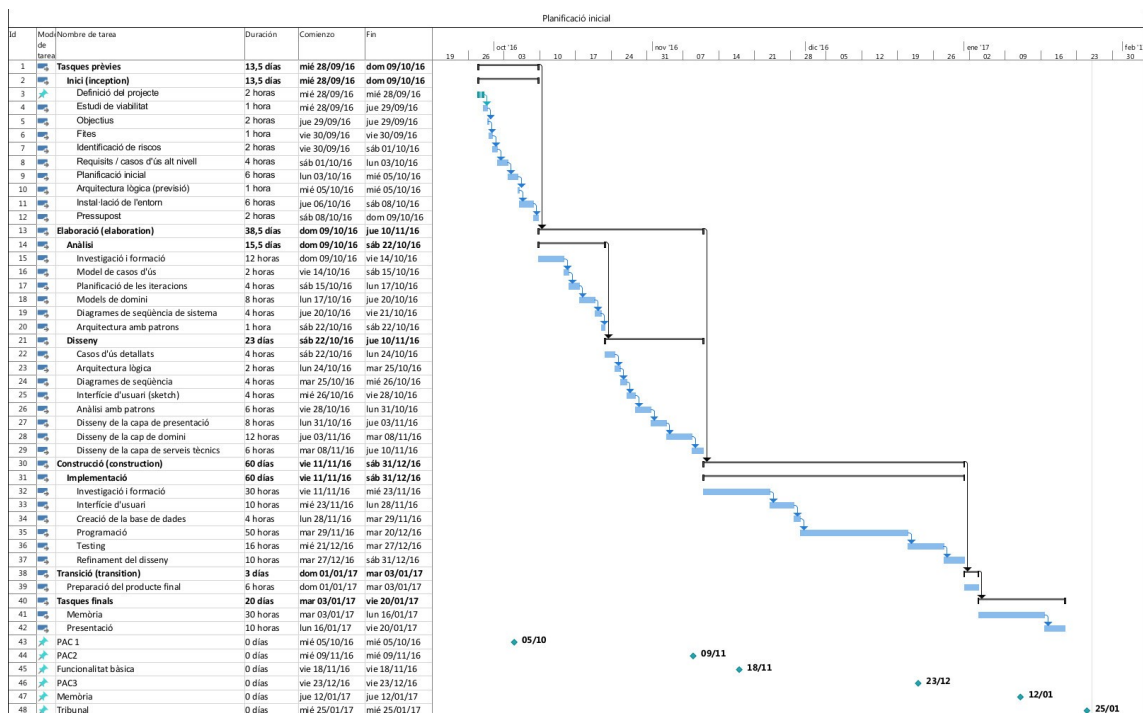
Pel que fa al sistema de plantilles, he optat per treballar amb [Transphorm](#), un projecte iniciat per Tom Butler i que té la característica d'utilitzar marques similars a CSS per fer el mapatge entre marques HTML i dades dinàmiques.

El codi font s'ha escrit seguint bàsicament els estàndards proposats a [PHP-FIG](#).

L'arquitectura utilitzada ha estat la de client-servidor, aprofitant que l'escola ja disposava d'un domini propi.

1.4. Planificació del Treball

Per la planificació s'han tingut en compte les dates imposades tant per part de l'escola com per part de la UOC.



Les dates imposades per l'escola eren:

- Entrada de valoracions i impressió dels informes del primer trimestre a partir de finals de novembre.
- Entrada de valoracions finals i impressió dels informes a partir de finals de maig.

Així doncs, es va decidir tenir la primera fase llesta la penúltima setmana de novembre, per tal que els mestres poguessin fer proves durant una setmana. Al final d'aquesta setmana, es van realitzar els canvis convenients a l'aplicació i es va eliminar totes les valoracions de prova per tal de poder començar a entrar les definitives a partir del dia 30 de novembre.

Les dades imposades per la UOC eren:

- 5/10/2016 Lliurament de la PAC1 (pla de treball)
- 9/11/2016 Lliurament de la PAC2 (anàlisi)
- 23/12/2016 Lliurament de la PAC3 (disseny)
- 12/01/2017 Lliurament de la memòria i la presentació
- 25/01/2017 Procés del tribunal

En la mesura que es pugui, per la data de la presentació de la memòria es preveu tenir l'aplicació pràcticament finalitzada.

1.5. Breu sumari de productes obtinguts

El producte obtingut és una aplicació web composta d'una sèrie d'arxius web (PHP, CSS, Javascript...), arxius de configuració de l'aplicació i una base de dades MySQL en forma d'un sql de text. Tots aquests arxius, copiats convenientment en un servidor, i amb l'arxiu sql traspassat a una base de dades, permeten fer funcionar completament l'aplicació.

A la base de dades es deixaran unes dades de mostra anònimes per tal de poder fer proves. Així mateix, també es penjarà tota l'estructura en un servidor web per tal que es puguin provar convenientment les funcionalitats sense necessitat d'instal·lació.

1.6. Breu descripció dels altres capítols de la memòria

2. Tasques prèvies: tasques necessàries que cal fer abans d'iniciar el projecte:

- Recollida inicial de requisits per tenir una idea de l'aplicació que vol el client.
- Cerca d'alternatives existents.
- Avaluació dels riscos més importants que pot comportar el projecte.
- Pressupost.

3. Anàlisi: tasques que formen part de l'anàlisi de la realitat i de la seva modelització:

- Recollida exhaustiva de requisits
- Casos d'ús
- Model del domini
- Model de casos d'ús

4. Arquitectura: arquitectura física i lògica (diferents components del sistema i interaccions entre ells).

5. Disseny: tasques que formen part del disseny de l'aplicació en totes les seves vessants:

- Interfície d'usuari
- Operacions de sistema
- Diagrames d'interacció
- Diagrames de classes
- Disseny de la base de dades

6. Implementació: algunes qüestions que s'han tingut en compte a l'hora d'implementar l'aplicació, com els estàndards utilitzats i l'entorn de treball utilitzat. També d'altres orientades a comprendre-la millor, com l'explicació de l'estructura de directoris.

7. Instal·lació: requisits mínims i instruccions per procedir a la instal·lació.

8. Conclusions: valoració de com ha anat el projecte des de diversos punts de vista, ja sigui

per les funcionalitats que resta implementar com pels objectius assolits. Possible recorregut de l'aplicació en un futur.

9. Fonts d'informació: relació dels llibres i webs consultats per dur a terme el projecte.

2. Tasques prèvies

2.1. Recollida inicial de requisits

Per tal de poder elaborar un pressupost i comprovar la viabilitat del projecte, va ser necessari fer una primera recollida de requisits.

Per fer-ho, em vaig basar en les següents fonts:

- L'aplicació que utilitzaven fins el moment.
- Els models d'informes que proposava el coordinador d'informàtica i que encara estaven en una fase inicial. Aquest aspecte era important, ja que els models finals podien fer variar l'estructura de l'aplicació.
- Diverses reunions mantingudes amb el coordinador d'informàtica.

2.2. Alternatives

Tenint una idea inicial de quins requisits funcionals i no funcionals es buscaven, es va procedir a cercar alguna alternativa existent.

Per estrany que sembli, avui en dia no existeix cap programa a nivell de totes les escoles catalanes per entrar les valoracions adaptada a les etapes d'educació infantil i primària.

Les dues alternatives que s'han valorat abans de dur a terme el projecte són:

- Seguir amb el sistema actual i adaptar-lo als canvis requerits pel Departament d'Ensenyament (bàsicament canvis en les dimensions d'algunes àrees). Com que és un sistema molt obsolet i no compleix molts dels requisits, no es considera una opció vàlida.
- Hi ha un programa de gestió integral dels centres a nivell català, el SAGA, que permet entrar valoracions trimestrals. Alguns mestres l'havien utilitzat en altres escoles on havien estat. No es considera una bona alternativa, ja que és un format molt farragós a l'hora d'entrar les dades i difícil de gestionar, a part de no tenir cap mena de flexibilitat en el format final dels informes.

2.3. Avaluació dels riscos

Després d'avaluar i descartar les alternatives existents, és important avaluar els principals riscos abans de presentar el pressupost.

Els riscos més importants que s'han detectat són:

- Manca d'experiència en projectes integrals d'abast important programats en PHP.
- El client no té totalment definits els models d'informe, per tant és molt probable que hi hagi canvis durant el projecte.
- Necessitat de complir dos tipus diferents de fites, una per cada client (escola i UOC).

2.4. Pressupost

El pressupost que es va presentar a l'escola fou el següent:

- Creació de l'aplicació.....1.000€
- Lloguer anual del servidor d'allotjament.....50€

El pressupost no incloïa el lloguer del domini, ja que l'escola en disposava d'un de propi, jor.cat.

El cost de la creació de l'aplicació es va basar en una feina aproximada de 200 hores. Normalment cobro les hores a 20€, però tenint en compte que aquest projecte em servia de treball de final de grau, vaig rebaixar el preu a la meitat per compartir els costos.

El pressupost va ser acceptat, i així el projecte es va poder tirar endavant.

3. Anàlisi

3.1. Recollida de requisits

Stakeholders

A l'hora de fer la recollida de requisits, és important saber quins són els *stakeholders* interessats en l'aplicació.

Mestres: volen poder entrar trimestralment i anual les valoracions dels alumnes per cadascuna de les assignatures de les quals en són responsables. També volen obtenir-ne informes per enviar als pares.

Hi ha tres tipus de mestres:

- Tutors: valoren les assignatures comunes (matemàtiques, llengua, entorn...) de la classe que tenen assignada.

- Especialistes: valoren la seva assignatura (educació física, anglès, música...) de totes les classes.
- Mestres de suport: tenen alumnes amb necessitats especials i els atenen de forma individualitzada.

Administradors del sistema: volen poder modificar les dades base del sistema (alumnes, anys, mestres, aules...) i obtenir informes de les dades entrades.

Pares i mares: volen obtenir informació clara i concisa sobre el rendiment acadèmic del seus fills.

Administració: imposa unes condicions de privacitat de les dades de mestres i alumnes segons la LOPD.

Escola: determina unes mesures de seguretat per tal que no es perdin les dades en cas de fallada del servidor.

Per realitzar la recollida de requisits a partir dels stakeholders, es van seguir 2 procediments:

- Entrevistes amb alguns dels principals *stakeholders*: un tutor, un especialista, el coordinador d'informàtica, la directora del centre i la mare d'un alumne.
- Qüestionari.

Entrevistes amb els Stakeholders

Jaume Mussarra, tutor de 6è B

Naturalment, he de poder entrar les valoracions dels alumnes de la meua aula (excepte les valoracions dels especialistes). A part de les avaluacions trimestrals, també he d'entrar unes altres valoracions al final de cada cicle.

A més, hauria de poder veure les notes de cada alumne dels trimestres anteriors, i fins i tot no estaria de més poder veure les d'anys anteriors. Hauria de veure d'alguna manera de quins alumnes em falta entrar la valoració.

És indispensable que pugui entrar les valoracions tant des de l'escola com de de casa, en qualsevol lloc i a qualsevol hora. I si pot ser amb la tauleta o el telèfon encara millor.

El que m'agradava del programa actual és que desava automàticament els canvis que entrava.

La nova versió també ho hauria de fer.

Per temes de privacitat, vull poder canviar la meva contrasenya.

També seria útil poder enviar un missatge a l'administrador del sistema, per si detecto algun problema amb el programa o les dades.

Lluís Autonell, especialista d'educació física

Actualment em trobo amb un problema a l'hora de fer les avaluacions. Com que només es fan 2 hores de gimnàstica a la setmana per classe, i he de fer la gimnàstica dels nens de tota l'etapa de primària, molts d'ells no els reconec fàcilment pel seu nom i cognoms. M'aniria molt bé que pogués veure una foto del nen que estic valorant.

Carles Suero, coordinador d'informàtica

Tant la gestió com l'entrada d'informació s'han de poder fer a qualsevol hora i des de qualsevol dispositiu (ordinador, tauleta i telèfon intel·ligent). Potser s'aconseguiria si l'aplicació fos en format web. Per temes administratius, actualment tenim un domini (*lor.cat*) que es podria aprofitar per accedir a l'aplicació (per exemple, amb una adreça de tipus *lor.cat/informes*). El que encara no tenim és allotjament, i és molt important que el seu cost anual sigui mínim (menys de 100€). Voldríem fer còpies de seguretat periòdiques i automàtiques de totes les dades.

Les feines bàsiques que ha de poder fer l'administrador són: crear, modificar i eliminar cursos escolars, i modificar tant els mestres com els alumnes del trimestre actual. Les classes no cal que es puguin modificar, no comptem que hi hagi canvis en els cursos o els grups.

El sistema ha de poder tenir més d'un administrador. Les altes i baixes d'administració les ha de poder fer un administrador, però un administrador no es pot donar de baixa d'administració ell mateix. Així evitem que, en algun moment, el sistema es pugui quedar sense cap administrador. L'administrador no cal que sigui un mestre; podria ser, per exemple, el director.

L'entrada al sistema ha de ser individualitzada, amb un nom d'usuari i una contrasenya (que inicialment les podria designar l'administrador). Cal un sistema de contrasenyes segur, fiable i versàtil. Cada usuari ha de poder canviar la seva contrasenya, i en cas que la perdi hi ha d'haver alguna forma de recuperar-la. No estaria malament que també poguessin canviar el seu nom d'usuari. Podria ser interessant que es registri cada cop que un usuari intenta entrar a l'aplicació, ja sigui de forma reeixida o no.

Per poder gestionar els alumnes, aniria bé poder copiar els del curs o trimestre anterior. Els alumnes es distribueixen en 2 grups, A i B. Cal poder decidir cada any quins van a cada grup. En el cas del primer curs (P3), seria ideal poder entrar els alumnes a partir d'un arxiu de tipus Excel (disposem d'un programa de gestió del centre, el SAGA, que té les dades de tots els alumnes i permet fer una exportació a aquest format).

L'aplicació ha de guardar l'històric de totes les dades: mestres, alumnes, classes, cursos, valoracions... S'han de poder recuperar les valoracions de qualsevol alumne en qualsevol curs.

Mercè Soler, directora del centre

Actualment el centre ofereix dues etapes: el segon cycle d'educació infantil i tots els tres cycles de primària (inicial, mitjà i superior). Educació infantil (EI) té 3 nivells (P3, P4 i P5), i els altres cycles en tenen 2 (1r i 2n, 3r i 4t, 5è i 6è). Cada nivell té 2 grups (A i B). No es preveu que aquesta estructura pugui canviar al llarg dels anys.

Cada classe (combinació de nivell i grup) té assignat un tutor, que pot canviar a cada curs. Cal preveure, però, la possibilitat que una classe tingui més d'un tutor (tutoria compartida), i que durant l'any hi hagi un canvi de tutor (mestre substituït). Els tutors només han de poder valorar les assignatures que imparteixen a la seva classe.

A part dels tutors, hi ha els especialistes, que imparteixen les assignatures d'anglès, música, educació física i religió de tota una etapa. Algunes assignatures les imparteixen més d'un especialista (per exemple, actualment tenim 2 professors de gimnàstica). Pot ser que un mestre sigui a la vegada tutor i especialista.

Un tipus a part de mestre són els d'educació especial i audició i llenguatge. Aquests no imparteixen una assignatura en sí, sinó que ofereixen atenció especialitzada als alumnes en dificultats.

He de poder obtenir les valoracions de qualsevol alumne de qualsevol any. També vull obtenir estadístiques per classes i, a poder ser, per alumnes.

Pel que fa als informes, n'hi ha d'haver 2 models: un per educació infantil i un altre per primària. Interessa que es puguin imprimir per cada alumne individualment i per tota una classe de cop. Cada trimestre ha de mostrar les seves dades i les dels trimestres anteriors.

Cadascuna de les etapes té les seves pròpies assignatures (al departament d'ensenyament en diuen àrees). Les àrees estan agrupades en àmbits. De cada àrea se n'ha de poder valorar una sèrie de dimensions (valoració parcial). També es fa una valoració general tota l'àrea (valoració

global). L'àrea de religió, que és optativa, només té valoració global.

A primària es fan 3 valoracions de cada àrea, una per trimestre. En el cas d'educació infantil, només 2.

Esperança Niubò, mare d'un alumne

Vull poder veure d'una forma clara i concisa les valoracions que els mestres han fet del meu fill trimestralment. Els informes que rebo actualment estan una mica desordenats, són visualment poc atractius i tenen moltes pàgines, algunes amb molts espais en blanc.

Qüestionari

Aquest és el resultat del qüestionari que van omplir els diferents *stakeholders* així com altres membres de l'equip directiu.

1 / 0 / -1 (d'acord, interessant, no implementar)	
Com a mestre, vull:	
Entrar trimestralment les valoracions dels meus alumnes	1
Saber les valoracions de trimestres anteriors d'un alumne	1
Saber de quins alumnes em manca fer les valoracions	1
Fer els informes des de l'escola	1
Fer els informes des de casa	1
Fer els informes des d'un telèfon mòbil o tauleta	1
Canviar la meva contrasenya	1
Saber les valoracions de cursos anteriors d'un alumne	0
Poder enviar un avís a l'administrador de l'aplicació	0
Com a administrador, vull:	
Modificar els cursos	1
Modificar els alumnes del curs actual	1
Modificar els mestres del curs actual	1
Copiar els alumnes del curs anterior	1
Entrar els alumnes del curs actual a partir d'un fitxer	1
Copiar els mestres del curs anterior	1
Copiar les aules del curs anterior	1
Assignar els alumnes per curs	1
Assignar els mestres per curs	1
Assignar les aules per curs	1
Imprimir els informes individualment	1
Imprimir els informes de tota una aula	1
Administrar l'aplicació des de l'escola	1
Administrar l'aplicació des de casa	1
Canviar la meva contrasenya	1
Poder gestionar els administradors del sistema	1
Que l'aplicació guardi els històrics dels alumnes, aules, cursos, valoracions, mestres...	1
Obtenir les valoracions dels alumnes de qualsevol curs	1
Obtenir estadístiques per aules	1
Tenir un registre de l'activitat de cada mestre dins l'aplicació	0
Administrar l'aplicació des d'un telèfon mòbil o tauleta	0
Obtenir estadístiques per alumnes	0
Modificar les aules del curs actual	-1
Assignar els alumnes per trimestre	1
Assignar les especialitats, tutories i reforç per trimestre	1
Imprimir els informes d'un grup d'alumnes	-1
Imprimir els informes de tot un curs	-1
Com a pare, vull:	
Saber les valoracions trimestrals del meu fill a final de curs desglossades en assignatures	1
Saber l'evolució del meu fill durant els diferents cursos	0

3.2. Requisites

A partir de les entrevistes i el qüestionari, es va procedir a elaborar els requisits no funcionals i els funcionals (aquests en forma de casos d'ús).

Requisits no funcionals

Operacionals

El sistema s'ha de poder utilitzar en qualsevol dispositiu (sobretaula, portàtil, tauleta i telèfon) i en qualsevol moment. Sobretot és important en els usuaris de tipus mestre.

Manteniment i suport

El sistema ha de ser barat de mantenir.

Seguretat

El sistema registrarà el nom d'usuari, la data i el resultat (positiu o negatiu) cada cop que un usuari vulgui accedir al sistema.

Culturals

Només es contempla fer-lo en català.

Legals

Només han de poder accedir a les dades els administradors, tutors, especialistes i mestres de suport actuals.

Casos d'ús (nivell alt)

Inicialment es va realitzar una llista dels casos d'ús més importants en forma d'alt nivell, per tal d'obtenir l'aprovació de l'equip directiu.

Entrar una valoració (tutor)

Un tutor escull entrar les valoracions dels alumnes del trimestre actual. Per cada secció, omple les valoracions de cada alumne i el sistema les emmagatzema. En cas que existeixin valoracions, el sistema les mostra. Quan finalitza la secció, passa a la següent, i així fins a l'última. Si s'ha oblidat de valorar algun alumne, el sistema l'avisava.

Entrar una valoració (especialista)

Un tutor escull entrar les valoracions dels alumnes del trimestre actual. Per cada classe, omple les valoracions de cada alumne i el sistema les emmagatzema. En cas que existeixin valoracions, el sistema les mostra. Quan finalitza la classe, passa a la següent, i així fins a l'última. Si s'ha oblidat de valorar algun alumne, el sistema l'avisava.

Entrar una valoració (mestre de suport)

Un mestre de suport escull entrar les valoracions dels alumnes del trimestre actual. El sistema mostra un llistat de tots els alumnes, i el mestre en tria un. En cas que existeixin valoracions, el sistema les mostra.

Veure valoracions d'anys / períodes anteriors (mestre)

El mestre escull un període, una secció i un alumne i demana veure'n les valoracions. El sistema les mostra.

Enviar un missatge a l'administrador (mestre)

El mestre escriu un missatge i l'envia. El sistema el desa i el distribueix a tots els administradors.

Canviar la contrasenya (usuari)

L'usuari entra al seu perfil i canvia la contrasenya. El sistema la valida i, si és correcta, desa els canvis.

Modificar el curs actual (administrador)

L'administrador crea un nou curs o en modifica un d'existent. Si vol eliminar un any que té dades assignades, el sistema no ho admet. En cas contrari, el sistema desa els canvis.

Donar d'alta els mestres del trimestre (administrador)

L'administrador assigna els mestres a cadascuna de les classes / especialitats. També determina si són administradors. El sistema els emmagatzema.

Donar d'alta els alumnes del curs (nivell P3, administrador)

L'administrador escull importar un llistat d'alumnes des d'un fitxer. El sistema llegeix els alum-

nes del fitxer i els introdueix com a alumnes de P3. En cas que ja existeixin alumnes de P3, el sistema els substitueix pels nous.

Donar d'alta els alumnes del curs (nivells > P3, administrador)

L'administrador escull actualitzar els alumnes del primer trimestre del curs. El sistema copia els alumnes del curs anterior. L'administrador modifica el llistat. El sistema emmagatzema els canvis.

Donar d'alta els alumnes del trimestre (administrador)

L'administrador escull actualitzar els alumnes d'un trimestre diferent del primer. El sistema copia els alumnes del trimestre anterior. L'administrador modifica el llistat. El sistema emmagatzema els canvis.

Restaurar la contrasenya (administrador)

L'administrador demana restaurar una contrasenya a un usuari. El sistema li assigna la contrasenya per defecte.

Obtenir estadístiques d'una classe (administrador)

L'administrador escull la classe i demana l'estadística. El sistema la mostra per tots els anys de l'històric.

Imprimir els informes d'una classe (mestre)

L'administrador escull la classe de la qual vol imprimir els informes. El sistema mostra les opcions d'impressió del SO per tal que l'administrador les pugui modificar. El sistema envia els informes a la impressora.

Modificar una dimensió (administrador)

L'administrador escull una assignatura i un trimestre, i afegeix o suprimeix dimensions. Si les dimensions eliminades no contenen valoracions, el sistema desa els canvis.

Login (usuari)

L'usuari entrarà les seves dades de nom d'usuari i contrasenya. El sistema les validarà i el deixarà accedir en cas positiu. En qualsevol cas, el sistema registrarà l'intent d'accés.

A partir d'aquesta llista inicial de casos d'ús es van seleccionar els que tenien més risc i els més urgents de cara al primer lliurament. Posteriorment es van detallar aquests casos d'ús i es va procedir a fer el model del domini. A mesura que l'anàlisi avançava, s'anaven trobant nous ca-

sos d'ús que calia implementar. Aquests varen ser:

- Login
- Entrar una valoració
- Veure valoracions d'un alumne
- Imprimir informes d'un alumne
- Imprimir informes d'un curs

Aquest procés fou iteratiu. En aquesta memòria, per qüestions d'ordre, es llisten els requisits detallats en la seva globalitat.

3.3. Casos d'ús detallats

Cu01: Login

Actor principal: usuari

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: cap

Garanties mínimes: el sistema registrarà l'intent d'entrada al sistema.

Garanties en cas d'èxit: l'usuari podrà accedir al sistema i tindrà accés a les funcions corresponents al seu rol.

Escenari principal d'èxit:

1. L'usuari entra el seu nom d'usuari i la contrasenya.
2. El sistema verifica les dades i registra l'intent d'accés.
3. El sistema permet l'accés a l'usuari.

Extensions:

- 3a. El sistema no reconeix l'usuari
 - 3a1. El sistema informa l'usuari de l'error
 - 3a2. Tornem al pas 1

Cu02: Logout

Actor principal: usuari

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari està registrat

Garanties mínimes:

Garanties en cas d'èxit: el sistema no té cap usuari registrat.

Escenari principal d'èxit:

1. L'usuari surt del sistema.

Cu03: Entrar una valoració

Actor principal: mestre

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema, i té assignada alguna aula, especialitat o aula de reforç en el període actiu.

Garanties mínimes:

Garanties en cas d'èxit: el sistema emmagatzemarà les dades introduïdes pel mestre i les hi mostrarà. El mestre podrà navegar per totes les classes on tingui alumnes per poder valorar.

Escenari principal d'èxit:

1. El mestre escull un alumne.
2. El sistema mostra les valoracions anuals de l'alumne en el curs actiu.
3. El mestre modifica les valoracions de cada dimensió i àrea.
3. El sistema enregistra automàticament cadascun dels canvis.
4. L'usuari finalitza les valoracions.

Extensions:

Cu04: Veure valoracions d'un curs d'un alumne

Actor principal: mestre

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema i ser tutor de l'alumne en el període actiu.

Garanties mínimes:

Garanties en cas d'èxit: l'usuari té totes les valoracions de l'alumne en el curs demanat.

Escenari principal d'èxit:

1. El mestre escull un alumne i un curs, i en demana veure les valoracions.
2. El sistema mostra les valoracions anuals de l'alumne.

Extensions:

- 2a. L'alumne no té cap valoració en el període escollit.
 - 2a1. El sistema informa que l'alumne no té cap valoració assignada.
 - 2a2. Tornem al pas 1.

Cu05: Imprimir els informes anuals d'una classe

Actor principal: usuari

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema i és tutor d'almenys una classe o adminis-

trador.

Garanties mínimes:

Garanties en cas d'èxit: l'usuari disposa dels informes anuals de tots els alumnes de la classe escollida.

Escenari principal d'èxit:

1. L'usuari escull una classe i en demana els informes.
2. El sistema genera un arxiu agrupant els informes anuals dels alumnes de la classe en el curs actiu.
3. El mestre imprimeix els informes a doble cara.

Extensions:

- 2a. No existeixen valoracions pels alumnes de la classe escollida.
 - 2a1. El sistema informa l'usuari que no hi ha valoracions.
 - 2a2. Tornem al pas 1.

Cu06: Imprimir els informes anuals d'un alumne

Actor principal: usuari

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema i és tutor d'almenys una classe o administrador.

Garanties mínimes: l'usuari disposa dels informes anuals de l'alumne escollit.

Garanties en cas d'èxit:

Escenari principal d'èxit:

1. El mestre escull un alumne i un curs.
2. El sistema genera un arxiu amb l'informe anual de l'alumne en el curs escollit.
3. El mestre imprimeix l'informe a doble cara.

Extensions:

- 2a. No existeixen valoracions per l'alumne en el curs escollit.
 - 2a1. El sistema informa l'usuari que no hi ha valoracions.
 - 2a2. Tornem al pas 1.

Cu07: Mostrar els alumnes per diferents filtres

Actor principal: usuari

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema.

Garanties mínimes:

Garanties en cas d'èxit: l'usuari disposa de les dades dels alumnes segons el filtre escollit.

Escenari principal d'èxit:

1. El mestre entra un o més filtres segons els quals vol visualitzar els alumnes.
2. El sistema mostra una llista amb els alumnes segons el filtre.

Extensions:

Cu08: Enviar missatge

Actor principal: mestre

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a mestre.

Garanties mínimes:

Garanties en cas d'èxit: els administradors reben el missatge, que queda enregistrat al sistema

Escenari principal d'èxit:

1. El mestre envia un missatge.
2. El sistema enregistra el missatge i l'envia a tots els administradors.

Extensions:

Cu09: Canviar les dades personals

Actor principal: usuari

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema.

Garanties mínimes: les dades de l'usuari es mantenen.

Garanties en cas d'èxit: les dades de l'usuari queden modificades.

Escenari principal d'èxit:

1. L'usuari modifica les seves dades de contrasenya, correu electrònic, nom i cognoms.
2. El sistema valida les dades i les enregistra.

Extensions:

- 2a. Les dades contenen algun error de validació.
 - 2a1. El sistema informa de l'error.
 - 2a2. Tornem al pas 1.

Cu10: Crear un curs

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista de cursos no es modifica.

Garanties en cas d'èxit: es crea un curs nou.

Escenari principal d'èxit:

1. L'administrador crea un curs nou.
2. El sistema demana un nom pel curs i l'usuari l'entra.
3. El sistema valida el nom del curs entrat i l'enregistra com a curs nou.
4. El sistema mostra a l'administrador la llista de cursos.

Extensions:

- 3a. El nom del curs ja existeix o el format és invàlid.
 - 3a1. El sistema informa de l'error.

3a2. Tornem al pas 2.

Cu11: Modificar un curs

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista de cursos no es modifica.

Garanties en cas d'èxit: la llista de cursos conté el curs amb les dades modificades.

Escenari principal d'èxit:

1. L'administrador modifica un curs de la llista i li assigna un nom nou.
2. El sistema valida el nom del curs modificat i enregistra els canvis.

Extensions:

- 2a. El nom del curs ja existeix o el format és invàlid.
 - 2a1. El sistema informa de l'error.
 - 2a2. Tornem al pas 1.

Cu12: Eliminar un curs

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista de cursos no es modifica.

Garanties en cas d'èxit: la llista de cursos no conté el curs eliminat.

Escenari principal d'èxit:

1. L'administrador demana eliminar un curs de la llista.
2. El sistema comprova que el curs no tingui dades assignades i l'elimina de la llista de cursos.

Extensions:

- 1a. Hi ha dades al sistema que fan referència al curs.
 - 2a1. El sistema informa de l'error.

Cu13: Establir el període actiu

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes:

Garanties en cas d'èxit: el període actiu és el curs i el trimestre establerts al cas d'ús.

Escenari principal d'èxit:

1. El mestre canvia el curs i el trimestre actius.
2. El sistema enregistra els canvis.

Extensions:

Cu14: Afegir un usuari

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista d'usuaris es manté.

Garanties en cas d'èxit: existeix un usuari nou amb les dades entrades per l'administrador.

Escenari principal d'èxit:

1. L'administrador demana entrar un nou usuari.
2. L'usuari entra les dades de nom, cognoms, correu electrònic i nom d'usuari.
3. El sistema valida les dades i enregistra un nou usuari amb aquestes dades i la contrasenya per defecte.

Extensions:

- 3a. El nom d'usuari ja existeix o el format de les dades no és vàlid.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2.

Cu15: Eliminar un usuari

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista d'usuaris es manté.

Garanties en cas d'èxit: la llista d'usuaris no conté l'usuari eliminat.

Escenari principal d'èxit:

1. L'administrador demana eliminar un usuari.
2. El sistema comprova que l'usuari no té dades associades i l'elimina.

Extensions:

- 2a. Hi ha dades al sistema relacionades amb l'usuari.
 - 2a1. El sistema informa de l'error.

Cu16: Canviar l'assignació d'una tutoria a un mestre

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: el mestre està assignat a les tutories inicials.

Garanties en cas d'èxit: les tutories del mestre reflecteixen els canvis entrats al cas d'ús.

Escenari principal d'èxit:

1. L'administrador escull un mestre.
2. L'administrador li canvia l'assignació en una classe, ja sigui per eliminar-la o crear-la nova.
3. El sistema valida el canvi d'assignació i l'enregistra.

Extensions:

- 3a. La classe desassignada ja conté valoracions entrades pel mestre en el període actual.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2

Cu17: Canviar l'assignació d'una especialitat a un mestre

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: el mestre està assignat a les especialitats inicials.

Garanties en cas d'èxit: les especialitats del mestre reflecteixen els canvis entrats al cas d'ús.

Escenari principal d'èxit:

1. L'administrador escull un mestre.
2. L'administrador li canvia l'assignació en una àrea, ja sigui per eliminar-la o crear-la nova.
3. El sistema valida el canvi d'assignació i l'enregistra.

Extensions:

- 3a. L'àrea desassignada ja conté valoracions entrades pel mestre en el període actual.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2

Cu18: Modificar el rol d'administrador d'un mestre

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: l'usuari afectat conserva el rol d'administrador.

Garanties en cas d'èxit: el rol d'administrador de l'usuari afectat correspon amb el valor entrat al cas d'ús.

Escenari principal d'èxit:

1. L'administrador demana canviar el rol d'administrador d'un usuari.
2. El sistema valida el canvi de dades i l'enregistra al sistema.

Extensions:

- 2a. L'usuari afectat és el mateix administrador.
 - 2a1. El sistema informa que un administrador no es pot donar de baixa ell mateix.

Cu19: Canviar l'assignació d'una aula de reforç a un mestre

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: el mestre està assignat a les aules de reforç inicials.

Garanties en cas d'èxit: les aules de reforç del mestre reflecteixen els canvis entrats al cas d'ús.

Escenari principal d'èxit:

1. L'administrador escull un mestre.
2. L'administrador li canvia l'assignació en una aula de reforç, ja sigui per eliminar-la o crear-la nova.
3. El sistema valida el canvi d'assignació i l'enregistra.

Extensions:

- 3a. L'aula desassignada ja conté valoracions entrades pel mestre en el període actual.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2.

Cu20: Donar d'alta els alumnes del curs (nivell P3)

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista d'alumnes de P3 del curs actiu no canvia.

Garanties en cas d'èxit: els alumnes de les classes de P3 del curs actiu corresponen als de l'arxiu extern.

Escenari principal d'èxit:

1. L'administrador escull un arxiu amb les dades dels alumnes de P3 i demana entrar-ne les dades al sistema.
2. El sistema valida les dades i crea una llista d'alumnes de P3 pel curs actiu corresponent als alumnes continguts a l'arxiu.
3. El sistema mostra els alumnes entrats.

Extensions:

- 2a. Les classes de P3 del curs actiu ja contenen valoracions.
 - 2a1. El sistema informa de l'error.
 - 2a2. Tornem al pas 1.
- 2b. El format de les dades d'entrada és incorrecte.
 - 2b1. El sistema informa de l'error.
 - 2b2. Tornem al pas 1.

Cu21: Donar d'alta els alumnes del curs (nivell > P3)

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador. El trimestre actiu és el primer.

Garanties mínimes: la llista d'alumnes del nivell seleccionat del curs actiu no canvia.

Garanties en cas d'èxit: els alumnes del nivell seleccionat del curs actiu corresponen als de l'últim trimestre del curs anterior.

Escenari principal d'èxit:

1. L'administrador selecciona un nivell i demana que els alumnes del primer trimestre del curs actiu siguin els mateixos que els de l'últim trimestre del curs seleccionat.
2. El sistema valida les dades i enregistra els canvis.

Extensions:

- 2a. L'últim trimestre del nivell seleccionat no té alumnes assignats.
 - 2a1. El sistema informa de l'error.
 - 2a2. Tornem al pas 1.
- 2b. El nivell ja té alumnes assignats pel primer trimestre del curs actiu.
 - 2b1. El sistema informa de l'error.
 - 2b2. Tornem al pas 1.

Cu22: Donar d'alta els alumnes del trimestre

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador. El trimestre actiu no és el primer.

Garanties mínimes: la llista d'alumnes del nivell seleccionat del trimestre actiu no canvia.

Garanties en cas d'èxit: els alumnes del nivell seleccionat del curs actiu corresponen als del trimestre anterior del curs actual.

Escenari principal d'èxit:

1. L'administrador selecciona un nivell i demana que els alumnes del trimestre i el curs actius siguin els mateixos que els del trimestre anterior.
2. El sistema valida les dades i enregistra els canvis.

Extensions:

- 2a. El trimestre anterior del nivell seleccionat no té alumnes assignats.
 - 2a1. El sistema informa de l'error.
 - 2a2. Tornem al pas 1.
- 2b. El nivell ja té alumnes assignats pel trimestre i el curs actius.
 - 2b1. El sistema informa de l'error.
 - 2b2. Tornem al pas 1.

Cu23: Canviar l'assignació d'alumnes al trimestre actual

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista d'alumnes no canvia.

Garanties en cas d'èxit: la llista d'alumnes reflecteix els canvis introduïts al cas d'ús.

Escenari principal d'èxit:

1. L'administrador selecciona un alumne del curs i trimestre actius.

2. L'administrador demana canviar l'assignació de classe de l'alumne.
3. El sistema valida l'assignació i enregistra els canvis introduïts.

Extensions:

- 3a. L'alumne té entrades valoracions en el curs i trimestre actius.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2.

Cu24: Eliminar alumnes del sistema

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista d'alumnes no canvia.

Garanties en cas d'èxit: l'alumne eliminat no existeix a la llista d'alumnes.

Escenari principal d'èxit:

1. El selecciona un alumne i demana eliminar-lo del sistema
2. El sistema valida l'acció i elimina l'alumne del sistema.

Extensions:

- 2a. Existeixen dades al sistema relacionades amb l'alumne.
 - 2a1. El sistema informa de l'error.
 - 2a2. Tornem al pas 1.

Cu25: Restaurar la contrasenya

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes:

Garanties en cas d'èxit: l'usuari té assignada la contrasenya per defecte.

Escenari principal d'èxit:

1. El mestre demana restaurar la contrasenya d'un usuari.
2. El sistema assigna a l'usuari la contrasenya per defecte.

Extensions:

Cu26: Obtenir estadístiques d'una classe

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes:

Garanties en cas d'èxit: l'administrador disposa d'un informe amb les estadístiques de la classe.

Escenari principal d'èxit:

1. El mestre selecciona una classe i demana un informe estadístic.
2. El sistema mostra les estadístiques de la classe per tots els cursos que n'existeixen valoracions.

Extensions:

Cu27: Afegir una dimensió a una àrea

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista de dimensions de l'àrea no es modifica.

Garanties en cas d'èxit: la llista de dimensions de l'àrea seleccionada conté una nova dimensió.

Escenari principal d'èxit:

1. L'administrador escull una àrea i una llista de cicles associats a l'etapa de l'àrea seleccionada.
2. L'administrador demana introduir una nova dimensió a l'àrea.
3. El sistema valida les dades i enregistra la nova dimensió i l'assigna a l'àrea per tots els cicles de l'àrea seleccionada.

Extensions:

- 3a. El nom de la dimensió ja existeix per qualsevol dels cicles seleccionats.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 1.
- 3b. Hi ha algun error de validació en les dades de la dimensió.
 - 3b1. El sistema informa de l'error.
 - 3b2. Tornem al pas 2.

Cu28: Eliminar una dimensió

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista de dimensions no es modifica.

Garanties en cas d'èxit: la llista de dimensions de l'àrea seleccionada no conté la dimensió.

Escenari principal d'èxit:

1. L'administrador escull una àrea.
2. L'administrador escull una dimensió i demana eliminar-la.
3. El sistema valida l'acció i elimina la dimensió del sistema.

Extensions:

- 3a. Existeixen dades al sistema relacionades amb la dimensió.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2.

Cu29: Canviar l'assignació d'una dimensió a un curs

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: la llista de dimensions no es modifica.

Garanties en cas d'èxit: la llista de dimensions reflecteix els canvis introduïts al cas d'ús.

Escenari principal d'èxit:

1. L'administrador escull una dimensió.
2. L'administrador demana canviar l'assignació que la dimensió té a un o més cicles.
3. El sistema valida l'acció i enregistra els canvis.

Extensions:

- 3a. Existeixen valoracions entrades per la dimensió en el trimestre i curs actius.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2.
- 3b. Existeix una dimensió amb el mateix nom a l'àrea seleccionada.
 - 3b1. El sistema informa de l'error.
 - 3b2. Tornem al pas 2.

Cu30: Modificar una dimensió

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes: les dades associades a la dimensió no es modifiquen.

Garanties en cas d'èxit: la llista de dimensions conté una dimensió amb les dades introduïdes.

Escenari principal d'èxit:

1. L'administrador selecciona una dimensió.
2. L'administrador demana modificar les dades de la dimensió.
3. El sistema valida les dades introduïdes i enregistra els canvis.

Extensions:

- 3a. Existeix una dimensió amb el mateix nom a l'àrea assignada en la que està assignada.
 - 3a1. El sistema informa de l'error.
 - 3a2. Tornem al pas 2.
- 3b. Hi ha algun error de validació en les dades de la dimensió.
 - 3b1. El sistema informa de l'error.
 - 3b2. Tornem al pas 2.

Cu31: Mostrar tots els mestres

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes:

Garanties en cas d'èxit: l'administrador disposa d'un llistat de tots els usuaris amb les dades relacionades.

Escenari principal d'èxit:

1. L'administrador demana un llistat dels mestres.
2. El sistema mostra un llistat amb els mestres, les seves dades personals i les tutories, especialitats i aules de reforç assignades en el període actiu.

Cu32: Mostrar totes les classes

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes:

Garanties en cas d'èxit: l'administrador disposa d'un llistat de totes les classes amb les dades relacionades.

Escenari principal d'èxit:

1. L'administrador demana un llistat de les classes.
2. El sistema mostra un llistat amb les classes, el número d'alumnes i els tutors assignats en el període actiu.

Cu33: Mostrar totes les especialitats

Actor principal: administrador

Àmbit: sistema

Nivell d'objectiu: usuari

Precondicions: l'usuari s'ha d'haver registrat al sistema com a administrador.

Garanties mínimes:

Garanties en cas d'èxit: l'administrador disposa d'un llistat de totes les especialitats amb les dades relacionades.

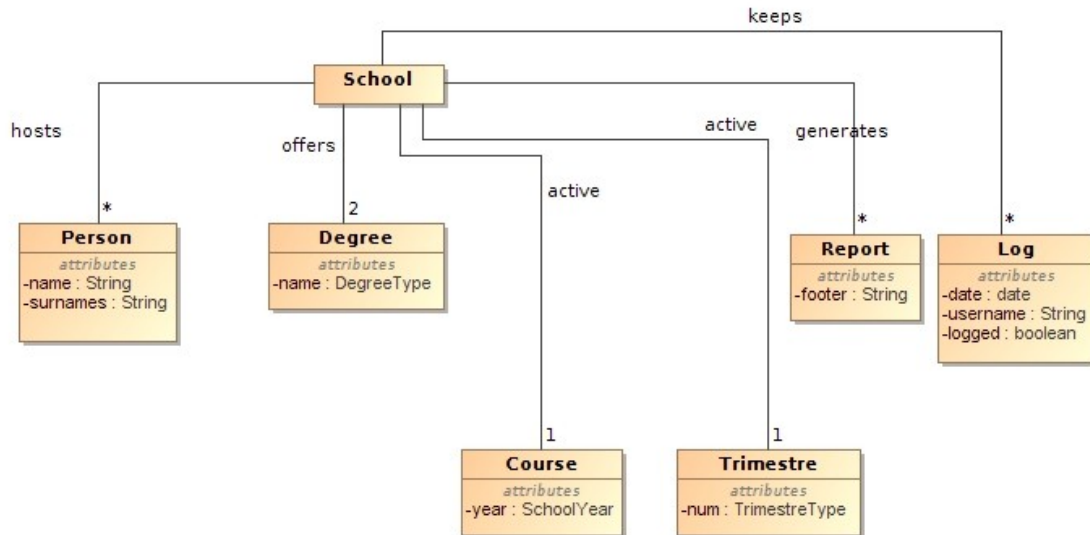
Escenari principal d'èxit:

1. L'administrador demana un llistat de les especialitats.
2. El sistema mostra un llistat amb les especialitats, el número d'alumnes de cadascuna i els especialistes assignats en el període actiu.

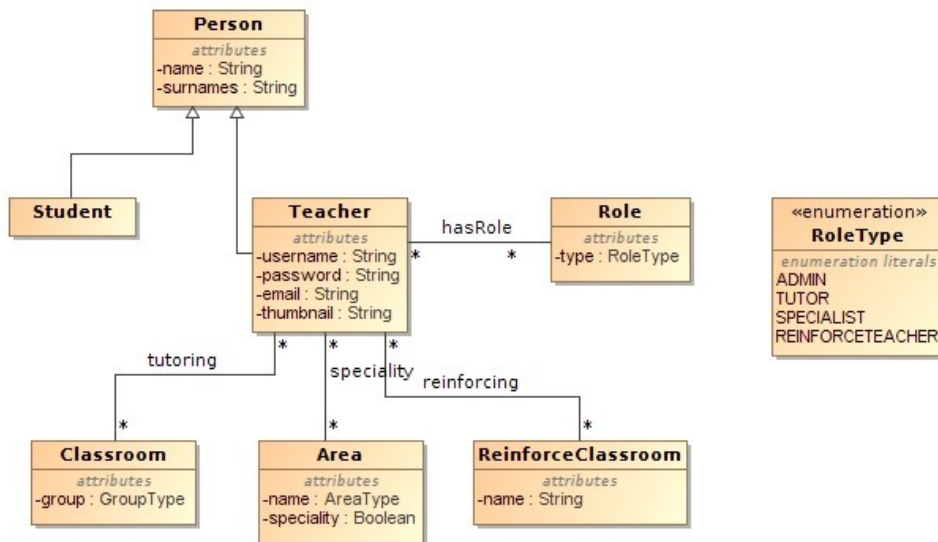
3.4. Model del domini

S'inclou el model de domini en forma de models parcials, ja que és més entenedor.

Estructura bàsica de l'escola



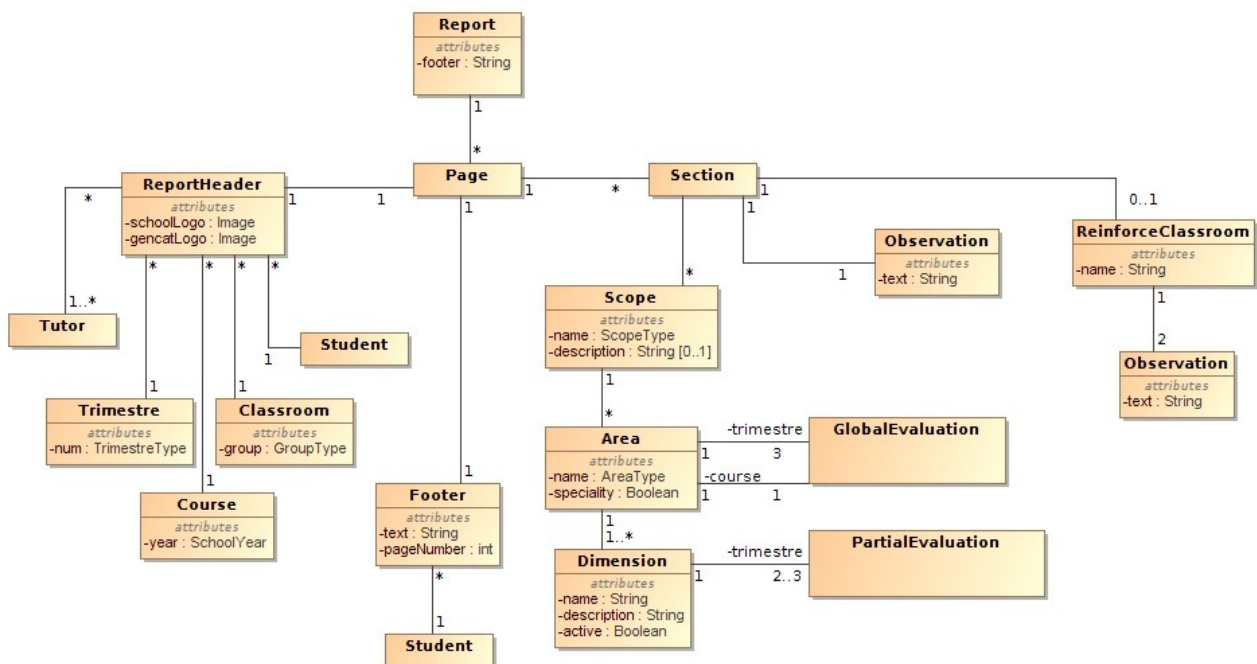
Personal associat a l'escola



Les relacions entre els professors i les tutories, especialitats i reforç individual són variables en el temps. En el diagrama detallat de l'escola queda reflectit.

- Només pot existir una etapa (*Degree*) de cada tipus.
- L'etapa d'educació infantil té 1 cicle (segon cicle) i la de primària 3 (inicial, mitjà i superior).
- Cada nivell té 2 classes, una per cada tipus de grup.
- L'etapa d'educació infantil està associada als 2 primers trimestres, i la de primària a tots 3.

Informes



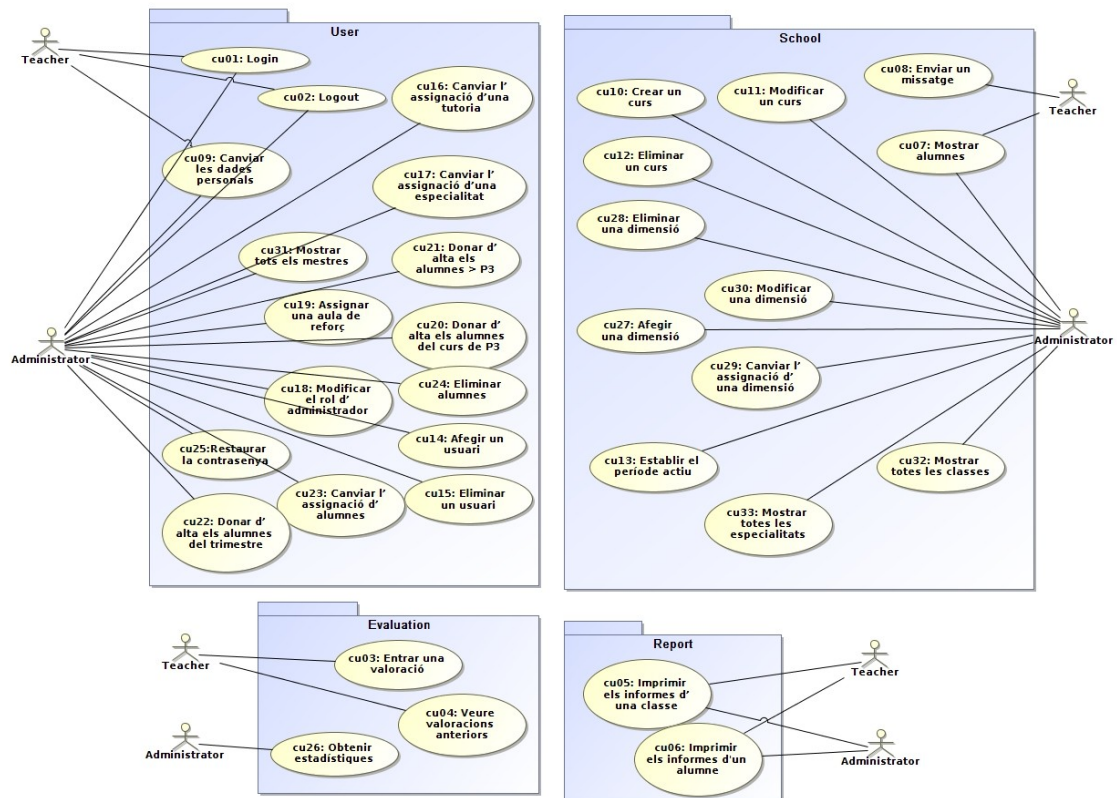
Les relacions temporals de trimestre i curs entre les avaluacions i les àrees i les dimensions, estan detallades al model d'avaluacions.

Restriccions textuais:

- Les dimensions que no continguin cap valoració per l'estudiant en el present curs no han de sortir als informes.
- Les àrees que no tinguin cap dimensió avaluada no han de sortir als informes.
- Els àmbits que no tinguin cap àrea avaluada no han de sortir als informes.
- Les seccions no poden partir-se entre dues pàgines.

3.5. Model de casos d'ús

Degut a la gran quantitat de casos d'ús, s'han dividit en 4 paquets segons la seva funcionalitat bàsica: gestió dels usuaris, de l'escola, de les valoracions i dels informes.



4. Arquitectura

4.1. Arquitectura física

L'arquitectura física segueix el model client-servidor. Tota l'aplicació està allotjada en un únic servidor, i els clients interactuen amb el servidor a través de missatges HTTP.

De la banda del **client**, tenim els següents elements:

- **Navegador:** aplicació que mostra a l'usuari les dades inserides en marques HTML que li proporciona el servidor, maquetades segons les instruccions dels arxius CSS i que executa el codi Javascript. Es comunica amb el servidor per HTTP. L'aplicació funciona en qualsevol navegador en les seves darreres versions.
- **HTML:** dades proporcionades pel servidor i marques que donen sentit semàntic a aquestes dades (determinen si formen part d'un paràgraf, una capçalera, una cel·la d'una taula, un enllaç...).

- **CSS:** instruccions de maquetació del contingut de la pàgina.
- **Javascript:** lògica que s'executa des del client i que té diversos propòsits:
 - Rebre les peticions de l'usuari.
 - Llegir parts de la pàgina.
 - Actualitzar les dades de la pàgina.
 - Comunicar-se amb el servidor a través del protocol HTTP mitjançant AJAX.
 - Utilitzar lògica per prendre decisions.

El **servidor** conté la informació en forma de base de dades i tota la lògica necessària en format PHP per poder servir les dades al client, a més del programari necessari. Els requeriments de programari del servidor són: Apache 2.0, PHP 7.0 i MySQL 5.7.

4.2. Arquitectura lògica

L'arquitectura està dividida en 3 capes: presentació, serveis i accés directe a dades. Les dues primeres capes estan dividides horitzontalment en 4 components, igual que els casos d'ús: usuari, escola, avaluacions i informes.

Presentació

La presentació segueix el paradigma MVC.

Cada pàgina té assignat el seu propi controlador, que és el mateix tant si es fan peticions directament des de la pàgina o a través de javascript. En cas que es facin des de javascript, un controlador (*AjaxController.php*) centralitza les peticions i les distribueix al controlador que li toca segons la pàgina que hagi enviat la petició. Per exemple, si es fa una petició des de la pàgina d'entrada (*login.php*), el controlador AJAX envia la petició al controlador *LoginController.php*.

HeaderViewModel.php és un model que s'encarrega d'obtenir i formatar les dades corresponents a la capçalera i que són iguals per totes les pàgines. Així totes les pàgines, quan es carreguen, li envien una petició per poder formatar la capçalera (nom d'usuari, menús...).

Tots els models hereten d'un model genèric, *MainViewModel.php*, que alberga funcions utilit-

zades pels altres models, com per exemple recuperar variables de sessió o transformar variables enviades per un formulari (post) a variables de sessió.

Pel que fa a la interacció entre els controladors i els models, qualsevol controlador pot fer peticions a qualsevol dels models. També poden accedir directament a la capa de serveis.

La funció dels models és generar un conjunt de dades en format vector per tal que la plantilla les transformi en marques HTML. En algunes ocasions utilitza la resta de models per obtenir aquestes dades ja formatades (per tant, els models interaccionen entre ells de forma horitzontal).

Una de les parts més delicades de la capa de presentació és decidir quin sistema es farà servir com a plantilla per generar les pàgines HTML a partir de les dades. En aquest cas he utilitzat una llibreria externa, [Transphporm](#), que utilitza un sistema que crec força interessant basat en una sintaxi similar a la que fa servir CSS. La pàgina final es genera a partir de dos arxius: un amb extensió XML que conté les marques HTML igual que qualsevol pàgina web; un altre amb extensió tss que especifica amb quines dades s'ha de substituir el contingut de l'arxiu XML. Així es manté totalment diferenciat el disseny de la pàgina de les dades. En general funciona molt bé, però no tolera algunes estructures amb diversos anidaments. En aquests casos he utilitzat una classe personalitzada per generar l'HTML, *TplEngine.php*.

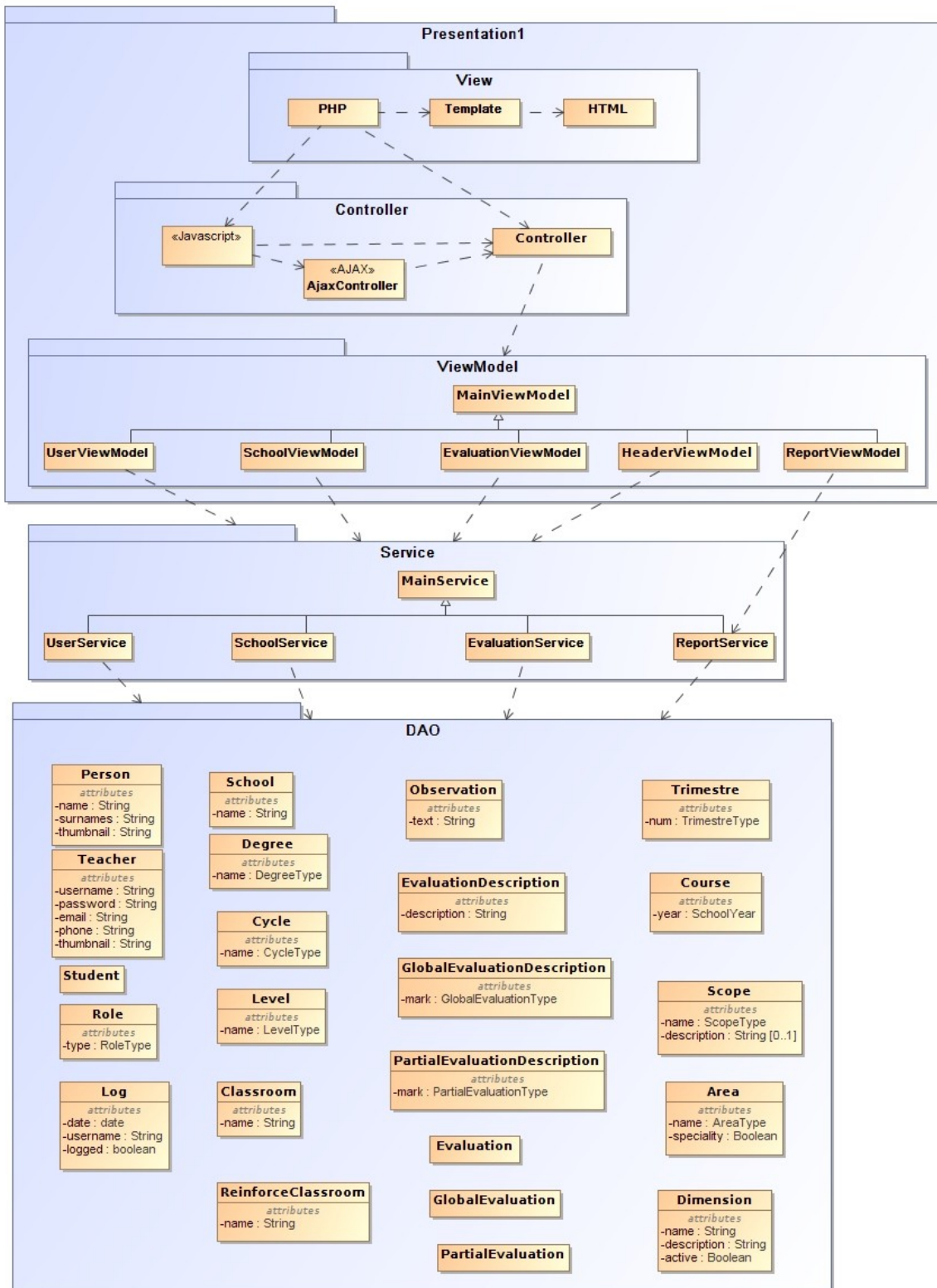
D'altra banda, també cal un sistema de plantilles per generar els informes en PHP. Per aquesta finalitat he creat una classe a partir de la llibreria [FPDF](#).

Serveis

La capa de serveis recupera, crea, actualitza i suprimeix una part de les dades de la capa DAO i, si convé, fa els càlculs convenients per enviar-les a la capa de model, segons les peticions que li arriben. Igual que en el model de la presentació, els serveis hereten d'un servei genèric, *MainService.php*, que conté alguns mètodes utilitzats per varis serveis, com ara retornar l'objecte escola (*School*).

DAO

A la capa DAO hi ha totes les entitats que formen part del projecte (estudiants, classes, mestres...). Per fer els accessos a la base de dades, s'ha utilitzat un framework de mapatge ORM ([Doctrine](#)), similar a Hibernate per Java. Així, les interaccions dels serveis i entitats amb la base de dades (creació, recuperació, eliminació i actualització) són directes.



5. Disseny

5.1. Interfície d'usuari

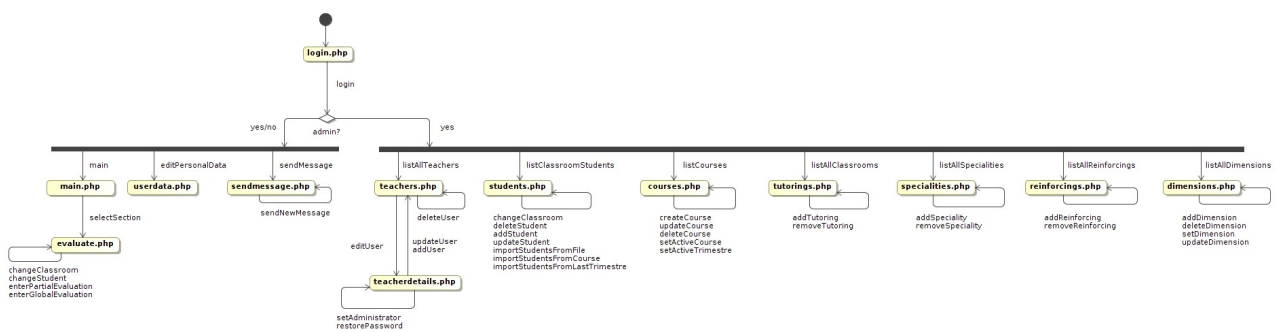
Navegació

L'esquema de la navegació conté les diferents pàgines i la navegació resultant de cadascuna de les operacions. És important perquè, a banda de mostrar les pàgines, també mostra la interacció del sistema amb l'exterior, essent un primer pas per definir les operacions de sistema.

Tal com es pot veure, la navegació es divideix en dues branques, depenent de si l'usuari és administrador o no.

Una vegada l'usuari està registrat, gairebé sempre pot navegar directament a qualsevol de les pàgines.

El nom assignat a les pàgines és important. D'una banda, a nivell extern cal que expliqui per si mateix la funció de la pàgina. D'altra banda, a nivell intern determina el nom de la resta de recursos associats a la pàgina (arxius CSS, Javascript, controladors...).

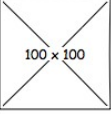


Disseny

S'adjunten alguns dels sketches de les pàgines. La resta es poden trobar a l'annex.

evaluate.php:

Header



Classroom ▼

Scope 1

Area 1

Dimension 1 A MA M MB B NV

Dimension 2 A MA M MB B NV

Dimension 3 A MA M MB B NV

Dimension 4 A MA M MB B NV

Scope 2

Area 2

Dimension 5 A MA M MB B NV

Dimension 6 A MA M MB B NV

Dimension 7 A MA M MB B NV

Dimension 8 A MA M MB B NV

Observacions

Observacions

Student 1

Student 2

Student 3

teachers.php

Header

Nom	<input type="text" value="name"/>
Cognoms	<input type="text" value="surnames"/>
Correu electrònic	<input type="text" value="email"/>
Telèfon	<input type="text" value="phone"/>
Nom d'usuari	<input type="text" value="username"/>

Administrador

S'adjunta també una pàgina amb el disseny final.

evaluate.php

Escola Josep Oriols Roca
Informe trimestral Curs: 2016-2017 | Trimestre: 1 Lluís Autonell Roger

1rA
Badredine Aeul

Àmbit lingüístic

Llengua Catalana

Comunicació oral A MA M MB B NV

Comprensió lectora A MA M MB B NV

Expressió escrita A MA M MB B NV

Hàbit lector i escriptor A MA M MB B NV

Ús de la llengua A MA M MB B NV

Qualificació Global AE AN AS NA NV

Llengua Castellana

Comunicació oral A MA M MB B NV

Comprensió lectora A MA M MB B NV

Hàbit lector i escriptor A MA M MB B NV

Ús de la llengua A MA M MB B NV

Qualificació Global AE AN AS NA NV

Àmbit matemàtic

Matemàtiques

Resolució de problemes A MA M MB B NV

Raonament i prova A MA M MB B NV

Connexions A MA M MB B NV

Comunicació i representació A MA M MB B NV

Qualificació Global AE AN AS NA NV

Àmbit de coneixement del medi

Coneixement del medi natural, social i cultural

Món actual A MA M MB B NV

5.2. Operacions de sistema

Són les operacions que rep el sistema a conseqüència d'accions externes, ja sigui per la interacció amb l'usuari, la càrrega d'una pàgina...

Són importants perquè ens ofereixen un contracte entre la petició d'entrada i l'estat del sistema després de dur a terme l'operació.

@pre el sistema enregistra l'intent de login.

@post si existeix, retorna l'identificador de l'usuari especificat, fals en cas contrari.

int **login** (String username, String password)

@pre cert.

@post cap usuari té accés al sistema.
void **logout** ()

@pre existeixen l'estudiant, el trimestre, el curs, la dimensió i el tipus de valoració parcial.
@post existeix una avaluació parcial amb el valor especificat assignada a l'estudiant, que correspon a la dimensió, trimestre i curs especificats.
void **enterPartialEvaluation** (int studentId, int trimestre, int courseId, int dimensionId, PartialEvaluationType mark)

@pre existeixen l'estudiant, el trimestre, el curs, l'àmbit i el tipus de valoració global. El trimestre pot ser nul (avaluació global de final de curs)
@post existeix una avaluació global amb el valor especificat assignada a l'estudiant, que correspon a l'àmbit, trimestre i curs especificats.
void **enterGlobalEvaluation** (int studentId, int trimestre, int courseId, int scopeId, GlobalEvaluationType mark)

@pre existeixen el curs, l'estudiant i la secció especificats.
@post retorna una llista amb els àmbits, dimensions, valoracions globals i parcials corresponents a l'estudiant durant el curs especificat.
List<Scope, Dimension, PartialEvaluation, GlobalEvaluation> **getCourseEvaluation** (int courseId, int studentId, int sectionId)

@pre existeixen el trimestre, el curs, l'estudiant i la secció especificats.
@post retorna una llista amb els àmbits, dimensions, valoracions globals i parcials corresponents a l'estudiant durant el trimestre i el curs especificats.
List<Scope, Dimension, PartialEvaluation, GlobalEvaluation> **getTrimestreEvaluation** (int trimestre, int courseId, int studentId, int sectionId)

@pre existeix l'identificador d'usuari.
@post existeix un usuari amb l'identificador, nom, cognoms, correu electrònic i contrasenya especificats.
void **updatePersonalData** (int userId, String name, String surnames, String email, String password)

@pre no existeix cap curs amb el nom especificat.
@post existeix un curs amb el nom especificat.
void **createCourse** (SchoolYear name)

@pre existeix un curs amb l'identificador especificat i el nom no correspon amb cap altre curs.
@post existeix un curs amb l'identificador i el nom especificats.
void **updateCourse**(int courseId, SchoolYear name)

@pre existeix un curs amb l'identificador especificat i no hi ha cap valoració corresponent a aquest curs.
@post no existeix cap curs amb aquest identificador.
void **deleteCourse**(int courseId)

@pre no existeix cap usuari amb el nom d'usuari especificat.
@post existeix un usuari amb les dades especificades.
int **addUser** (String name, String surnames, String email, String username)

@pre existeix un usuari amb l'identificador especificat i no té cap valoració entrada.
@post no existeix cap usuari amb aquest identificador.
void **deleteUser** (int userId)

@pre existeixen l'usuari, la classe, el trimestre i el curs especificats, i l'usuari no té la tutoria d'aquesta classe en aquest trimestre i curs.
@post l'usuari té assignada la tutoria de la classe en el trimestre i curs especificats.
void **addTutoring** (int userId, int classroomId, int trimestre, int courseId)

@pre existeixen l'usuari, classe, trimestre i curs especificats, l'usuari té assignada la tutoria de la classe en aquest trimestre i curs, i l'usuari no ha fet cap valoració d'aquesta classe en aquest trimestre i curs.
@post l'usuari no té assignada la tutoria de la classe en aquest trimestre i curs.
void **removeTutoring** (int userid, int classroomId, int trimestre, int courseId)

@pre existeixen l'usuari, àrea, trimestre i curs especificats, i l'usuari no és especialista d'aquest àrea en aquest trimestre i curs.
@post l'usuari és especialista d'aquesta àrea en aquest trimestre i curs.
void **addSpeciality** (int userId, int areaId, int trimestre, int courseId)

@pre existeixen l'usuari, àrea, trimestre i curs especificats, l'usuari és especialista d'aquesta àrea en aquest trimestre i curs, i no ha fet cap valoració d'aquesta àrea en aquest trimestre i curs.
@post l'usuari no és especialista d'aquesta àrea en aquest trimestre i curs.
void **removeSpeciality** (int userid, int areaId, int trimestre, int courseId)

@pre l'administrador i l'usuari no són iguals.
@post l'usuari té el rol d'administrador si el paràmetre *administrator* és cert, no el té en cas contrari.
void **setAdministrator** (int adminId, int userId, boolean administrator)

@pre existeixen l'usuari, trimestre i curs especificats. L'usuari no té entrada cap valoració com a mestre de reforç en aquest trimestre i curs.
@post l'usuari té el rol de mestre de reforç per aquest trimestre i curs si *reinforceTeacher* és cert, no el té en cas contrari.
void **setReinforceTeacher** (int userId, boolean reinforceTeacher, int trimestre, int courseId)

@pre existeix l'arxiu especificat i té el format correcte. El trimestre actual és el primer i no hi ha cap estudiant de nivell P3 en el curs actual.
@post existeixen els alumnes continguts en l'arxiu i estan assignats al primer trimestre i al curs actual en el nivell de P3.
List<Student> **importStudentsFromFile** (String file)

@pre el trimestre actual és el primer i el curs actual no és el primer.
@post els estudiants assignats al tercer trimestre en el cas de primària i al segon trimestre en el cas d'educació infantil en el curs especificat, i que cursaven nivells < 6º, estan assignats al primer trimestre del curs actual a un nivell superior i al mateix grup.
List<Student> **importStudentsFromLastCourse** ()

@pre el trimestre actual no és el primer
@post els estudiants assignats al trimestre anterior estan assignats al trimestre actual i a la mateixa classe.
List<Student> **importStudentsFromLastTrimestre** ()

@pre existeixen l'estudiant, el trimestre i el curs especificats, l'estudiant no té cap valoració per aquest trimestre i curs.

```

@post l'estudiant està assignat al trimestre i curs.
void setStudentTrimestre (int studentId, int trimestre, int courseId)

@pre els alumnes no tenen cap valoració.
@post els alumnes no existeixen al sistema i retorna la llista d'alumnes eliminats.
List<Student> removeStudents(int[] studentId)

@pre l'usuari existeix.
@post l'usuari té assignada la contrasenya per defecte.
void restorePassword (int userId)

@pre la classe existeix.
@post retorna les estadístiques de tots els cursos per la classe especificada.
data getClassroomData (int classroomId)

@pre existeixen la classe i el curs especificats.
@post retorna una llista amb els informes de tots els alumnes d'aquesta classe i
aquest curs.
List<Report> printClassroomReports (int classroomId, int courseId)

@pre existeixen l'estudiant i el curs especificats.
@post retorna l'informe amb les valoracions d'aquest estudiant.
Report printStudentReport (int studentId, int courseId)

@pre existeix l'àrea especificada, i no existeix cap dimensió dins d'aquesta àrea amb
aquest nom i descripció.
@post existeix una dimensió dins d'aquesta àrea amb aquest nom i descripció.
void addDimension (int areaId, String name, String description)

@pre existeix la dimensió i no té cap valoració assignada.
@post no existeix aquesta dimensió.
void deleteDimension (int dimensionId)

@pre existeixen la dimensió, cicle i curs especificats. La dimensió no està assignada
a aquest curs o bé hi està assignada però no té cap valoració en aquest curs.
@post la dimensió està assignada a aquest curs segons el valor del paràmetre assign.
void setDimension (int dimensionId, int cycleId, int courseId, boolean assign)

@pre existeix la dimensió dins l'àrea i cicle especificats i no hi ha cap altra dimen-
sió en aquesta àrea amb el nom i descripció especificats.
@post existeix una dimensió en aquesta àrea amb aquest nom i aquesta descripció.
void updateDimension (int areaId, int cycleId, int dimensionId, String name, String
description)

@pre true.
@post els administradors del sistema han rebut un missatge amb el text especificat.
void sendMessage (String text)

@pre existeix el trimestre.
@post el trimestre està registrat com a trimestre actiu.
void setActiveTrimestre (int trimestre)

@pre true
@post retorna una llista amb tots els mestres registrats al sistema.
List<Teacher> listAllTeachers()

```


@pre existeixen el trimestre i el curs especificats.
 @post retorna una llista de totes les classes i els tutors assignats en aquest trimestre i curs.
 List<Classroom, List<Tutor>> **listAllClassrooms**(int trimestre, int courseId)

@pre existeixen el trimestre i el curs especificats.
 @post retorna una llista de totes les àrees d'aquest curs i trimestre amb especialista assignat i els especialistes assignats a elles.
 List<Area, List<Specialist>> **listAllSpecialities**(int trimestre, int courseId)

@pre existeixen el trimestre i el curs especificats.
 @post retorna una llista amb els estudiants assignats en aquesta classe, trimestre i curs.
 List<Student> **listAllClassroomStudents**(int classroomId, int trimestre, int courseId)

@pre existeix l'usuari especificat.
 @post retorna una llista dels rols actuals d'aquest usuari.
 List<Role> **getAllUserRoles** (int userId)

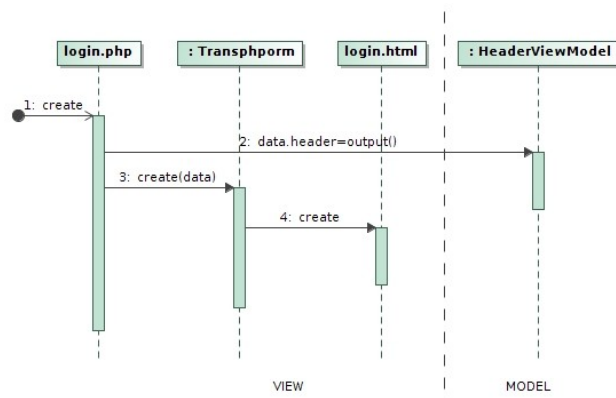
@pre cert
 @post retorna el curs actiu
 Course **getActiveCourse**()

5.3. Diagrames d'interacció

És important mostrar tant les interaccions que es donen en la part dels serveis i de les entitats com en la part de presentació. De fet és en la presentació on les interaccions són més complexes.

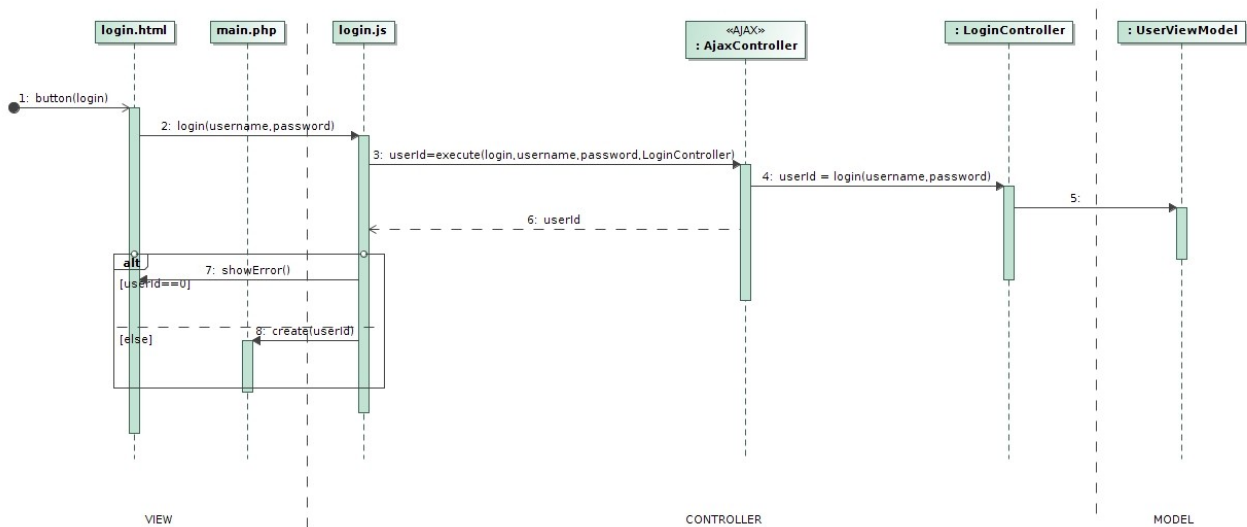
Registrar-se al sistema (login)

Aquesta seria la seqüència de la banda de la presentació en el moment de crear la pàgina.



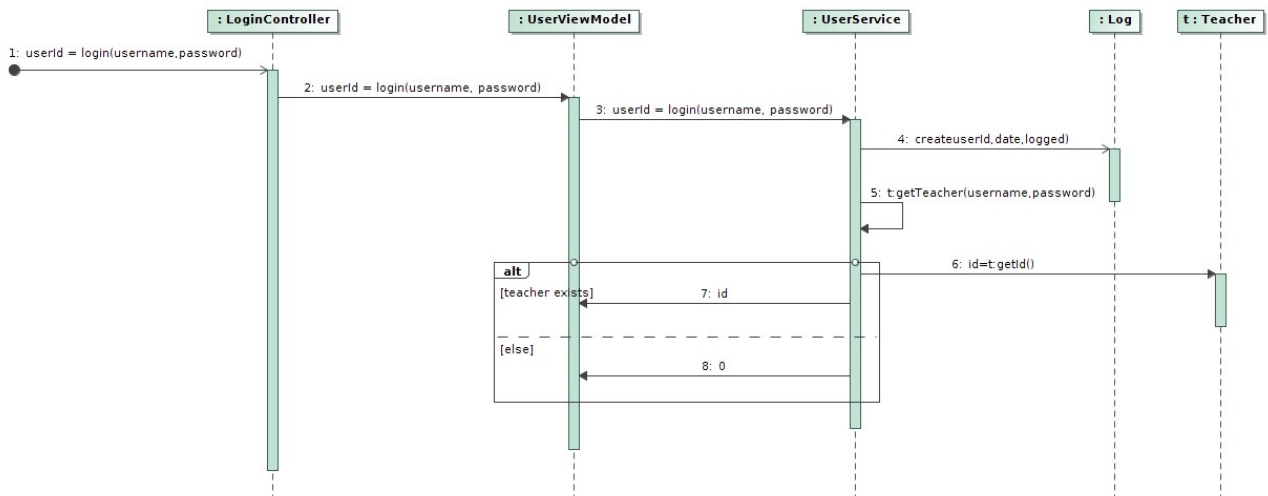
En el moment de crear la pàgina (*login.php*), es demanen les dades de la capçalera a *HeaderViewModel*. En aquest cas són les úniques dades dinàmiques que ens interessin. Les enviem al framework Transphpform per tal que ens creï la pàgina que veurà l'usuari (en aquest cas l'he anomenada *login.html* per diferenciar-la, però en realitat segueix essent *login.php*). En aquest cas no cal un controlador, la pàgina rep les dades directament del model.

Aquesta seria la seqüència de la banda de la presentació quan l'usuari ha entrat les dades i prem el botó *Acceptar*.



Tal com es pot veure en el diagrama, aquest cas és força diferent, ja que la petició no la fa la pàgina en el moment de crear-se sinó que la fa l'usuari. És la funció de Javascript (dins de *login.js*) la que fa la petició a través d'AJAX al controlador *AjaxController* perquè li envii les dades. Aquest envia la petició al controlador corresponent (*LoginController*) que, per la seva banda, envia la petició al model de la vista. Aquest cridarà els serveis necessaris per poder obtenir les dades, les formatarà convenientment en forma de vector i les retornarà *LoginController*, que les enviarà a *AjaxController*, que les enviarà a l'script *login.js*. En cas que les dades d'usuari i contrasenya fossin incorrectes (*userid=0*), es mostrarà l'error per pantalla a la mateixa pàgina de *login.php*. En cas contrari, es carregarà la pàgina principal, *main.php*.

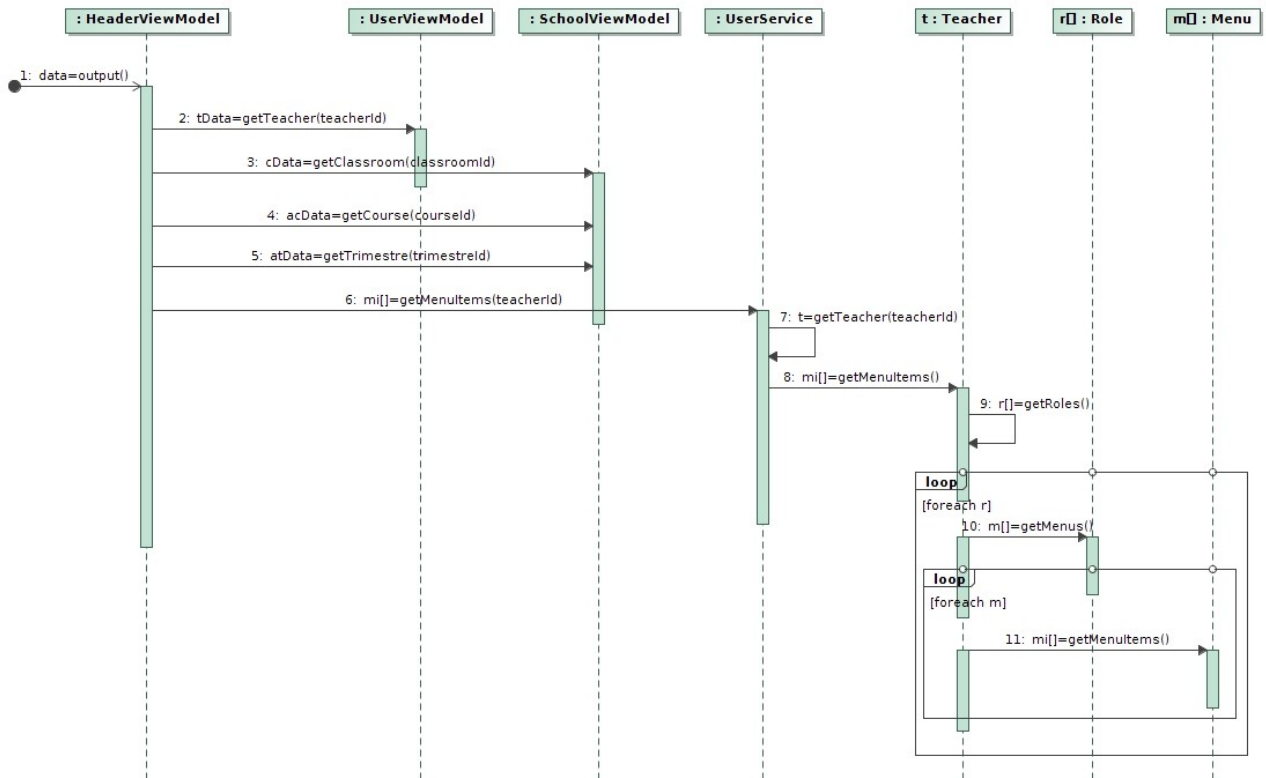
Aquesta seria la interacció a la banda dels serveis:



UserViewModel envia la petició de login a *UserService*. Aquest carrega les entitats necessàries per dur a terme l'operació i conté la lògica per comprovar que les dades entrades siguin correctes. En qualsevol cas, genera una línia de registre (*Log*) que ens indica que algú ha intentat entrar al sistema, ja sigui amb èxit o no. Finalment retorna les dades a *UserViewModel*, que les enviarà tal com havíem vist al diagrama anterior.

Capçalera

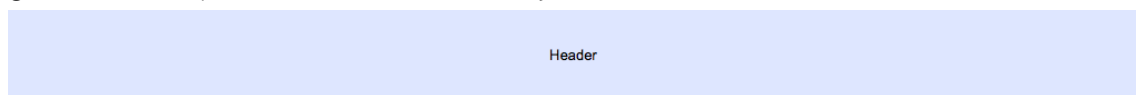
S'inclou també la seqüència corresponent a la recuperació de dades de la capçalera a partir del model. Aquestes són: nom de l'usuari registrat, curs actual, trimestre actual i elements del menú (que varien segons el rol d'usuari). Per obtenir els elements de menú ha calgut crear 2 entitats no incloses en l'anàlisi: *Menu* (lligat als rols d'usuari) i *MenuItems* (lligat als menús). Ambdues estan especificades al diagrama de classes.



Pàgina principal (main)

En aquest cas mostrem la interacció que es produeix quan es carrega la pàgina, a partir que les dades arriben al controlador. Encara que no s'hagi inclòs, anteriorment s'han carregat les dades de la capçalera, com sempre.

La pàgina mostra les diferents seccions en les que el mestre està registrat en l'actualitat, ja siguin tutories, especialitats o classes de reforç individual.



Tutoria

[- Classe 1](#)

[- Classe 2](#)

Especialitat

[- Àrea 1](#)

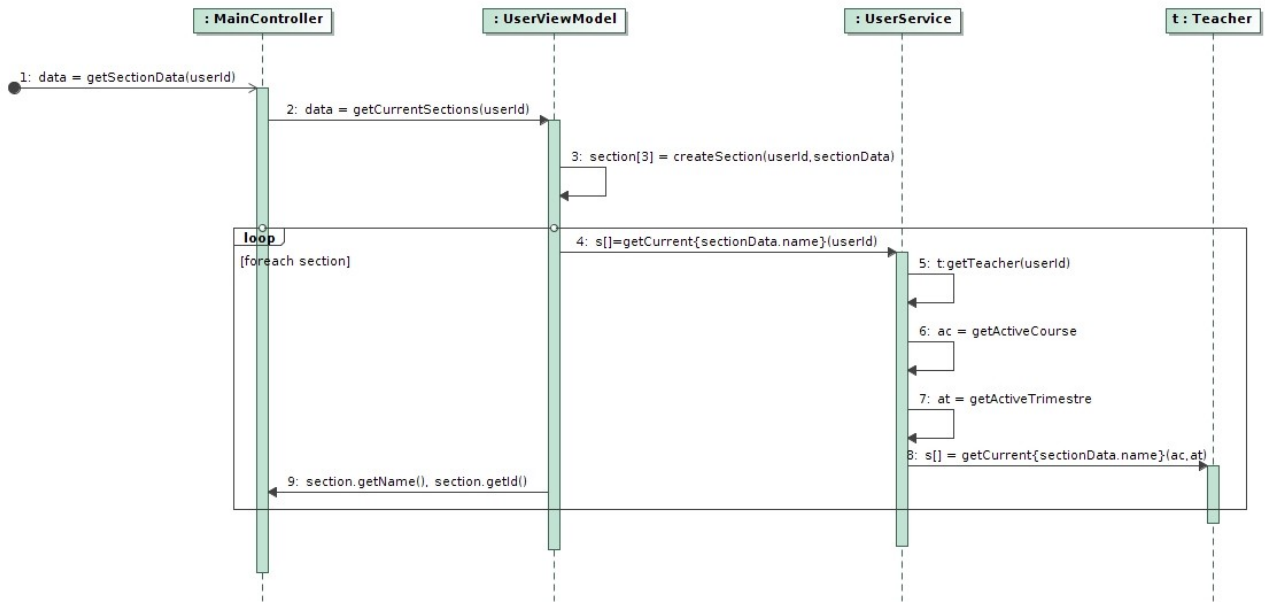
[- Àrea 2](#)

Reforç

[- Educació especial](#)

[- Audició i llenguatge](#)

Quan l'usuari premi una de les seccions en les que està registrat, es carregarà la pàgina d'avaluacions on podrà modificar les avaluacions de cada alumne en el curs i trimestre actual de la secció corresponent.



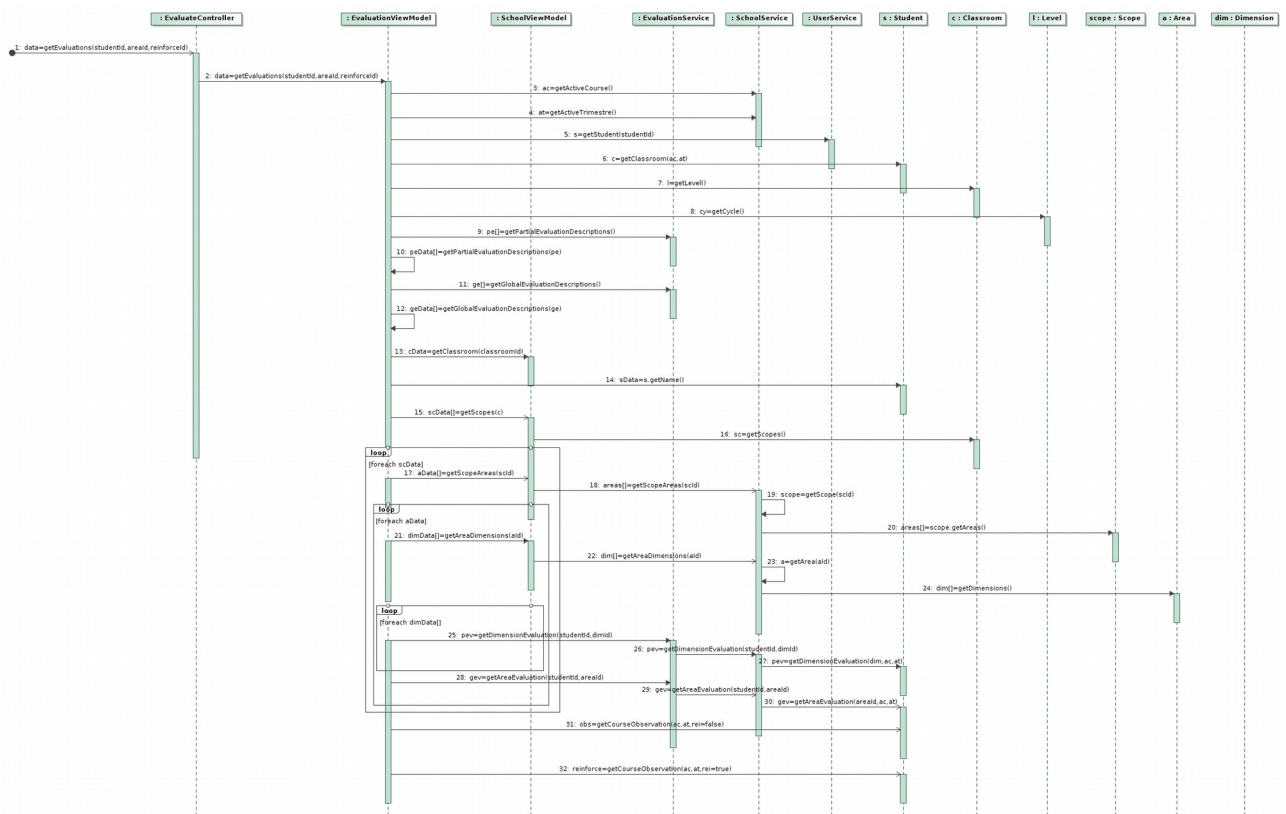
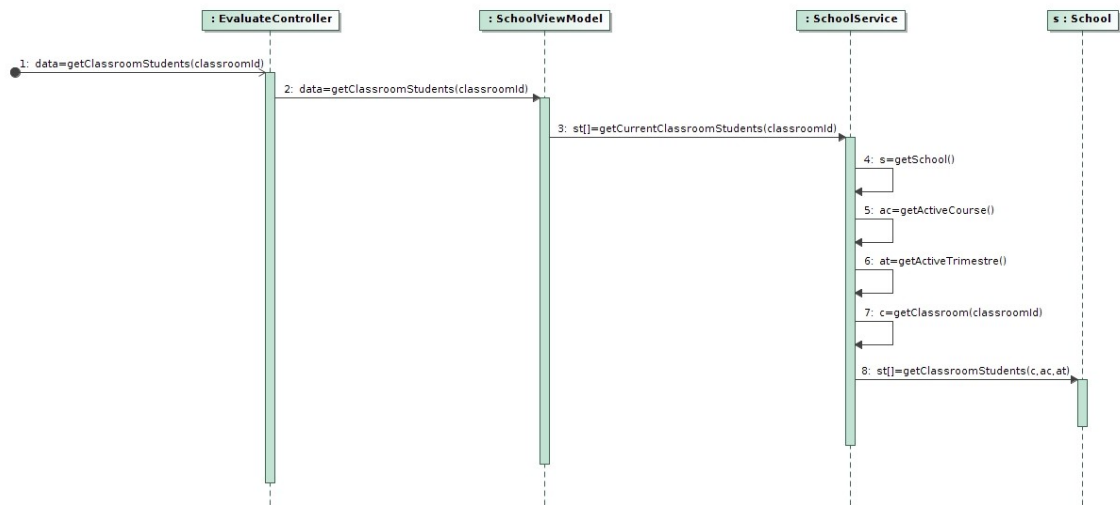
L'encarregat de recuperar les dades i formatar-les és *UserViewModel*. Aquest recupera les 3 seccions corresponents al mestre registrat a través d'*UserService*, que al seu torn les demana a l'entitat *Teacher*. Destacar que el nom del mètode que executa *UserService* s'obté de forma dinàmica a partir del nom de la secció (*tutorings*, *specialities* i *reinforcings*).

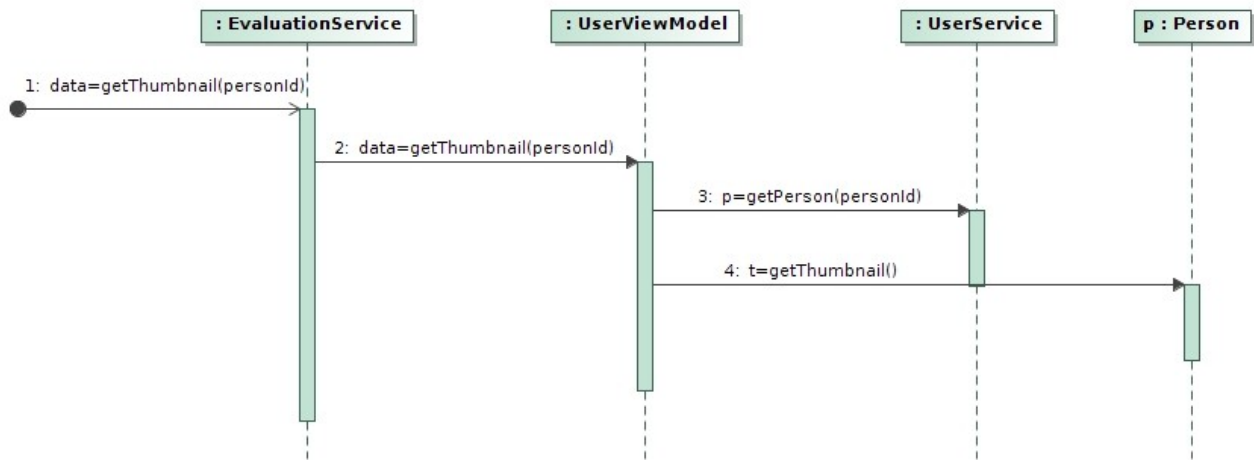
Pàgina d'entrada d'avaluacions (evaluate)

Aquesta és una de les pàgines més complexes de l'aplicació. La seqüència en el moment de la creació és la següent:

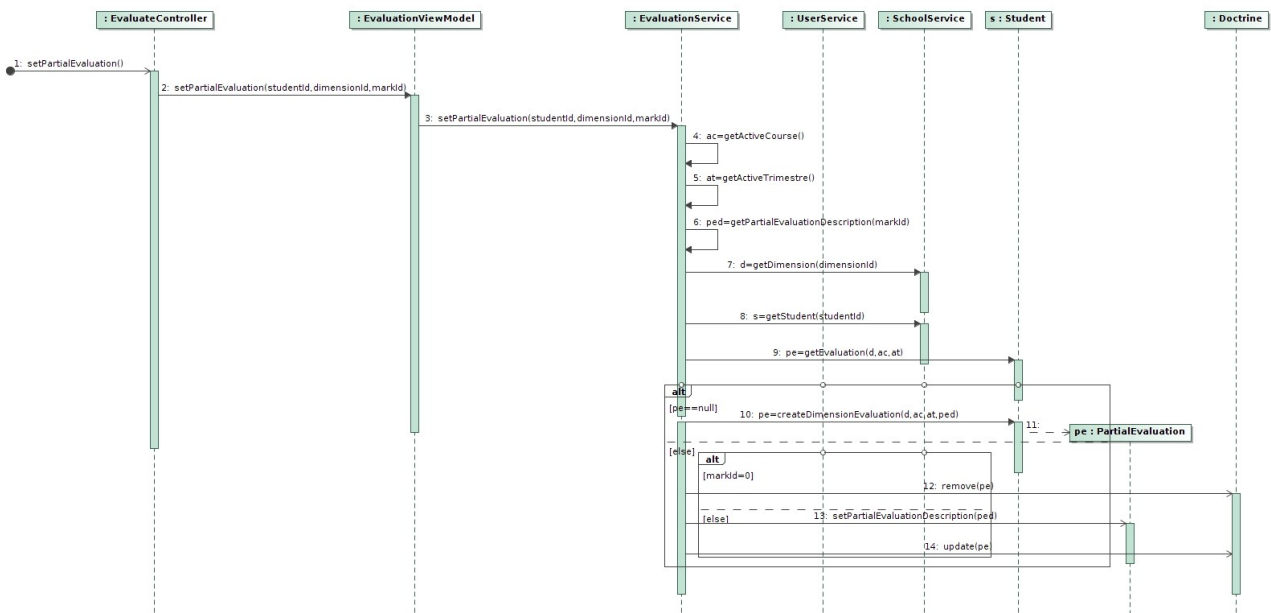
- Com sempre, s'obtenen les dades de la capçalera
- En cas que la secció sigui una tutoria, s'obté la llista d'alumnes de la classe corresponent. En cas contrari, es carrega la classe per defecte i es permet navegar per les diferents classes de l'escola.
- Se selecciona el primer alumne i se'n carreguen les avaluacions i la imatge.

Els diagrames d'interacció a partir del controlador són els següents:





Dins de la mateixa pàgina, el mestre pot canviar les avaluacions, tant si són parcials (dimensió), globals (àrea) o observacions. S'inclou la seqüència corresponent a les valoracions parcials:

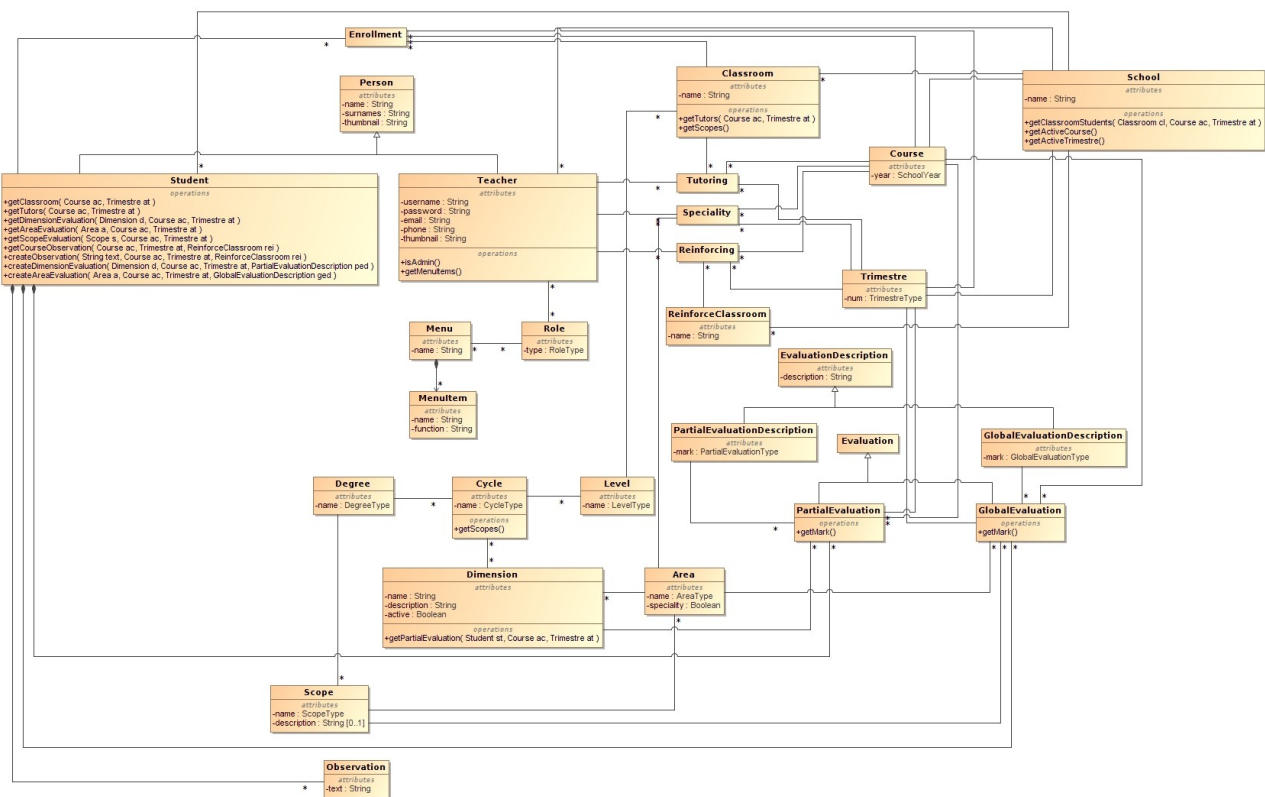


5.4. Diagrames de classes

Diagrama d'entitats

El diagrama de classes parteix del model de domini fet anteriorment, al qual s'han anat afegint les operacions de sistema i les trobades en els diagrames d'interacció. També s'han afegit algunes entitats que no existien al model d'anàlisi, com són les referents als menús.

La tasca de creació dels objectes en primera instància es delega a la capa de serveis. Aquests no estan inclosos en el diagrama d'entitats i es mostren més endavant.



Models de vista

Recordem que la responsabilitat dels models de vista és la de traduir les dades que generen els serveis en un format vectorial que permeti a les plantilles generar el codi HTML que es presentarà a l'usuari.

Per tant, tenen importància en les operacions corresponents a la recuperació de dades que

han de ser mostrades a l'usuari. No és així, en canvi, per les operacions d'actualització, i algunes es deleguen directament a la cap de serveis des dels controladors.

MainViewModel
<i>attributes</i>
-entityManager : EntityManager
<i>operations</i>
+setSession() : void
+getSessionVar(String sessionVar) : String
+addData(Array data) : void
+getData() : Array

UserViewModel
<i>operations</i>
+login(String username, String password) : int
+getStudent(int studentId) : Array
+getTeacher(int teacherId) : Array
+importStudentsFromLastTrimestre() : Array
+importStudentsFromFile(String file) : Array
+importStudentsFromLastCourse() : Array
+listAllTeachers() : Array
+getCurrentSections(int teacherId) : Array
+getMenuItems(int userId) : Array
+getThumbnail(int personId) : String

SchoolViewModel
<i>operations</i>
+listAllSpecialities(int trimestre, int courseId) : Array
+listAllClassrooms(int trimestre, int courseId) : Array
+getActiveCourse()
+getActiveTrimestre()
+getClassroomStudents(int classroomId, int trimestre, int courseId) : Array
+getReinforceClassroom(int reinforceId) : Array
+getClassroom(int classroomId) : Array
+getAreaDimensions(int areald, int cycleId) : Array
+getScopes(int classroomId) : Array
+getScopeAreas(int scopeld, int areald) : Array
+getArea(int areald) : Array
+getClassrooms()
+listAllCourses()

EvaluationViewModel
<i>operations</i>
+getClassroomData(int classroomId) : Array
+getPartialEvaluation(int studentId, int dimensionId, int courseId, int trimestre) : PartialEvaluation
+getGlobalEvaluation(int studentId, int areald, int courseId, int trimestre) : GlobalEvaluation
+getCourseEvaluations(int courseId, int studentId, int sectionId) : Array
+getTrimestreEvaluations(int trimestre, int courseId, int studentId, int sectionId) : Array
+getEvaluations(int tudentId, int areald, int reinforceId, boolean includeSpecialities) : Array

ReportViewModel
<i>operations</i>
+getData() : Array

HeaderViewModel
<i>operations</i>
+output()

Serveis

MainService	
<i>attributes</i>	
-entityManager	: EntityManager
<i>operations</i>	
+getSchool()	: School
+getCourse(SchoolYear courseId)	: Course
+getActiveCourse()	: Course
+getTrimestre(int trimestre)	: Trimestre
+getActiveTrimestre()	: Trimestre

UserService	
<i>operations</i>	
+login(String username, String password)	: Teacher
+getTeacher(int teacherId)	: Teacher
+getPerson(int personId)	: Person
+getStudent(int studentId)	: Student
+getStudents(Array filter)	: Array
+updatePersonalData(int userId, String name, String surnames, String email, String password)	: void
+deleteTeacher(int teacherId)	: void
+addTutoring(int userId, int classroomId, int trimestre, int courseId)	: void
+removeTutoring(int userId, int classroomId, int trimestre, int courseId)	: void
+addSpeciality(int userId, int areald, int trimestre, int courseId)	: void
+removeSpeciality(int userId, int areald, int trimestre, int courseId)	: void
+addReinforceClassroom(int userId, int reinforceId, int trimestre, int courseId)	: void
+removeReinforceClassroom(int userId, int reinforceId, int trimestre, int courseId)	: void
+importStudentsFromFile(String file)	: Array
+importStudentsFromLastTrimestre()	: Array
+importStudentsFromLastCourse()	: Array
+updateStudent(int studentId, String name, String surnames, int classroomId)	: void
+removeStudents(Array studentId)	: Array
+restorePassword(int userId)	: void
+getAllTeachers()	: Array
+getCurrentTutorings(int teacherId)	: Array
+getCurrentSpecialities(int teacherId)	: Array
+getCurrentReinforcings(int teacherId)	: Array
+getMenuItems(int userId)	: Array
+updateTeacher(int teacherId, String name, String surnames, String email, String phone, String username, boolean isAdmin)	: void
+isAdmin(int teacherId)	: boolean

SchoolService	
<i>operations</i>	
+getCurrentTutorings()	: Array
+getCurrentSpecialities()	: Array
+getCurrentReinforcings()	: Array
+getClassroom(int classroomId)	: Classroom
+getClassrooms()	: Array
+getScope(int scopeld)	: Scope
+getScopes(int classroomId)	: Array
+getClassroomScopes(int classroomId)	: Array
+getArea(int areald)	: Area
+getScopeAreas(int scopeld, int areald)	: Array
+getReinforceClassroom(int reinforceId)	: ReinforceClassroom
+sendMessage(String text)	: void
+createCourse(SchoolYear name)	: void
+updateCourse(int courseId, SchoolYear name, boolean isActive)	: void
+deleteCourse(int courseId)	: void
+setActiveTrimestre(int trimestre)	: void
+getCurrentClassroomStudents(int classroomId)	: Array
+addDimension(int areald, String name, String description)	: void
+deleteDimension(int dimensionId)	: void
+setDimension(int dimensionId, int cycleId, int courseId, boolean assign)	: void
+updateDimension(int areald, int cycleId, int dimensionId, String name, String description)	: void
+getAreaDimensions(int areald, int cycleId)	: Array
+getDimension(int dimensionId)	: Dimension
+getCourses()	: Array
+getTrimestres()	: Array

EvaluationService	
<i>operations</i>	
+getPartialEvaluation(int studentId, int dimensionId, int courseId, int trimestre)	: PartialEvaluation
+setPartialEvaluation(int studentId, int trimestre, int courseId, int dimensionId, PartialEvaluationType mark)	: void
+getGlobalEvaluation(int studentId, int areald, int courseId, int trimestre)	: GlobalEvaluation
+setGlobalEvaluation(int studentId, int trimestre, int courseId, int scopeld, GlobalEvaluationType mark)	: void
+getPartialEvaluationDescription(PartialEvaluationType pedId)	: void
+getPartialEvaluationDescriptions()	: Array
+getGlobalEvaluationDescription(GlobalEvaluationType gedId)	: void
+getCourseEvaluations(int courseId, int studentId, int sectionId)	: Array
+getGlobalEvaluationDescriptions()	: Array
+getTrimestreEvaluations(int trimestre, int courseId, int studentId, int sectionId)	: Array
+setObservation(int studentId, String text, int reinforceId)	: void
+getDimensionEvaluation(int studentId, int dimensionId, int courseId, int trimestreId)	: void
+getScopeEvaluation(int studentId, int scopeld, int courseId, int trimestreId)	: void

5.5. Patrons de disseny utilitzats

Bàsicament s'han utilitzat cinc patrons GRASP a l'hora d'assignar responsabilitats:

- **Expert:** segurament el patró més tingut en compte, intentant esbrinar en cada operació quina entitat era la coneixedora de la informació.
- **Alta cohesió:** s'ha intentat que les responsabilitats que té cada classe siguin el màxim de cohesionades possible. Malgrat tot, hi ha dos nivells en els que ha estat difícil arribar a una alta cohesió: els arxius Javascript —en que un sol arxiu es cuida de la funcionalitat de tota una pàgina—, i la capa de serveis en general.
- **Baix acoplament:** s'ha fet èmfasi en que les diferents entitats estiguessin el mínim d'acoblades possible. També entre les diferents capes.
- **Creador:** en general s'ha deixat la responsabilitat de crear noves instàncies a la capa de serveis.
- **Façana:** un exemple clar de façana són els controladors de la capa de presentació, que reben totes les peticions tant de la seva vista associada com del controlador AJAX, i les distribueixen a altres capes.

5.6. Disseny de la base de dades

Les entitats es relacionen directament amb la base de dades a través de l'ORM [Doctrine](#). La relació es fa a través d'anotacions dins de la mateixa declaració de cadascuna de les classes.

Disseny lògic

Relació de totes les entitats per ordre alfabètic

Area (id, name, scope_id, speciality)
• scope_id és clau forana de Scope (id)

Classroom (id, name, level_id, school_id)
• level_id és clau forana de Level (id)
• school_id és clau forana de School (id)

Course (id, year)

Cycle (id, name, degree_id)

- degree_id és clau forana de Degree (id)

Degree (id, name)

Dimension (id, name, description, active, area_id)

- area_id és clau forana de Area (id)

Enrollment (id, student_id, classroom_id, course_id, trimestre_id)

- student_id és clau forana de Student (id)
- classroom_id és clau forana de Classroom (id)
- course_id és clau forana de Course (id)
- trimestre_id és clau forana de Trimestre (number)

Evaluation (id)

EvaluationDescription (id, description)

GlobalEvaluation extends Evaluation (student_id, course_id, trimestre_id, area_id, scope_id, globalevaluationdescription_id)

- student_id és clau forana de Student (id)
- course_id és clau forana de Course (id)
- trimestre_id és clau forana de Trimestre (number)
- area_id és clau forana de Area (id)
- scope_id és clau forana de Scope (id)
- globalevaluationdescription_id és clau forana de GlobalEvaluationDescription (id)

GlobalEvaluationDescription extends EvaluationDescription (mark)

Level (id, name, cycle_id)

- cycle_id és clau forana de Cycle (id)

Log (id, username, date, logged)

Menu (id, name)

MenuItem (id, menu_id, name, function, order)

- menu_id és clau forana de Menu (id)

Observation (id, text, course_id, trimestre_id, reinforceclassroom_id, student_id)

- course_id és clau forana de Course (id)
- trimestre_id és clau forana de Trimestre (number)
- reinforceclassroom_id és clau forana de ReinforceClassroom (id)
- student_id és clau forana de Student (id)

PartialEvaluation extends Evaluation (student_id, course_id, trimestre_id, dimension_id, partialevaluationdescription_id)

- student_id és clau forana de Student (id)
- course_id és clau forana de Course (id)
- trimestre_id és clau forana de Trimestre (number)
- dimension_id és clau forana de Dimension (id)
- partialevaluationdescription_id és clau forana de PartialEvaluationDescription (id)

PartialEvaluationDescription extends EvaluationDescription (mark)

Person (id, name, surnames, thumbnail, school_id)

- school_id és clau forana de School (id)

ReinforceClassroom (id, name, school_id)

- school_id és clau forana de School (id)

Reinforcing (id, teacher_id, reinforceclassroom_id, course_id, trimestre_id)

- teacher_id és clau forana de Teacher (id)
- reinforceclassroom_id és clau forana de ReinforceClassroom (id)
- course_id és clau forana de Course (id)
- trimestre_id és clau forana de Trimestre (number)

Role (id)

School (id, name, active_course, active_trimestre)

- active_course és clau forana de Course (id)
- active_trimestre és clau forana de Trimestre (number)

Scope (id, name, description, degree_id)

- degreeid és clau forana de Degree (id)

Speciality (id, teacher_id, area_id, course_id, trimestre_id)

- teacher_id és clau forana de Teacher (id)
- area_id és clau forana de Area (id)
- course_id és clau forana de Course (id)
- trimestre_id és clau forana de Trimestre (number)

Student extends Person ()

Teacher extends Person (email, phone, username, password)

Trimestre (number)

Tutoring (id, teacher_id, classroom_id, course_id, trimestre_id)

- teacher_id és clau forana de Teacher (id)
- classroom_id és clau forana de Classroom (id)
- course_id és clau forana de Course (id)
- trimestre_id és clau forana de Trimestre (number)

Taules que relacionen entitats però que no contenen informació extra:

dimensions_cycles (dimension_id, cycle_id)

- dimension_id és clau forana de Dimension (id)
- cycle_id és clau forana de Cycle (id)

hasroles (teacher_id, role_id)

- teacher_id és clau forana de Teacher (id)
- role_id és clau forana de Role (id)

roles_menus (role_id, menu_id)

- role_id és clau forana de Role (id)

- menu_id és clau forana de Menu (id)

6. Implementació

6.1. Estàndards utilitzats

A l'hora de programar, és important seguir unes pautes que siguin el màxim de comunes per la comunitat de programadors. El fet de seguir uns protocols ens ajuda a millorar tant la intel·ligibilitat del nostre codi com les bones pràctiques de programació.

El punt de partida a l'hora de programar en PHP han estat els estàndards suggerits a [PHP-FIG](#). Aquests abarquen aspectes com: espais de noms, etiquetes, definicions (variables, funcions, mètodes, classes...), autoloaders...

També s'ha posat èmfasi que el codi sigui el màxim d'autoexplicatiu, per tal que necessiti el mínim de comentaris per la seva comprensió.

6.2. Entorn de treball

Infraestructura utilitzada

L'eina de treball principal ha estat un portàtil amb processador Intel® i7, amb 4GB de RAM i disc dur SSD de 128GB. El sistema operatiu era una distribució Ubuntu 16.04 de 64 bits.

Eines externes

Per fer el mapatge entre entitats i relacions de la base de dades s'ha utilitzat el framework [Doctrine](#).

Com a programari per crear les pàgines HTML a partir de dades en format vector, s'ha utilitzat [Transhporm](#).

Per generar els informes en format PDF s'ha utilitzat la llibreria [FPDF](#).

Per enviar correus electrònics des de l'aplicació s'ha utilitzat la llibreria [PHPMailer](#).

Per fer les còpies de seguretat, s'ha utilitzat [Git](#).

Per fer la programació dels arxius PHP s'ha utilitzat [Eclipse](#).

Per la resta de programació (Javascript, CSS...) s'ha utilitzat l'editor [Bluefish](#).

Com a servidor de pàgines PHP, s'ha utilitzat [Apache](#).

Com a motor de base de dades, s'ha utilitat [MySQL](#).

Instal·lació i configuració del programari

Els frameworks externs utilitzats (Doctrine, Transhporm) s'han instal·lat a partir de Composer.

Eclipse s'ha instal·lat descarregant-lo de l'última versió (Neon) i descomprimint l'arxiu.

Les llibreries PHPMailer i FPDF s'han instal·lat descarregant l'última versió de la pàgina web.

Bluefish, Apache, Git i MySQL s'han instal·lat directament des d'Ubuntu (`apt-get install`).

6.3. Estructura de directoris

/

Al directori arrel hi ha les pàgines php que carrega el navegador. També hi ha 2 arxius necessaris per treballar amb Doctrine, **bootstrap.php** i **cli-config.php**, i la utilitat per modificar la configuració general de l'aplicació (**config.php**).

config

Conté els arxius necessaris per configurar l'aplicació, les definicions de constants i la classe que permeten carregar automàticament les classes a mesura que es necessiten.

images

logos

Logos utilitzats als informes

thumbnails

Miniatures dels estudiants.

js

Arxius de Javascript. A cada pàgina li correspon un arxiu. A part, hi ha un arxiu genèric **scripts.js**, amb funcions que es poden cridar des de qualsevol pàgina, i un arxiu **compact-nav.js** que controla el comportament dels menús.

presentation

controller

Controladors de la capa de presentació. Cada pàgina té el seu controlador. També hi ha un controlador general d'on hereten tots els altres (**Controller.php**), i un controlador que rep les peticions AJAX des de Javascript (**AjaxController.php**).

css

Arxius de maquetació. Cada pàgina té el seu. A més, hi ha un arxiu genèric (**styles.css**) i un pels menús (**compact-nav.css**).

images

Imatges utilitzades internament per la capa de presentació.

model

Models de vista.

template

Arxius de plantilla. Cada pàgina necessita 2 arxius, un XML amb les marques HTML i un TSS amb les instruccions per insertar les dades dinàmiques a la pàgina.

view

Generadors de codi HTML i PDF personalitzats.

service

Classes corresponents a la capa de serveis.

entities

Entitats derivades del diagrama de classes que s'ha construït durant el disseny.

utils

Classe estàtica amb funcions útils per diverses altres classes.

vendor

Llibreries i frameworks externs utilitzats durant el projecte (Doctrine, Transphorm, PHPMailer, FPDF...).

6.4. Consideracions

Espais de noms

Es considera una bona pràctica utilitzar espais de noms (namespace) per tal que cada classe quedi identificada de manera individual. En el projecte actual, s'ha tingut en compte el protocol que utilitza el carregador automàtic de classes (autoloader) que incorpora Doctrine i que ha estat l'escollit per carregar les classes dinàmicament. Perquè aquest funcioni cal que l'espai de noms segueixi escrupulosament l'estructura de directoris. Per aquesta raó els espais comencen en minúscula, malgrat no sigui el mètode aconsellat per l'estàndard PSR-4. L'alternativa hagués estat capitalitzar tots els directoris, el qual es va descartar ja que trenca

amb la metodologia de treball de la majoria de llenguatges de programació.

Impressió dels informes

Les opcions que es van plantejar en un principi a l'hora d'imprimir els informes era, o bé maquetar una pàgina web, o bé crear un PDF. Es va optar per la segona opció, ja que era molt important que els informes tinguessin el mateix format impresos des de qualsevol navegador, i aquest aspecte no estava clar que amb el format web estigués garantit.

La llibreria escollida per maquetar i crear els PDF va ser FPDF, ja que oferia una major senzillesa respecte PDFlib. No obstant, algunes d'aquestes funcionalitats, com el canvi de pàgina o els peus i capçaleres automàtics, no s'han pogut utilitzar degut a les característiques específiques dels informes, i s'han hagut de programar manualment.

7. Instal·lació

7.1. Requisits del sistema

Maquinari

Des del punt de vista del maquinari, els requisits del servidor on es vol instal·lar l'aplicació no són gaire exigents, tenint en compte que, el programa sol, només ocupa uns 15MB de disc dur. Aquesta mesura es pot veure incrementada notablement si assignem una foto a cada alumne.

Programari

Els requisits mínims del servidor des del punt de vista del programari són:

Apache 2.0

PHP 7.0

MySQL 5.7

7.2. Instal·lació i execució

Per procedir a la instal·lació cal seguir els següents passos:

- Descomprimir l'arxiu **infojor.zip** a la carpeta del servidor on desitgem.
- Comprovar que tenim accés de lectura a tots els arxius per tots els usuaris i que a més podem escriure a **config/dbparams.config**.
- Crear una base de dades amb el nom **infojor** i ordenació **utf8_unicode_ci**.
- Importar a la base de dades l'arxiu **infojor.sql**, que conté tant la informació de les taules, relacions, claus... com dades de prova.
- Executar **config.php** i entrar les dades corresponents a la configuració de la base de dades.

A partir d'aquí només cal que entrem al directori d'instal·lació des del nostre navegador i ja podrem fer funcionar l'aplicació.

L'arxiu **index.php** ens direccionarà a la pàgina de registre (**login.php**) si no estem registrats durant la sessió actual. En cas contrari, carregarà directament la pàgina **main.php**.

S'han habilitat alguns usuaris per fer proves:

- admin / 1234 Administrador i tutor de primària
- specialist / 1234 Especialista
- tutor2 / 1234 Tutor d'educació infantil
- multi / 1234 Tutor i especialista

Per provar les funcionalitats de l'aplicació també es pot fer a una versió que hi ha penjada a <http://carlescanellas.cat/informes-moja>.

8. Conclusions

8.1. Tasques pendents

En el moment del lliurament el projecte està en una fase avançada però no finalitzada. Queda pendent resoldre els següents casos d'ús:

- Importar alumnes des d'un fitxer.
- Importar alumnes de cursos i trimestres anteriors.
- Casos relacionats amb la generació d'estadístiques.

8.2. Objectius assolits

Pel que fa als objectius de cara a l'escola, es van aconseguir plenament, ja que en els terminis establerts els mestres van poder entrar les valoracions, i es van poder imprimir els informes amb els requisits establerts de bon principi. De fet, la valoració inicial tant dels mestres com de l'equip directiu després d'aquest primer lliurament ha estat molt positiva.

De cara al projecte com a treball de final de grau, s'han pogut fer els lliuraments parcials en les dates imposades de forma satisfactòria. S'han seguit tots els passos que estableix l'enginyeria del programari per la construcció d'aplicacions TIC. S'ha creat un producte de bona qualitat que ofereix la suficient flexibilitat per poder ser ampliat en un futur i adaptat a diferents situacions.

Hi ha un objectiu que no s'ha pogut aconseguir del tot, i és que en el moment del lliurament l'aplicació no està finalitzada del tot, ja que queden alguns casos d'ús per resoldre. No obstant, s'han resolt els més urgents i, segurament, alguns dels més complicats.

A nivell personal, he pogut aprofundir en un camp molt interessant i que crec que pot donar molt de sí per aplicacions de complexitat petita i mitjana, com és la creació de projectes en llenguatge PHP. Si bé ja tenia alguns fonaments d'aquest llenguatge gràcies a altres petits projectes en els que havia treballat, ha sigut molt interessant afrontar-ne un de complex seguint tots els passos de l'enginyeria del programari.

Un altre objectiu proposat era el d'iniciar-me en els procediments àgils d'una forma metòdica. Potser aquest és un dels aspectes en els que em queda més camí per recórrer. Tot i així, estic segur que és el camí que cal seguir en propers projectes.

8.3. Principals dificultats

Les principals dificultats que es van haver d'afrontar durant l'execució del projecte es corresponen amb força exactitud amb els riscos previstos durant les fases prèvies a l'anàlisi, afegides a algunes altres de més inesperades.

En general, ha estat un llast important la manca d'experiència en programació de projectes

complexos en PHP. Aquest fet ha implicat una inversió de temps molt important en diferents camps relacionats, com: el llenguatge en sí, recerca de llibreries associades, estructura del codi i les classes en general...

La manca de definició del projecte en alguns aspectes per part del client no ha estat tant important com es preveia. El bon treball de l'equip directiu i les reunions mantingues a l'inici del projecte van facilitar l'acceptació de la feina per part de la majoria, sinó tots, els mestres de l'escola.

La urgència per lliurar la primera fase del projecte operatiu a finals de novembre es va solucionar a base d'invertir moltes hores a l'inici del projecte, gràcies a la flexibilitat de poder aparcar altres projectes en aquell moment en marxa.

Una altra dificultat important que va sorgir va ser causada per la tria del motor de generació de pàgines HTML a partir de dades dinàmiques, també anomenat plantilles. Es va optar per confiar en el framework Transphorm davant d'altres opcions per la seva simplicitat i filosofia de funcionament. No obstant, s'hi han trobat importants mancances a causa de la impossibilitat de treballar amb dades profundament aniuades. Aquests casos han suposat una feina extra de programació d'un motor personalitzat.

8.4. Millores i ampliacions

Deixant de banda les funcionalitats que encara queden per implementar, aquesta és una aplicació que segurament admetrà diverses millores en diferents aspectes:

- Facilitar encara més la gestió de les valoracions per part dels mestres. De fet, ja hi ha hagut algunes peticions en aquest sentit, com per exemple crear una taula resum dels alumnes de cada classe amb la qualificació de cada dimensió.
- Adaptar l'aplicació perquè sigui més usable en pantalles de petit format, com telèfons mòbils. Bàsicament, consistirà en modificar els CSS.
- Facilitar la gestió per part dels administradors.

El pla de treball a partir d'aquest moment serà:

- Reunió amb l'equip directiu per valorar l'aplicació fins el moment i suggerir millores.
- Seguir implementant les funcionalitats pendents. Aquestes es preveu que estiguin llestes com a molt tard a finals del curs 2016-2017.

8.5. Visió de futur

En aquest moment l'aplicació està pensada per una escola en concret. Cal tenir en compte que el format dels informes trimestrals els decideix cada escola, ja que el Departament d'Ensenyament dóna molta flexibilitat en aquest sentit.

No obstant i, tenint en compte que les escoles necessiten fer aquests informes i que no se'ls ofereix cap eina realment útil amb la que puguin treballar, no és descartable que aquest projecte es pugui adaptar a diferents escoles, sempre que el format d'informe sigui similar al que han dissenyat a l'escola de Moià.

Una altra opció seria oferir el programa al Departament d'Ensenyament per tal que l'oferissin com a mètode estàndard per fer els informes trimestrals. En principi, aquesta opció es descarta, ja que pel que sembla una de les polítiques en aquest sentit és donar llibertat als centres perquè estipulin ells mateixos el tipus d'informe trimestral que desitgen donar als pares.

9. Fonts d'informació

9.1. Bibliografia

Larman, Craig (2013) *Applying UML and Patterns* (tercera edició). Pearson.

Cockburn, Alistair (2001) *Writing Effective Use Cases*. Addison-Wesley.

Fowler, Martin (1997) *Analysis Patterns: Reusable Object Models*. Addison-Wesley.

Generalitat de Catalunya (2016, juny) *Currículum i orientacions. Educació infantil. Segon cicle*. Disponible en línia.

<<http://ensenyament.gencat.cat/web/.content/home/departament/publicacions/colleccions/curriculum/curriculum-infantil-2n-cicle.pdf>>

Generalitat de Catalunya (2009, juny) *Currículum. Educació primària*. Disponible en línia.

<http://xtec.gencat.cat/web/.content/alfresco/d/d/workspace/SpacesStore/0031/034fc257-4463-41ab-b7f5-dd33c9982b4f/curriculum_ep.pdf>

Generalitat de Catalunya (2016, març) *Un currículum competencial per a l'educació primària*. Disponible en línia.

<http://ateneu.xtec.cat/wiki/form/wikiexport/_media/cursos/curriculum/inf_pri/cape/info_primaria_29_6.pdf>

També s'han utilitzat diverses publicacions incloses en els materials de diverses assignatures cursades a la UOC durant els meus estudis de GEI, especialment:

- Anàlisi i disseny amb patrons.
- Disseny de bases de dades.
- Enginyeria del programari

- Enginyeria del programari de components i sistemes distribuïts.
- Gestió de projectes.
- Programació orientada a objectes

9.2. Webs visitades:

<http://php.net/> Pàgina oficial del llenguatge de programació php.

<http://mysql.com/> Pàgina oficial del projecte MySQL, un gestor de bases de dades.

<https://www.phpmyadmin.net/> Pàgina oficial de Phpmyadmin, un administrador de bases de dades MySQL amb interfície d'usuari web.

<http://apache.org/> Pàgina oficial d'Apache Foundation, on desenvolupen un programari servidor de pàgines web.

<http://www.doctrine-project.org/> Pàgina oficial del projecte Doctrine, una sèrie de llibreries que permeten el mapatge entitat-relació (ORM).

<https://github.com/Level-2/Transphporm/wiki> Pàgina oficial del projecte Transphporm de generació de codi HTML a partir d'un arxiu amb marques HTML i dades dinàmiques.

<http://www.php-fig.org/> Pàgina amb una comunitat de membres que vetllen per oferir bones pràctiques i estàndards de programació en PHP.

<http://stackoverflow.com/> Comunitat de programadors en múltiples llenguatges.

<http://w3schools.com/> Web amb informació sobre l'API de diferents llenguatges (CSS, Javascript, HTML).

<http://askubuntu.com/> Portal amb una comunitat d'usuaris i desenvolupadors d'Ubuntu.

<https://developer.mozilla.org/> Base de coneixement de diverses tecnologies web.

<https://r.je/> Blog del programador Tom Butler amb diferents recursos i discussions per dur a terme bones pràctiques de programació.

<https://symfony.com/> Framework PHP.

<https://laravel.com/> Framework PHP.

<http://esadir.cat/> Portal lingüístic de la Corporació Catalana de Mitjans Audiovisuals.

<http://dlc.iec.cat/> Diccionari en línia de l'Institut d'Estudis Catalans.

<http://sinonims.iec.cat/> Diccionari en línia de sinònims de l'Institut d'Estudis Catalans.

10. Glossari

àmbit *m* Engloba les diferents àrees d'un mateix tipus. Per exemple: l'àmbit lingüístic engloba les àrees de llengua catalana, llengua catalana i llengua anglesa.

àrea *f* És el que es coneix normalment com a assignatura. Per exemple: llengua catalana.

classe *f* Cadascuna de les aules físiques on estan ubicats els alumnes. Per exemple: 1rA, 5èB...

cicle *m* Divisió d'una etapa. Per exemple, l'etapa d'educació primària es divideix en els cicles inicial, mitjà i superior.

curs *m* Any escolar. Per exemple: 2016-2017.

dimensió *f* Agrupació de les competències pròpies de cada àmbit segons la seva afinitat i el seu caràcter complementari. És cadascun dels aspectes dels que s'avalua un alumne. Per exemple: les dimensions de l'àrea de llengua catalana són la comunicació oral, la comprensió lectora, l'expressió escrita, l'hàbit lector i escriptor, i l'ús de la llengua.

especialista *m* Mestre responsable d'una assignatura. Per exemple: especialista de educació física.

etapa *f* agrupació de nivells. Per exemple: educació infantil, primària...

grup *m* També anomenada línia, és la divisió d'un nivell en un número de classes. S'utilitzen lletres (A, B, C...). Per exemple: el nivell de 5^è està dividit en 2 grups (A i B), que conformen les classes de 5^èA i 5^èB.

nivell *m* divisió d'una etapa segons l'edat dels seus alumnes. Per exemple: P4, primer, quart...

tutor *m* mestre responsable d'una classe.

11. Annexos

A part de la memòria s'han inclòs alguns documents que, per la seva extensió, no era apropiat incloure'ls en aquest document. Per veure'ls és necessari descomprimir el fitxer annexos.zip.

El sistema de directoris és el següent:

anàlisi

Diagrames de model de domini i de casos d'ús.

disseny

Diagrames de classes i de seqüència.

interfície d'usuari

Diagrama de navegació i esbossos de totes les pàgines.

models inicials d'informe

Informes que va presentar l'escola i que es van seguir inicialment com a model per dissenyar el projecte.

models finals d'informe

Informes impresos directament des de l'aplicació.

Fora dels directoris hi ha un arxiu amb el qüestionari presentat a l'equip directiu.

La presentació del projecte està disponible a <https://youtu.be/PiONrKS49vA>.