



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques

Universitat de Barcelona

ENTORN DE COMUNICACIÓ D'UNITATS D'INFORMACIÓ SOCIAL

Directors: Sergio Escalera i Marc Oliu

Departament de Matemàtica Aplicada i Anàlisi. UB

Barcelona, gener del 2015

Índex

1. Introducció i antecedents

1.1 Introductori	1
1.2 Antecedents	2

2. Definició d'objectius i motivació del problema

2.1 Objectius	3
2.2 Motivació	4

3. Metodologia i desenvolupament

3.1 Professionals	5
3.2 Casos	
3.2.1 Estats dels casos	6
3.2.2 Dinàmica dels casos	7
3.2.3 Línies de vida dels casos	8
3.2.3.1 Estàndard	9
3.2.3.2 Alternatives	10
3.3 Patró model – vista – controlador	
3.3.1 Definició	11
3.3.2 Exemple de funcionament	12
3.4 Casos d'ús	
3.4.1 Global	14
3.4.2 Per rol	15
3.4.3 Per controlador	17
3.5 Diagrama de classes	
3.5.1 Router, controladors i classes comunes	25
3.5.2 Per rol i controladors	26
3.5.3 Per rol i vistes	29
3.6 Simulació d'un cas i diagrames de seqüència	
3.6.1 Dades	32
3.6.2 Tasques	
3.6.2.1 Afegir cas	32
3.6.2.2 Acceptar el cas	36
3.6.2.3 Assignar participants i comentar en el fòrum d'un cas	38

3.6.2.4	Desenvolupar i tancar el cas	41
3.7	Seguretat i validació de dades	
3.7.1	Introducció	44
3.7.2	Sanejament i validació de les dades dels formularis	
3.7.2.1	Sanejament	45
3.7.2.2	Validació	45
3.7.2.3	Dades requerides	46
3.7.3	Encriptació Hash	47
3.7.4	Patró Post – Redirect – Get	48
3.7.5	Gestió de la informació. Visibilitat dels rols	49
3.8	Maquetació i disseny	
3.8.1	Maquetació	51
3.8.2	Disseny adaptatiu	53
3.9	Usabilitat	
3.9.1	Criteris d'usabilitat	55
3.9.2	Avaluació de la usabilitat	57
3.10	Base de dades	
3.10.1	Disseny	58
3.10.2	Taules	58
3.10.3	Querys importants	66
3.11	Plugins i llibreries externes	69
3.11.1	Tinymce	70
3.11.2	Typeahead	70
3.11.3	Jquery Upload File	71
3.11.4	Altres	71
3.12	Planificació	
3.12.1	Diagrama de Gantt	72
3.12.2	Estimació dels costos de producció	74
3.13	Eines utilitzades	75
4.	Conclusions	76
5.	Bibliografia	77
6.	Agraïments	78

1. Introducció i Antecedents

1.1 Introductori

Podria començar fent la introducció del meu treball final de grau, exposant la idea que vol transmetre, i sota la qual vaig decidir acceptar-lo i tirar-lo endavant:

“compartir informació i coneixement és positiu i ens permet fer més amb menys”

Aquesta idea, però, és tan àmplia que cal acotar-la i explicar quina és la situació sobre la qual la treballarem. És aquí on pronunciem per primer cop *Unitats d'Informació Social (UIS)*.

Les UIS són unitats reals o virtuals instrumentals dissenyades per a compartir informació en l'àmbit social al voltant de dos blocs:

1. Les Bases de dades generals: socials, legals, de guies d'actuació, sanitàries, educatives..
2. La Base de dades de casos: Història Social del ciutadà o de la família.

En l'àmbit sociosanitari, podríem definir el servei d'atenció a les persones, les seves unitats organitzatives i els seus professionals com a nodes, en un sistema en que les dades, la informació i el coneixement són claus.

Dins d'aquest àmbit, els professionals són treballadors del coneixement, en organitzacions que aprenen i canvien en un entorn en constant transformació i on les estratègies tradicionals es veuen superades en el seu intent de respondre de forma eficient a les problemàtiques socials actuals.

Comunicar aquests nodes per transmetre coneixement, ha de permetre una millora sota la qual és possible resoldre una problemàtica de manera més beneficiosa.

El meu projecte es centra en fer possible aquesta comunicació.

1.2 Antecedents

En les següents pàgines veurem quines han estat les tècniques utilitzades en l'àmbit informàtic per a aconseguir-ho. Abans però, fem una breu ullada a la resta de projectes ja existents que resolen de manera semblant aquesta problemàtica, i sota els quals, aquest ha pogut néixer amb l'objectiu de millorar els ja presents.

Els que mostrem a continuació són plataformes i actors claus que participen en el projecte.

Les Unitats d'Informació Clínica

Plataforma clínica que integra històries clíniques electròniques, bases de dades documentals, protocols, guies de pràctica clínica i altres fonts de coneixement, orientada al treball cooperatiu.

Xarxa de Salut Mental de El Prat de Llobregat

Xarxa que compta amb la participació transversal de diferents agents del territori que intervenen en el camp de la salut mental amb la finalitat de coordinar i establir una metodologia comuna de treball.

La Xarxa Socioeducativa

Xarxa de coordinació regular i estable, en l'àmbit del Districte Horta-Guinardó, entre els diferents serveis de salut, educació i atenció social adreçats a la població infantil. Revisa la qualitat assistencial, programa accions conjuntes i afavoreix l'intercanvi d'informacions i coneixements.

Xarxabaix

Xarxa que abasta tota la comarca del Baix Llobregat i que aplega els serveis dels àmbits de serveis socials, ensenyament, salut i salut mental.

2. Definició d'objectius i motivació del problema

2.1 Objectius

La problemàtica actual es basa en falta de connexió i comunicació entre les diferents serveis i organismes existents. Els diferents professionals dels àmbits sociosanitaris necessiten disposar d'una eina que solucioni aquesta mancança, i que possibiliti la col·laboració i intercanvi d'informació d'una manera lliure i centralitzada.

Quan parlem de col·laborar i participar en l'intercanvi d'informació, estem expressant el següent:

- Es vol incrementar els coneixements i competències professionals.
- Es vol augmentar el coneixement de la realitat de l'entorn.
- Es vol millorar la cooperació entre les diverses institucions.
- Es vol centrar el servei en l'usuari i prendre la millor decisió possible.

Però exactament què volem dir quan parlem de compartir informació? A quines institucions ens referim?

Veiem un exemple per a veure-ho més clar. *(A l'apartat 3.6 es mostra un exemple de com aquest tipus de casos es poden gestionar mitjançant l'eina desenvolupada).*

Imaginem que se'ns presenta un cas on l'usuari és una persona menor d'edat. Aquest nen habitualment no assisteix a les classes, i els professors sospiten que aquest fet és degut a una situació familiar "complexa".

A partir d'aquest punt, els professionals del centre escolar podrien decidir que aquest fet és prou important com per derivar aquesta informació a Serveis Socials Bàsics per tal de procedir a investigar les causes, amb la finalitat de resoldre el problema. Aquests s'encarregarien de moure els fils per recopilar totes les dades possibles sobre l'usuari i el seu entorn.

Aquestes no estan presents dins de les pròpies bases de dades on treballen, sinó que és necessària la cooperació amb altres organismes. Així per exemple, es necessitaria establir una comunicació amb el Centre de Salut Mental d'Adults per saber si algun progenitor de l'usuari presenta algun tipus de problema mental. Per la seva banda, els professionals d'aquest centre respondrien, en base a la legalitat vigent, amb la informació de la que disposen.

La interacció entre els diferents organismes, per tant, queda centralitzada en únic lloc. La facilitat de disposar de tot tipus de coneixement, permet arribar a conclusions d'una manera més ràpida i eficient. L'organisme actiu és Serveis Socials Bàsics. La resta d'organismes són únicament passius. Podríem dir que esperen a rebre una demanda per oferir la resposta.

Amb aquest exemple hem pogut comprovar que l'aplicació estarà formada per 2 tipus de rols principals: agents actius i passius. També podem començar a albirar alguna de les eines necessàries que hauran de ser presents en l'aplicació web:

- El fòrum, on es produeix la comunicació.
- La història social electrònica de l'usuari, on s'especifiquen els indicadors causants del problema.
- Un pla d'atenció, on es pugui deixar constància de l'evolució de l'usuari
- I varis més ...

No podem, doncs, començar a expressar la metodologia que s'haurà d'emprar si no tenim clars els objectius que es persegueixen. Podríem resumir-ne els principals a continuació:

- Establir i consolidar dinàmiques de col·laboració, cooperació estratègica i treball en xarxa amb el conjunt de serveis dels àmbits implicats.
- Definir i compartir models d'intervenció adients per donar resposta a les noves necessitats i realitats socials del territori.
- Disposar d'un marc de treball que permeti promoure el desenvolupament de treballar en xarxa i impliqui un sistema equitatiu d'atenció a la infància, adolescència i família.
- Realitzar accions preventives i coordinades per afavorir la promoció social i la qualitat de vida.

2.2 Motivació

La motivació del projecte, llavors, és ben clara: millorar la comunicació entre els diferents organismes sociosanitaris, aconseguint que el coneixement compartit pugui ser utilitzat per a resoldre de manera més eficient, els problemes socials que afecten a bona part de la societat actual.

El beneficiari directe, és l'usuari del cas. Però també fora bo veure i entendre, que d'aquí sorgeixen una sèrie de beneficiaris indirectes: els propis treballadors de l'àmbit sociosanitari i la resta de la societat.

Si disposem d'un conjunt de dades centralitzades i puntuades, podem, a posteriori, fer una explotació d'aquestes per a intentar trobar patrons de conductes (causes) que propicien un cert tipus de problemàtica social.

Aquest "*Big Data*" jo el veig com un conjunt de dades que poden ser utilitzades en un àmbit no comercial, perseguint l'objectiu de trobar mecanismes minimitzadors de conflictitats socials.

Aquest projecte deixa oberta una porta gran a futurs estudis estadístics.

3. Metodologia i desenvolupament

Els casos seran l'eina centralitzadora d'informació dels diferents organismes, de l'usuari, i de l'evolució d'aquest. En funció de condicionants i del propi progrés del cas, aquest tindrà un estat diferent.

3.1 Rols dels professionals

Dins de l'aplicació coexisteixen un conjunt de rols. Cada professional està associat exclusivament a un sol rol i pot realitzar unes funcionalitats concretes no necessàriament excloents a la resta de rols. Anem a veure'ls tots.

Professional de Serveis Socials Bàsics

Tots aquells professionals membres de SSB que actuen com a agents actius dins de la resolució del cas. Tenen una sèrie de privilegis tals com poder decidir a quin organisme volen assignar al cas, editar dades de l'usuari, afegir informació, manegar els estats del cas. Tenen un alt poder de decisió. El cas funciona si el professional de SSB funciona.

Professional d'organisme associat

Són tots aquells professionals no administradors que pertanyen a tots els organismes excepte a SSB. Actuen com a agents passius responen comentaris adreçats a ells. També tenen menys privilegis tals com no poder editar informació sensible i una visualització parcial de la comunicació. L'única interacció que veuen és la seva amb SSB. Normalment seran qui crearan i derivaran els casos.

Assessor

S'encarrega de resoldre els dubtes que els dos rols anteriors puguin tenir durant el desenvolupament d'un cas.

Administrador de Serveis Socials Bàsics

És qui té el màxim poder dins dels casos. És capaç d'assignar o desassignar membres de Serveis Socials Bàsics. També pot desestimar casos, acceptar-ne... però no tancar-los. La visualització de dades és limitada. Per exemple, sap els organismes que són presents en un cas però no sap els noms reals dels participants.

També disposa d'altres funcionalitats com veure estadístiques, exportar dades de les valoracions dels casos, afegir situacions...

Administrador d'enquestes

En finalitzar un cas, s'envien enquestes per a obtenir feedback d'aquest. És qui les rep i qui s'encarrega de revisar-ne els resultats.

Administrador de recursos

En funció dels casos i de l'organisme al que es pertany, cada professional (i en funció de les situacions d'un cas) podrà tenir assignats una sèrie de recursos en xarxa per a ser consultats en tot moment. Aquests recursos són afegits per aquest administrador.

Administrador de professionals

És qui s'encarrega de crear nous professionals de qualsevol tipus de rol.

3.2 Casos

Anem a veure primer de tot què són els casos i quina serà la seva línia de vida. Volem intentar mostrar la metodologia adequada que s'haurà de realitzar.

Com hem introduït anteriorment, els casos seran l'eina centralitzadora d'informació dels diferents organismes, de l'usuari, i de l'evolució d'aquest. En funció de condicionants i del propi progrés del cas, aquest tindrà un estat diferent.

3.2.1 Estats dels casos

Pendent: Serà un cas pendent aquell que ha estat creat i derivat per un organisme que no és Serveis Socials Bàsics i que es troba en una fase prèvia a l'acceptació.

Desestimat: Serà l'estat conseqüent de no ser acceptat per un membre de SSB. S'haurà d'informar dels motius de no acceptació.

Obert: Aquest estat es pot donar en tres situacions. Quan un membre de SSB accepta un cas que se li havia derivat a ell, quan un membre de SSB crea per ell mateix un cas des de zero, o bé quan un cas torna a ser obert després d'estar un temps tancat.

Tancat: Quan el membre de SSB assignat al cas ho consideri oportú, el cas pot ser tancat. Els motius de tancament són propis i dependents dels resultats

obtinguts. Abans de tancar el cas es procedirà a omplir un qüestionari per tal de puntuar els diferents organismes participants en el cas.

Com es pot comprovar, l'estat i vida del cas no és estàtic, sinó que és dinàmic. Un cas pot ser reobert tantes vegades com es vulgui. De la mateixa manera, un cas desestimat pot ser obert si es rep nova informació considerada rellevant. Les funcionalitats que poden ser aplicades a cada cas, variaran en funció d'aquests estats.

3.2.2 Dinàmica dels casos

Passem ara a explicar els casos i la metodologia apropiada. Primer haurem d'entendre els mecanismes, saber com funcionen i quina és la forma en la qual han estat pensats per tal de ser utilitzats de forma oportuna. No ens referim, però, a mostrar de manera exacta els passos concrets a l'hora de realitzar una tasca. Aquesta informació existeix, i pot ser consultada en la pròpia web així com en l'annex d'aquest propi document.

Com dèiem, un cas està conformat per molta informació. En un inici qui crea el cas introdueix els indicadors/problematiques, escriu les dades de l'usuari i del representant (si en té) i selecciona les situacions.

Cada membre de Serveis Socials Bàsics també té varies situacions assignades. Les situacions serveixen per definir un cas i derivar-lo al professional correcte. La persona que crea el cas és l'encarregada de seleccionar-les, encara que posteriorment poden ser editades per l'Administrador de Serveis Socials Bàsics. Les situacions defineixen clarament les problemàtiques del cas: És un cas sobre **assetjament escolar**? Doncs s'hauran de seleccionar les etiquetes *assetjament escolar* i *violència escolar* per exemple.

Un cop el cas és correctament derivat per un professional d'organisme associat, el membre de SSB a qui ha sigut derivat haurà de decidir si l'accepta o no. En cas afirmatiu el cas s'obre i es comença el procediment habitual per a resoldre'l.

Els passos següents que haurà de fer el membre de SSB és anar recopilant informació i afegint-ne de nova on sigui necessària. Així per exemple pot afegir familiars a l'arbre sociogenealògic, afegir esdeveniments en el timeline, pot assignar nous participants, etc..

La metodologia de resolució del cas és pròpia de cada professional i sobrepassa la capacitat d'explicació que podria donar-se en aquest document. Sí podem expressar, que totes les eines han estat pensades per ser utilitzades, no de manera obligatòria, sinó com a ajut per a facilitar la comprensió de les dades.

La meua memòria es centrarà a partir d'ara en explicar la metodologia i desenvolupament de la pàgina web en l'àmbit de la informàtica, que és precisament l'àmbit que com a estudiant d'aquesta disciplina em pertoca.

3.2.3 Línia de vida dels casos

3.2.3.1 Línia de vida principal

Suposem que es donen les condicions òptimes per a l'obertura, desenvolupament i tancament d'un cas. Els passos o etapes que es produirien són els següents:

- **Obertura:** Durant aquesta etapa, un membre que no és de SSB detecta que un usuari té una problemàtica. Crea un cas nou informant d'això i el deriva. El cas queda en un estat de pausa. El membre de SSB que ha quedat assignat al cas, revisa les dades i indicadors introduïts, i decideix acceptar-lo per intentar resoldre'l.
- **Desenvolupament:** El membre de SSB intenta resoldre el cas recopilant dades de diversos organismes. Es produeix un intercanvi d'informació i s'interactua amb les diferents opcions presents al cas. El *timeline* és completat, els recursos consultats, l'arbre sociogenealògic complementat...
Aquesta etapa és cíclica i no acaba fins que el membre de SSB ho decideix.
- **Tancament:** El professional de SSB escriu els motius de tancament del cas i puntua tots els organismes participants.

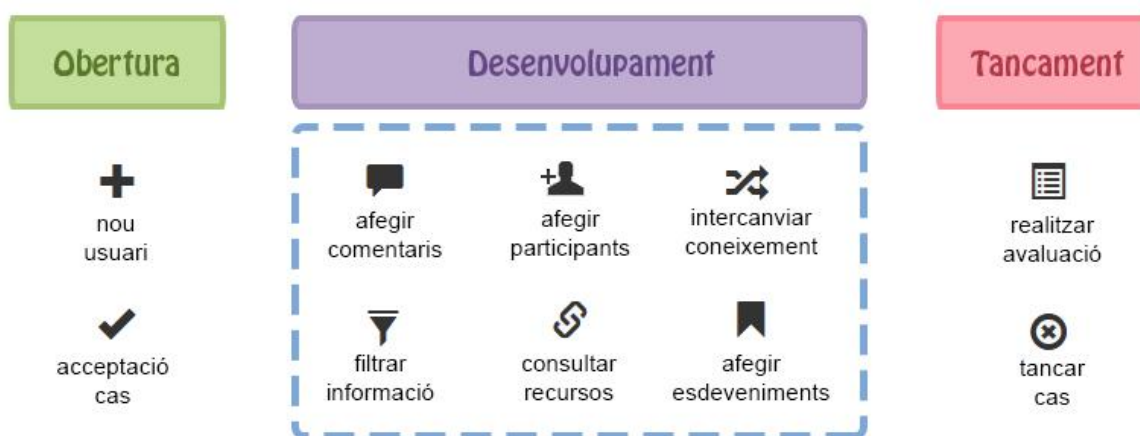


Figura 1. Mostra les etapes de la vida d'un cas estàndard

3.2.3.2 Línies de vida alternatives

Cas creat per SSB

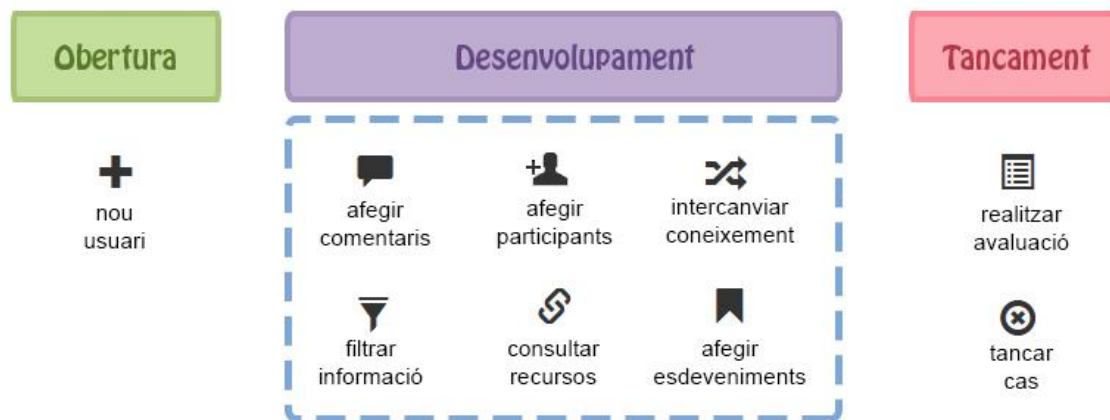


Figura 2. Mostra les etapes de la vida creada per un membre de SSB

En aquesta variació, el mateix membre de SSB que crea el cas és qui queda assignat. Ja no és necessària la revisió i acceptació del cas. La resta d'etapes són les mateixes que en l'estàndard.

Desestimació del cas

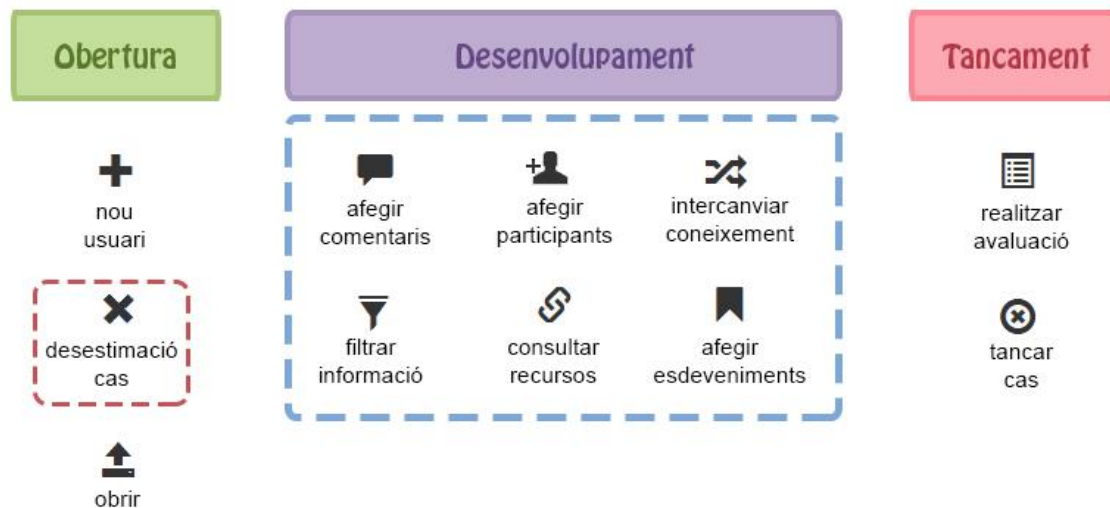


Figura 3. Mostra les etapes de la vida d'un cas reobert

Si es considera que la informació és incompleta, no rellevant o no prou important com per tirar-lo endavant, el cas es desestima i queda en un estat de congelació. Fins que no es reverteix la situació i s'obre el cas, no comença el seu desenvolupament.

Reobertura del cas



Figura 4. Mostra les etapes de la vida d'un cas desestimat

Un cas que prèviament hagi estat tancat pot tornar a ser obert. Aquesta situació pot ser donada quan es presenta un nou símptoma de problemàtica social, en un usuari que prèviament havia format part de la base de dades de casos.

3.3 Patró Model Vista Controlador

3.3.1 Definició

Per a desenvolupar el software he utilitzat el patró d'arquitectura MVC (Model – Vista – Controlador).

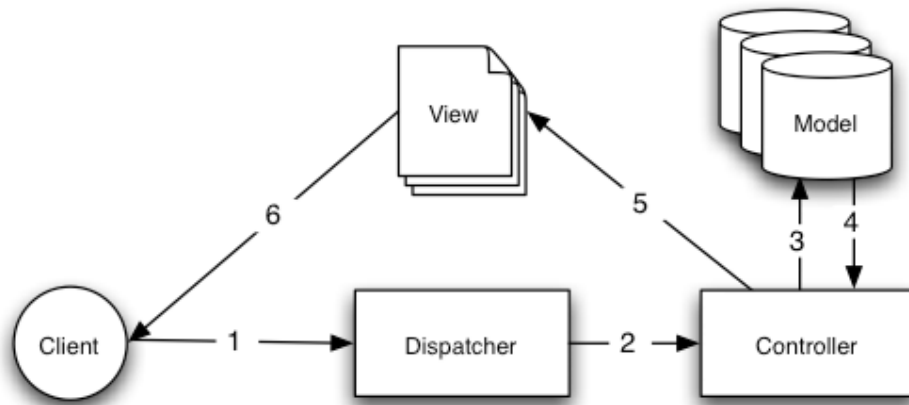


Figura 5. Mostra la separació de capes utilitzades en el patró MVC

Aquest patró divideix l'aplicació en tres parts: el model de dades (*model*), la interfície d'usuari (*view*) i la lògica de control (*controller*).

Separar les funcions de l'aplicació en diverses capes fa que aquesta sigui molt lleugera, fàcilment adaptable i que pugui ser desenvolupada ràpidament de forma modular i mantenible. Els canvis efectuats en una capa de l'aplicació no repercuteixen en la resta.

Anem a veure el funcionament bàsic del patró per intentar comprendre'l millor:

1. Hi ha un usuari que realitza una petició
2. El *dispatcher* (o *router*) comprova la URL i li passa la petició al controlador adequat.
3. El controlador fa les crides als models corresponents que són qui interactuen amb la base de dades.
4. El model contesta al controlador enviant les dades demanades.
5. El controlador envia aquestes dades a la vista
6. La vista mostra a la informació cridant *templates*.

3.3.2 Exemple de funcionament

Anem a veure de forma més detallada cada un dels passos anteriors tot mostrant un exemple (*suposarem que tot funciona de manera correcta, més endavant expliquem la gestió d'errors*).

Tota petició passa per el [index.php](#).

Ell s'encarrega de cridar al *dispatcher*. El *dispatcher*, qui a partir d'ara anomenarem *router*, rep la següent *url*

www.uis.cat/participants/obtenir-participant/17

Ara serà necessari explotar aquesta *url*. En la meva aplicació el *router* realitza aquestes 4 accions:

1. Obtenir la url:
Es crida a una classe externa que la saneja i l'obté. Dividim la *url* en trossos tallant per cada *slash "/"* guardant cada part en una llista.
2. Obtenir el rol de l'usuari:
A partir de les variables de sessió que han sigut emmagatzemades, després de fer el *login* correctament.
3. Carregar el controlador adequat:
En funció del rol de l'usuari, es cridarà el fitxer *ParticipantsController* present al directori especificat pel rol. Això és molt ràpid de dir, però realment no estem expressant els motius de perquè succeeix això. El que fem és construir una ruta a partir de les dades que disposem i de diverses variables estàtiques. En la meva aplicació, tot controlador porta el sufix *Controller* i va precedit pel nom especificat per la *url*. L'ordre d'elements presents a la *url* també és decisió nostra, jo des d'un inici he seguit la següent configuració.
controlador / nom-de-la-funció / paràmetre 1 / ... / paràmetre N
4. Crida al mètode especificat passant-li els paràmetres
Com que ja tenim el controlador carregat correctament, només ens faltaria fer la crida al mètode del controlador, accedint a la posició número 2 de la llista i passant tants paràmetres com hi hagi a la *url*.

El procés de *routing* hauria finalitzat correctament i ara ens trobaríem dins del mètode *afegirParticipant* de la classe *ParticipantsController*.

El següent pas correspondria a fer les crides a la base de dades. Ho farem a partir de diversos models. Abans de tot, anem a veure com he entès els models i els motius que m'han portat a realitzar-los d'aquesta manera.

Habitualment, quan cercava informació sobre la creació de models, i en no ser un patró estrictament definit, trobava disparitat d'informació. Mentre que una part de la docència informàtica entenia els models com a entitats relacionals pures de la base de dades (és a dir, les taules definint les columnes com a variables), d'altres l'entenien com una capa de comunicació amb les taules però no constituïda com una entitat pura. Finalment vaig decidir seguir aquesta segona opció ja que va ser la que més apropiada vaig trobar i la que millors resultats em proporcionava.

Els models, per tant, no són representacions estrictes de les taules de la BD, sinó que contenen un conjunt de mètodes per a fer consultes, insercions, actualitzacions.. sobre una o varies taules estretament relacionades i agrupades dins del mateix model. Cal dir que els models són independents dels rols, tot i que els seus mètodes seran cridats en funció dels privilegis que atorgui el propi rol.

Els models, a més, treballen de manera independent respecte a la base de dades. En iniciar-se l'aplicació, s'obre una connexió a ella i s'emmagatzema com a global per a que pugui ser utilitzada per qualsevol model. La classe externa que farà les crides a la BD pot ser tractada com un element diferent i fàcilment modificable. Si per exemple volguéssim migrar de sistema gestor de bases de dades i el codi no fos modulable, hauríem de canviar cada *query* de l'aplicació, dificultant moltíssim la mantenibilitat del codi. Ara, en canvi, caldria canviar una línia de codi especificant quin és el gestor a utilitzar.

Seguint l'exemple anterior, estaríem dins del mètode del controlador demanat, i carregariem un model (*ParticipantsModel*), on faríem una crida al mètode *obtenirParticipant*.

Dins d'aquest mètode es cridaria a la classe auxiliar d'interacció amb la BD passant-li la query i els paràmetres adients. El resultat obtingut seria retornat al controlador.

Ara el controlador disposa del participant especificat, es tracta d'una *array* creada en el model però no de cap classe (recordem que no fèiem servir entitats pures). Aquestes dades conjuntament amb d'altres serien enviades a la classe de la vista corresponent, que a l'inici del controlador havia estat prèviament carregada.

En concret es cridaria a un mètode d'aquesta vista, com per exemple *mostrarParticipant*. Finalment, i a base de diversos *require*, s'anirien carregant diversos *templates* per a formatar correctament les diverses parts que conformaran la interfície de la web. Els *templates* són re aprofitables la majoria de cops.

Cal comentar, que no s'ha utilitzat cap *framework* de *php* tot i que n'hi ha un munt de gratuïts. Els motius són principalment dos: Volia aprendre des de 0 com funciona el patró MVC i les dificultats que suposa la intercomunicació entre capes. També va ser un factor clau el fet de que aprendre a utilitzar un *framework* requereix una corba d'aprenentatge normalment alta. Considero que abans de fer servir un *framework* s'ha de saber com funciona la idea que proposa.

3.4 Casos d'ús

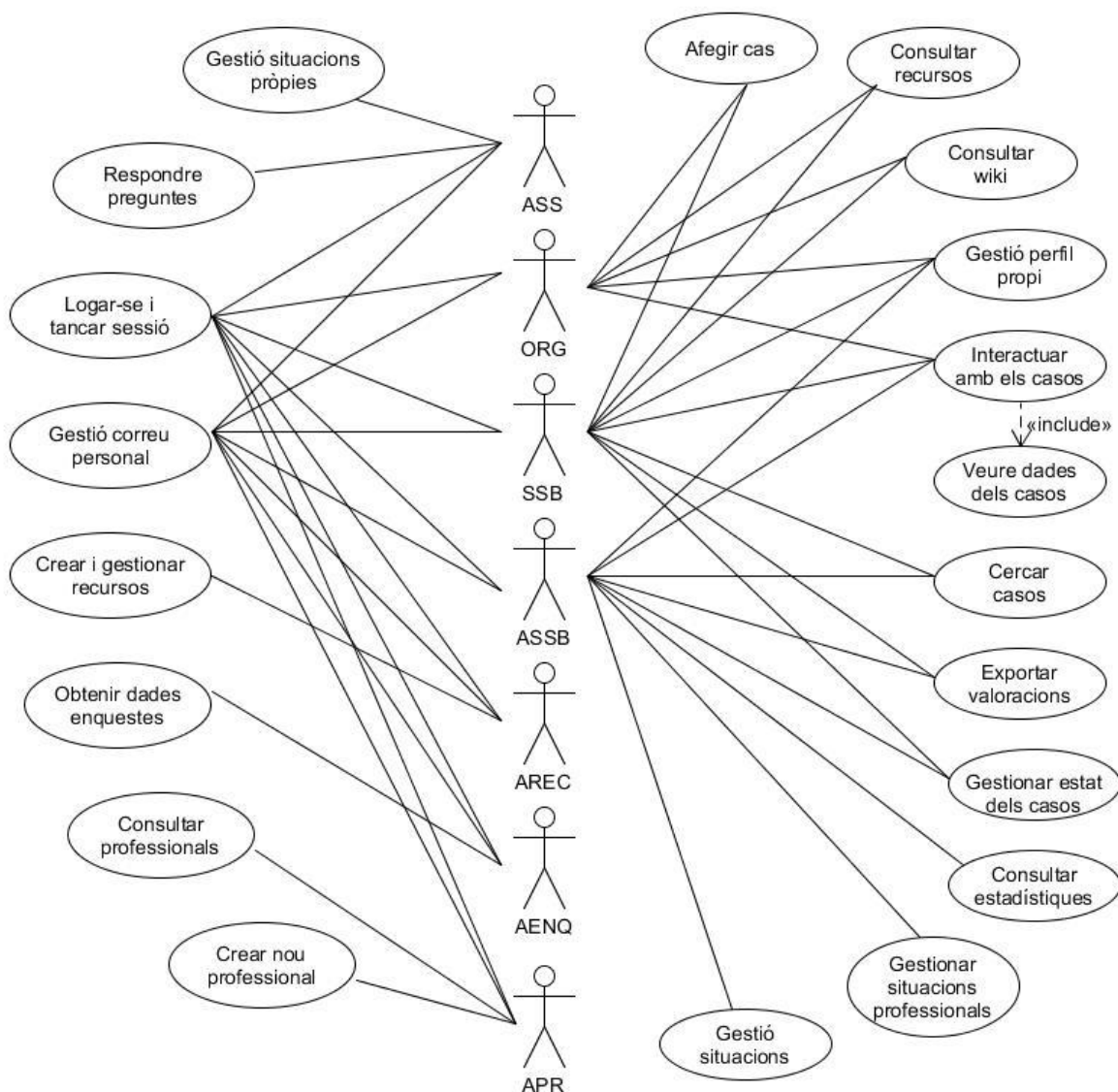
Anem a mostrar tots els diferents escenaris de casos d'ús presents en l'aplicació

Llegenda:

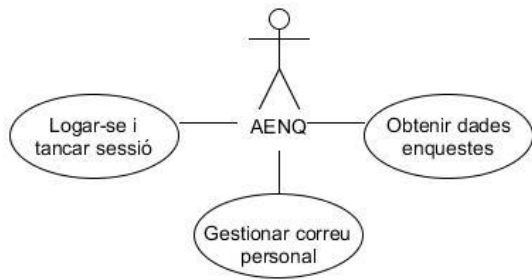
SSB	Professional de Serveis Socials Bàsics
ORG	Professional d'Organisme Associat
ASSB	Administrador de Serveis Socials Bàsics
AENQ	Administrador d'enquestes
AREC	Administrador de recursos
APR	Administrador de professionals
ASS	Assessor
NLG	Professional no loguejat

3.4.1 Global

En l'apartat general podrem obtenir una visió global que serà la precursora de la visió detallada dels controladors.



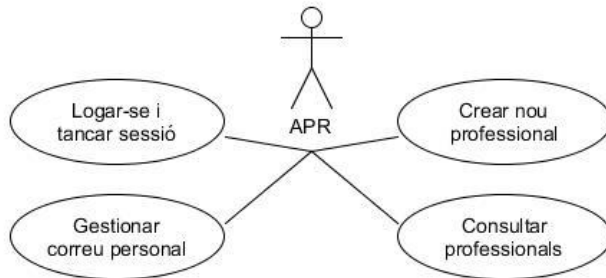
3.4.2 Per rol



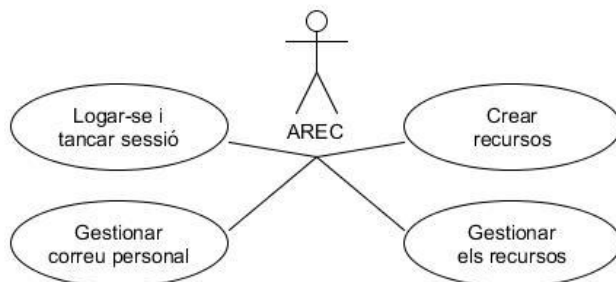
LoginController: Logar-se i tancar sessió.

CorreuController: Gestionar correu personal i obtenir dades enquestes.

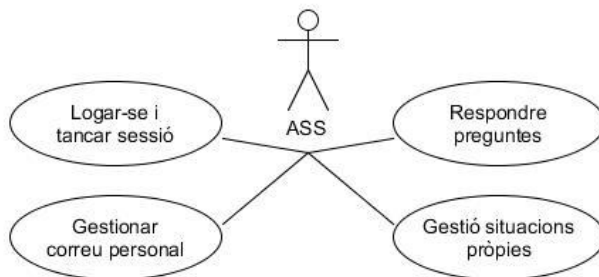
(Com que logar-se i tancar sessió i gestionar correu personal son activitats presents en tots els rols, no es nombraran més.)



ProfessionalsController: Crear nou professional i consultar-ne..



RecursosController: Crear recursos i gestionar-los.



AssessoramentController: Respondre preguntes.

SituacionsController: Gestió situacions pròpies.



ProfessionalsController: Gestió perfil propi, gestionar situacions professionals.

SituacionsController: Gestió situacions.

EstadistiquesController: Consultar estadístiques.

IndexController: Cercar casos.

ValoracionsController: Exportar valoracions.

--

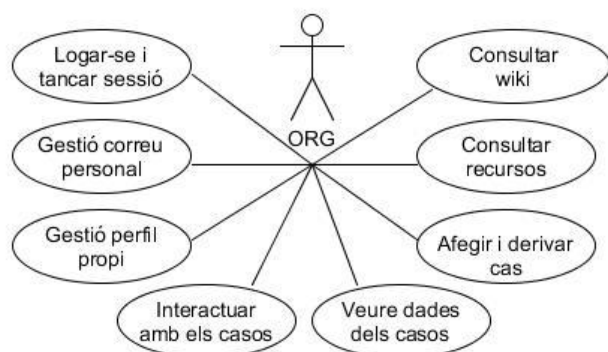
Interactuar amb els casos

Afegir participants de SSB

(*ParticipantsController*).

Gestionar estats dels casos: desestimar, reobrir, acceptar (*CasController*).

Actualitzar situacions (*SituacionsController*).



Veure dades dels casos

Arbre sociogenealògic (*ArbreController*), fòrum de treball (*ForumController*), història social electrònica (*HseController*), pla d'atenció (*TimelineController*), dades usuari (*UsuariController*), valoracions (*ValoracionsController*), estadístiques (*EstadistiquesController*).

ProfessionalsController: Gestió perfil propi.

CasController: Afegir i derivar cas.

RecursosController: Consultar recursos.

WikiController: Consultar wiki.

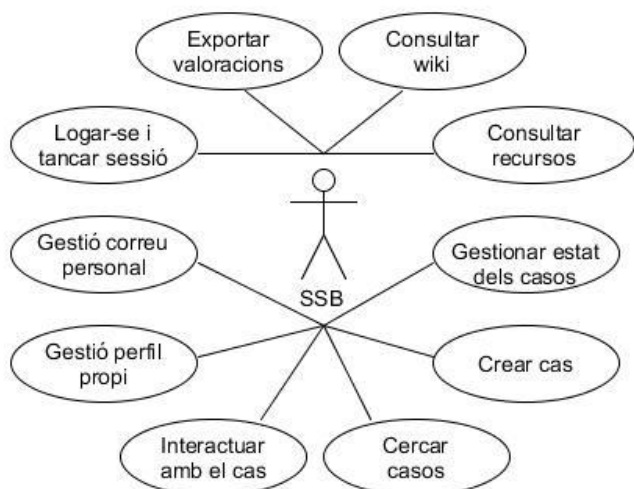
--

Interactuar amb els casos

Comentar en el fòrum de treball (*ForumController*), preguntar en l'espai d'assessorament (*AssessoramentController*).

Veure dades dels casos

Arbre sociogenealògic (*ArbreController*) història social electrònica (*HseController*), pla d'atenció (*TimelineController*), dades usuari (*UsuariController*).



ProfessionalsController: Gestió perfil propi.

CasController: Crear cas.

IndexController: Cercar casos.

RecursosController: Consultar recursos.

WikiController: Consultar wiki.

ValoracionsController: Exportar valoracions.

--

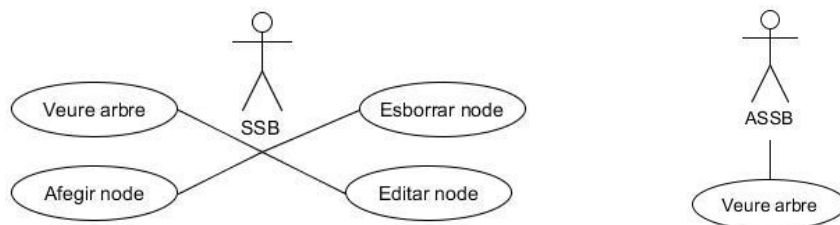
Interactuar amb els casos

Comentar en el fòrum de treball (*ForumController*), preguntar en l'espai d'assessorament (*AssessoramentController*), interactuar amb l'arbre sociogenealògic (*ArbreController*), actualitzar la història social electrònica (*HseController*), gestionar el pla d'atenció (*TimelineController*), actualitzar les dades usuari (*UsuariController*), afegir i consultar valoracions (*ValoracionsController*), gestionar participants (*ParticipantsController*). Gestionar estats dels casos (acceptar, desestimar, tancar i reobrir).

3.4.3 Per controlador

Arbre sociogenealògic

Conté les funcionalitats relacionades amb l'arbre sociogenealògic d'un cas.



Veure arbre: SSB i ASSB. Observació de l'arbre sociogenealògic.

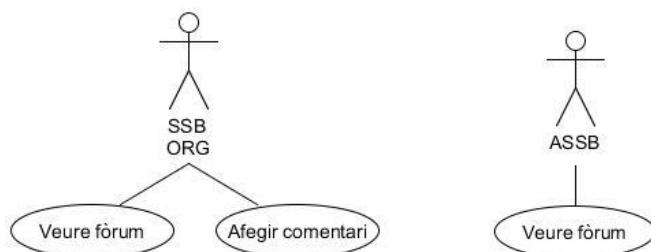
Afegir node: SSB. Afegir un nou familiar, amic.. a l'arbre sociogenealògic.

Editar node: SSB. Editar les dades d'un dels familiars o amics ja existents dins l'arbre.

Esborrar node: SSB. Esborrar un familiar o amic ja existent dins de l'arbre.

Fòrum de treball

Conté les funcionalitats relacionades amb els fòrums interns d'un cas.

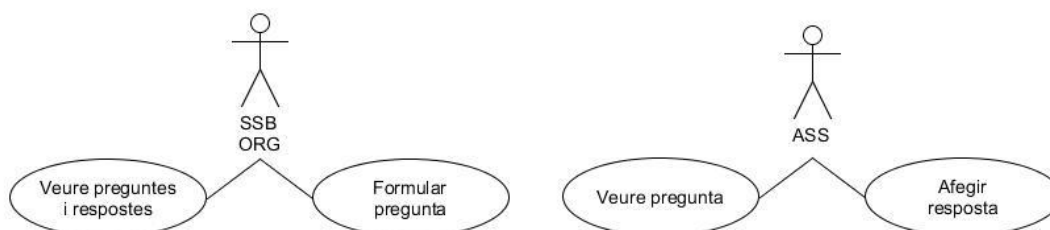


Veure fòrum: SSB, ORG i ASSB. Visualització dels comentaris. ORG només interactua amb SSB, per tant només veu la conversa que té amb ell. SSB pot interactuar amb tots els ORG que consideri necessari. Els dos només veuen aquells comentaris amb menció directa sobre ells. ASSB veu tota la conversa però no els noms reals dels membres participants.

Afegir comentari: SSB i ORG. Afegir un nou comentari al fòrum del cas.

Espai d'assessorament

Conté les funcionalitats relacionades amb l'espai d'assessorament d'un cas.



Veure preguntes i respostes: SSB i ORG. Observar el conjunt de preguntes i respostes fetes per cada professional dins de l'espai d'assessorament.

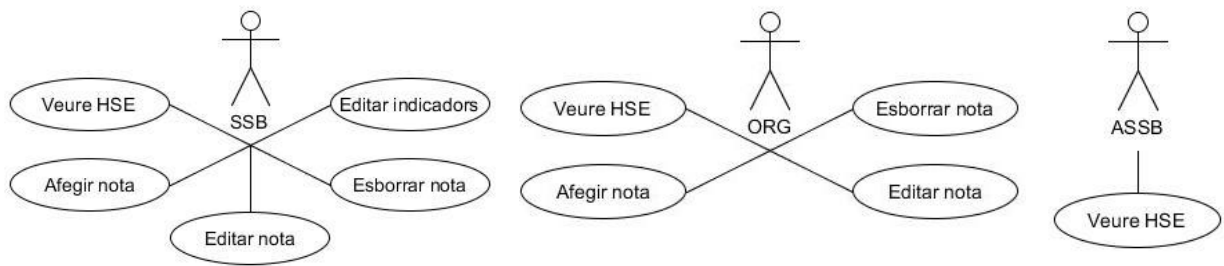
Formular pregunta: SSB i ORG. Realitzar una pregunta a un Assessor.

Veure pregunta: ASS. Veure la pregunta (pot estar contestada o no).

Afegir resposta: ASS. Respondre una pregunta que encara no ha estat resolta.

Història Social Electrònica

Conté les funcionalitats relacionades amb la Història Social Electrònica d'un cas, així com amb les notes personals de seguiment.



Veure HSE: SSB, ORG i ASSB. Observació de l'història social electrònica de l'usuari. En el cas de SSB i ORG també poden veure les seves notes personals.

Afegir nota: SSB i ORG. Afegir una nova nota personal relacionada amb el cas.

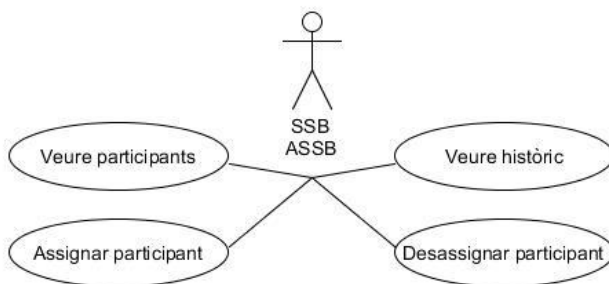
Editar nota: SSB i ORG. Editar una de les notes personals ja existents.

Esborrar nota: SSB i ORG. Esborrar una de les notes personals ja existents.

Editar indicadors: SSB. Editar els indicadors del cas (afegir-ne de nous, complementar-ne d'existents..)

Participants

Conté les funcionalitats relacionades amb els participants d'un cas.



Veure participants: SSB i ASSB. Consultar els participants actuals assignats al cas. ASSB únicament veu els noms dels organismes en el cas que no siguin SSB. SSB, en canvi, veu el nom dels professionals i l'organisme al que pertanyen.

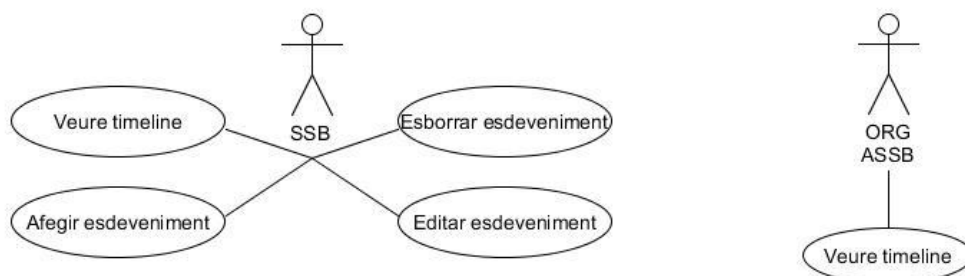
Assignar participant: SSB i ASSB. Assignar un nou participant al cas. SSB pot assignar professionals de qualsevol organisme sempre que aquell organisme no sigui present dins del cas. ASSB únicament pot afegir membres de SSB al cas (poden ser varis).

Desassignar participant: SSB i ASSB. Desconnectar un professional d'un cas on prèviament hi era.

Veure històric: Veure l'històric d'assignacions i desassignacions de professionals al cas.

Timeline

Conté les funcionalitats relacionades amb el *timeline* d'un cas. Es mostren de manera cronològica tots aquells esdeveniments que es consideren importants.



Veure *timeline*: SSB, ORG i ASSB. Consultar els esdeveniments considerats importants relacionats amb un cas en concret.

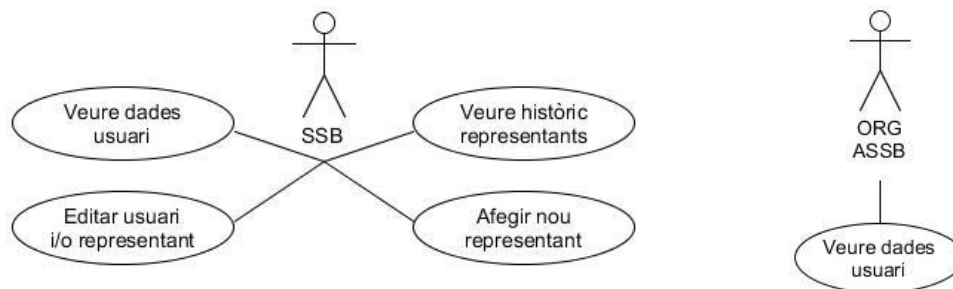
Afegir esdeveniment: SSB. Afegir un nou tipus d'esdeveniment al *timeline*.

Editar esdeveniment: SSB. Editar un esdeveniment ja existent al *timeline*.

Esborrar esdeveniment: SSB. Esborrar un esdeveniment ja existent al *timeline*.

Usuari

Conté les funcionalitats relacionades amb l'usuari del cas



Veure dades usuari: SSB, ORG i ASSB. Consultar les diferents dades personals d'un usuari. SSB veu totes les dades. ASSB i ORG només veuen nom i cognoms i NIF.

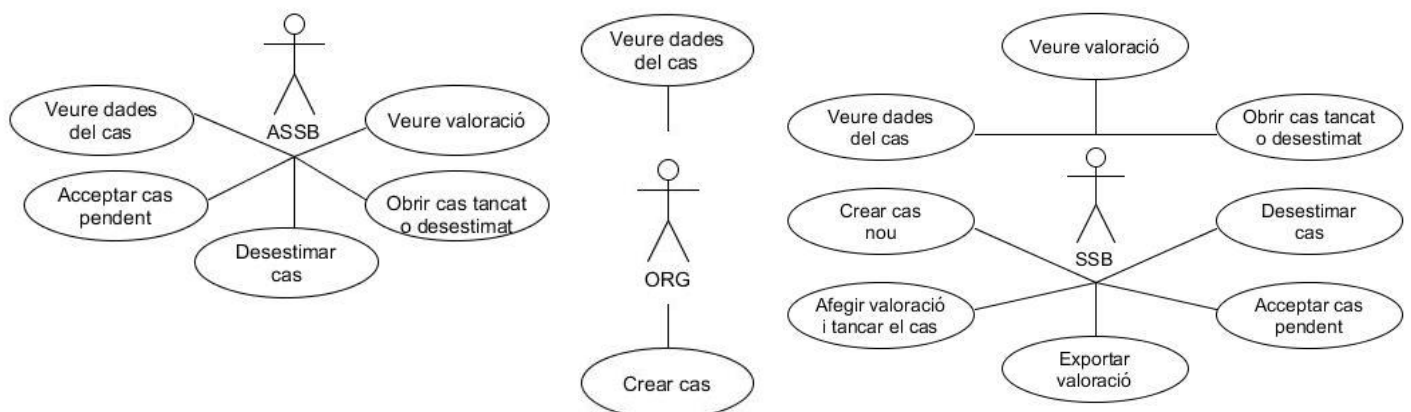
Editar usuari i/o representant: SSB. Poder editar totes les dades personals de l'usuari excepte el NIF. També pot editar les dades del representant actual.

Afegir representant: SSB. Afegir un nou representant de l'usuari. Automàticament, el representant anterior (si n'hi havia) passa a l'històric de representants.

Veure històric representants: SSB. Consultar tots els representants que ha tingut aquell usuari al llarg del cas.

Cas

Conté les funcionalitats relacionades amb fets concrets del cas.



Veure dades del cas: SSB, ORG i ASSB. Consultar l'identificador del cas, l'estat actual i l'històric d'estats, les situacions, etc..

Crear cas: SSB i ORG. Afegir un cas nou. Seguir els passos indicats. Si SSB crea cas es deriva automàticament a ell. Si ORG crea un cas, es deriva segons les situacions escollides.

Afegir valoració i tancar cas: SSB. Escollir els motius de tancament del cas i realitzar la puntuació dels organismes existents. Finalment es produeix el tancament del cas.

Exportar valoració: SSB. Exportar en csv les puntuacions dels organismes participants del cas.

Acceptar cas pendent: SSB i ASSB. Permetre que el cas pugui ser iniciat i s'habilitin totes les opcions de gestió i manipulació internes.

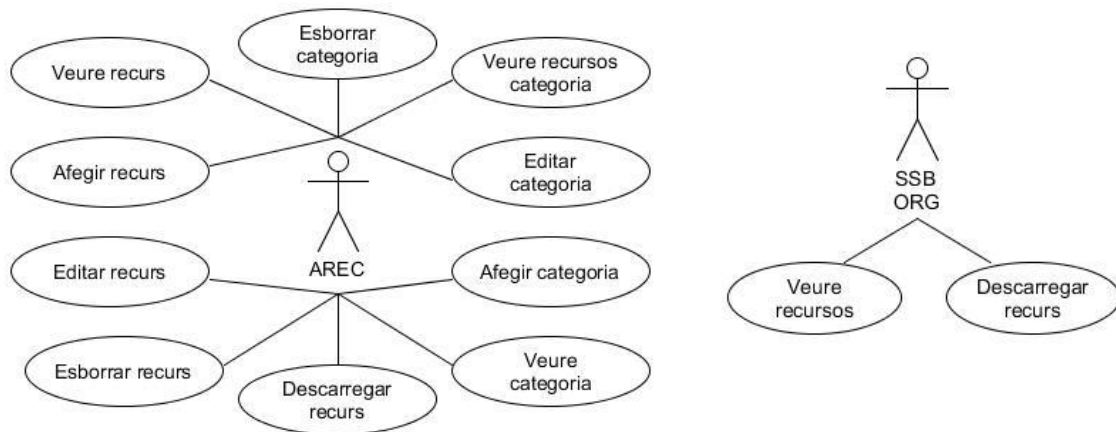
Desestimar cas: SSB i ASSB. No acceptar la derivació del nou cas creat. S'hauran d'escriure els motius de desestimació.

Obrir cas tancat o desestimat: SSB i ASSB. Permetre la reobertura de casos tancats, o bé iniciar la obertura de casos desestimats.

Veure valoració: Consultar totes les valoracions d'un cas a l'hora del seu tancament.

RecursosController

Conté les funcionalitats relacionades amb la pàgina inicial de l'aplicació.



Veure recursos: SSB i ORG. Consultar els recursos assignats a aquell cas. La visualització de segons quins recursos depèn de les situacions del cas i de l'organisme del propi professional. AREC veu tot el llistat de recursos creats.

Descarregar recurs: SSB, ORG i AREC. Descarregar el recurs si es tracta d'un tipus de recurs de fitxer.

Veure recurs: AREC. Veure totes les dades d'un recurs prèviament creat.

Afegir recurs: AREC. Afegir un nou recurs del tipus fitxer o enllaç.

Editar recurs: AREC. Editar les dades d'un recurs ja existent. Si es tracta d'un recurs de tipus arxiu no es pot editar el fitxer. Si es tracta d'un enllaç si que es pot editar la *url*.

Esborrar recurs: AREC. Esborrar un recurs ja existent.

Veure categoria: AREC. Consultar el nom i data d'actualització d'una categoria ja existent.

Afegir categoria: AREC. Crear una nova categoria.

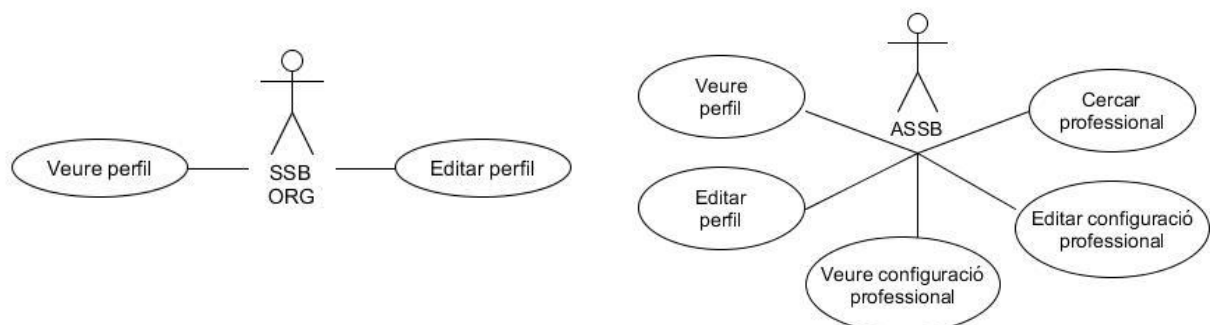
Editar categoria: AREC. Editar el nom d'una categoria ja existent.

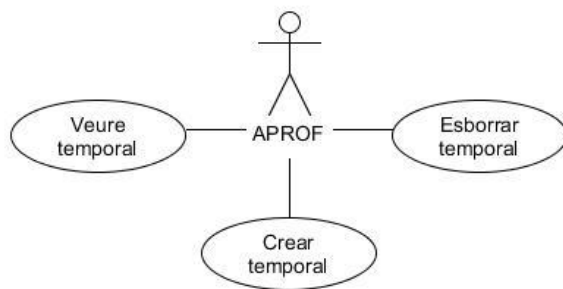
Esborrar categoria: AREC. Esborrar una categoria ja existent.

Veure recursos categoria: AREC. Veure el llistat de tots els recursos presents dins d'una única categoria.

Professionals

Conté les funcionalitats relacionades amb els professionals de l'aplicació.





Veure perfil: SSB, ORG i ASSB. Consultar les dades del teu propi perfil, o bé les dades d'algun altre professional. Alguns camps, com el del correu personal resten ocults.

Editar perfil: SSB, ORG i ASSB. Editar les teves pròpies dades i poden actualitzar la contrasenya.

Veure configuració professional: ASSB. Observar dades d'un professional de SSB tals com l'organisme al que pertany i les situacions que té assignades.

Editar configuració professional: ASSB. Editar les situacions assignades que té un professional de SSB.

Cercar professional: ASSB. Cercar professional per noms, cognoms o situacions.

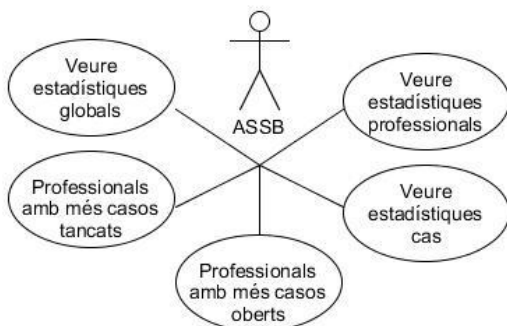
Veure temporal: APROF. Veure les dades d'un professional temporal que encara no ha activat el seu compte.

Crear temporal: APROF. Crear un nou professional de qualsevol tipus de rol.

Esborrar temporal: APROF. Esborrar un professional temporal que encara no ha activat el seu compte.

Estadístiques

Conté les funcionalitats relacionades la visualització de tot tipus d'estadístiques.



Veure estadístiques globals: ASSB. Veure un conjunt d'estadístiques globals tals com el número de casos oberts i tancats, el número de professionals existents i l'organisme al que pertanyen..

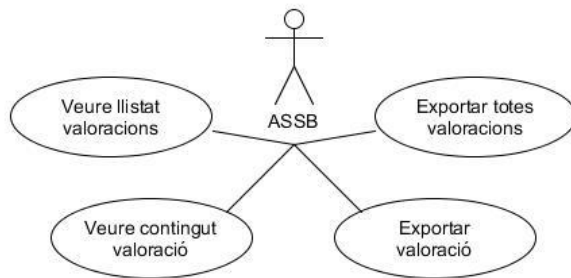
Professionals amb més casos tancats: ASSB. Veure el llistat de professionals que han tancat més casos i encara estan assignats a ells.

Professionals amb més casos oberts: ASSB. Veure el llistat de professionals que han obert més casos i encara estan assignats a ells.

Veure estadístiques cas: ASSB. Veure el conjunt d'estadístiques relacionades amb un cas tals com el número de dies que porta actiu, el número de comentaris rebuts per organisme, etc..

Veure estadístiques professional: ASSB. Veure el conjunt d'estadístiques relacionades amb un professional tals com el número de comentaris escrits, els casos en que ha participat, etc..

Valoracions



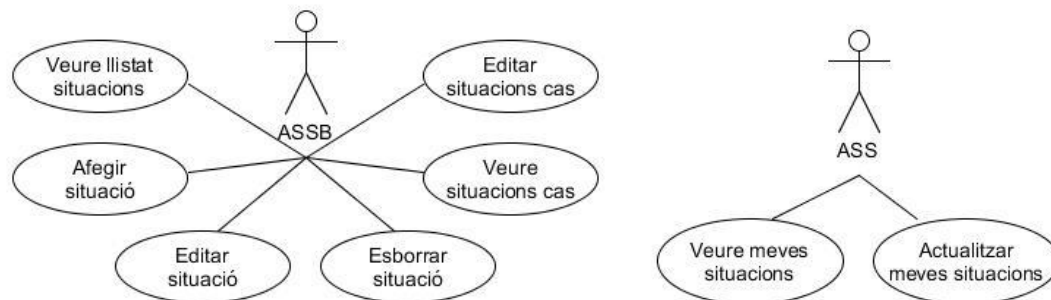
Veure llistat valoracions: ASSB. Veure totes les valoracions realitzades pels membres de Serveis Socials Bàsics.

Veure contingut valoració: ASSB. Veure els motius de tancament del cas i les puntuacions dels organismes d'una valoració en concret.

Exportar valoració: ASSB. Exportar en csv les puntuacions dels organismes present en la valoració d'un cas.

Exportar totes valoracions: ASSB. Exportar en csv les puntuacions dels organismes de totes les valoracions realitzades.

Situacions



Veure llistat situacions: ASSB. Veure totes les situacions actuals presents en la base de dades de l'aplicació.

Afegir situació: ASSB. Afegir una nova situació.

Editar situació: ASSB. Editar una situació ja existent.

Esborrar situació: ASSB. Esborrar una situació ja existent.

Veure situacions cas: ASSB. Consultar totes les situacions que hi ha assignades a un cas.

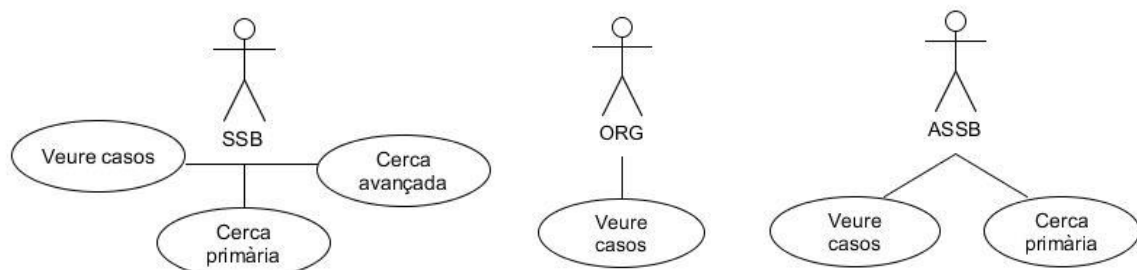
Editar situacions cas: ASSB. Assignar noves situacions a un cas o des assignar-ne d'existents.

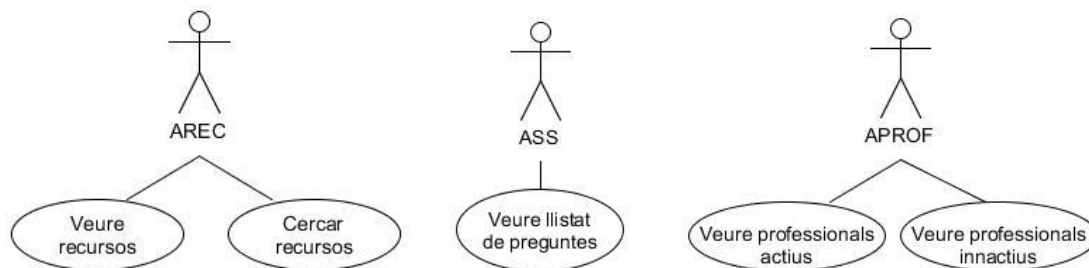
Veure meves situacions: ASS. Consultar totes les situacions que el propi professional té assignades.

Actualitzar meves situacions: ASS. Actualitzar les pròpies situacions.

Index

Conté les funcionalitats relacionades amb la pàgina inicial de l'aplicació.





Veure casos: SSB, ORG i ASSB. En el cas de SSB i ORG, veuen els casos que tenen assignats. En el cas de ASSB, veu el llistat de tots els casos.

Cerca primària: SSB i ASSB. SSB pot fer una cerca primària per nom o cognoms de l'usuari, o pel nom de les situacions del cas. ASSB pot cercar casos pel seu identificador.

Cerca avançada: SSB. Cerca complexa que permet escollir diverses variables tals com l'estat del cas, les dates d'obertura o de tancament, cercar per *nif* d'usuari...

Veure recursos: AREC. Veu el llistat de tots els recursos existents.

Veure recursos: AREC. Pot cercar recursos pel seu nom.

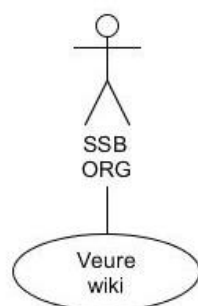
Veure professionals actius: APROF. Veure el llistat de tots els professionals actius de l'aplicació. No pot consultar les dades personals, únicament l'identificador.

Veure professionals innactius: APROF. Veure el llistat de tots els professionals innactius de l'aplicació. Són aquells que han de validar el registre a través del seu identificador i d'una clau d'activació.

Veure llistat de preguntes. ASS. Veu un resum de totes les preguntes que ha rebut en format de llistat. No es mostra el contingut de la pregunta.

**Pel que fa a l'Administrador d'enquestes, la seva pàgina principal és la del correu.*

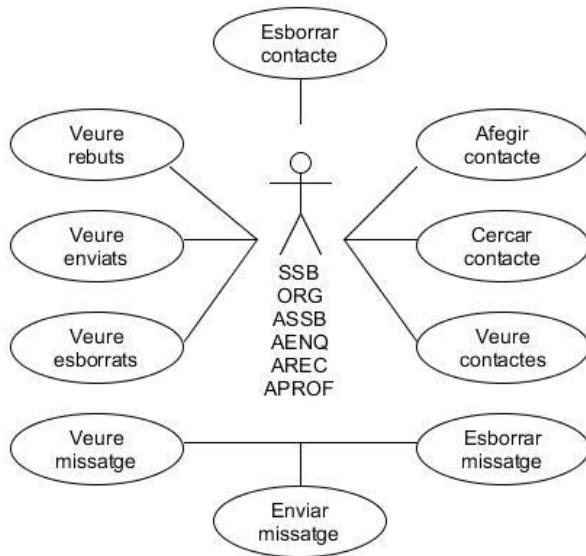
Wiki



Veure wiki: SSB i ORG. Veure la documentació de l'aplicació web. Inicialment es mostra un índex des del qual és possible accedir al punt que es desitgi de la documentació.

Correu

Conté les funcionalitats relacionades amb el sistema de missatgeria extern de l'aplicació.



Veure missatge: Tots. Consultar contingut del missatge.

Enviar missatge: Tots. Escriure un nou missatge i enviar-lo.

Esborrar missatge: Tots. Esborrar un missatge de la llista de rebuts o d'enviats.

Veure contactes: Tots. Llistat dels contactes personals.

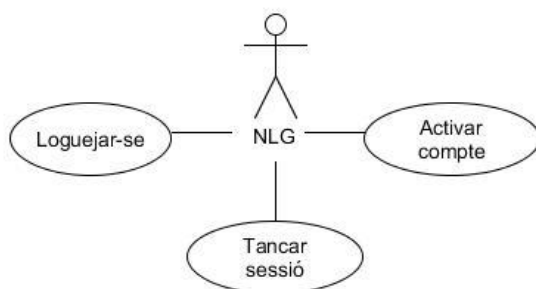
Cercar contacte: Tots. Cercar un nou professional per l'identificador, amb l'objectiu d'incorporar-lo a la llista personal de contactes.

Afegir contacte: Tots. Afegir un nou professional a la llista personal de contactes.

Esborrar contacte: Tots. Esborrar un professional existent de la llista personal de contactes.

Login

Controlador que conté les funcionalitats relacionades amb el *login* de l'aplicació, amb el tancament de sessió, i amb l'activació d'un nou compte.



Loguejar-se: Accedir a la aplicació web a través de l'identificador i contrasenya.

Tancar sessió: Sortir de l'aplicació web tancant la sessió actual.

Activar compte: Registrar-se a l'aplicació web a través d'un identificador i clau d'activació prèviament donats.

3.5 Diagrama de classes

Com que s'ha utilitzat MVC, tots els controladors tenen una estructura general quan es relacionen amb altres classes. Per aquest motiu no és necessari definir aquestes relacions cas a cas.

3.5.1 Router, controladors i classes comunes

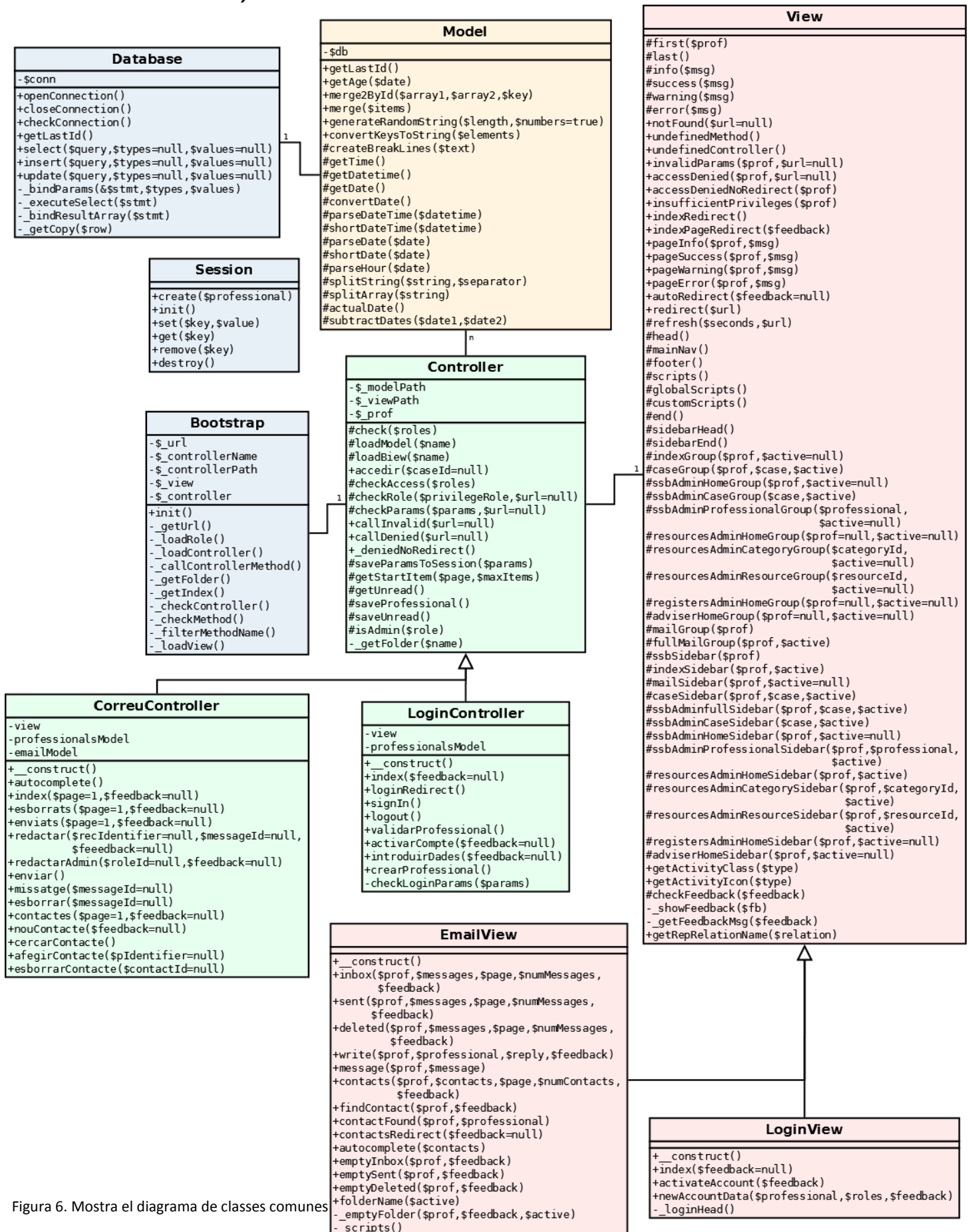


Figura 6. Mostra el diagrama de classes comunes

La classe *Bootstrap* és l'anomenat *router*. És qui rep la *url* i s'encarrega de distribuir les peticions als controladors adequats. Les classes *Email* i *Login*, tant els controladors com les vistes són comunes per tots els rols. A continuació anem a observar el diagrama de classes dels controladors de cada rol.

3.5.2 Per rol i controladors

Controladors dels rols SSB i ORG

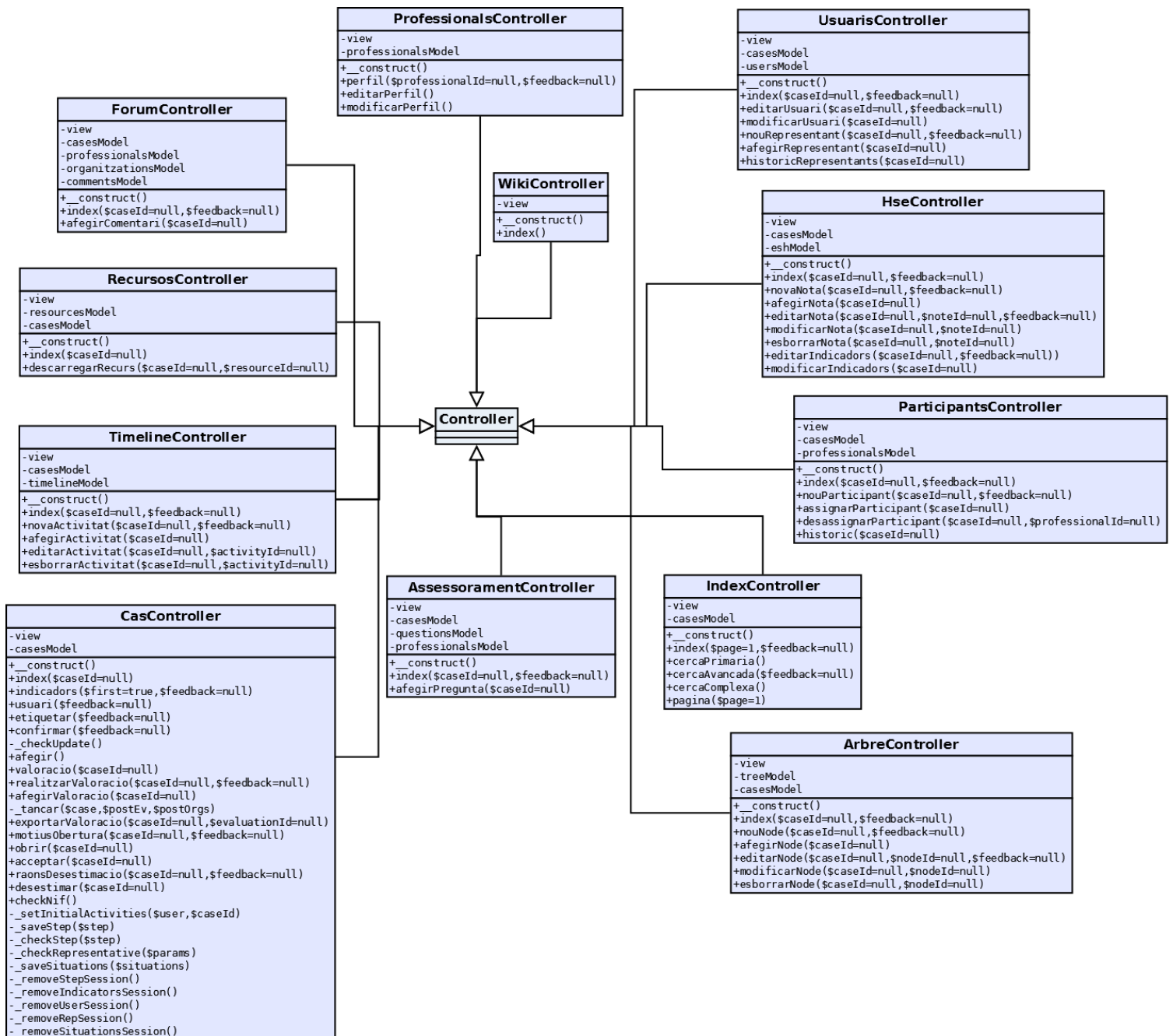


Figura 7. Controladors presents en els rols SSB i ORG. Tots hereten de la classe *Controller*

Controladors del rol ASSB

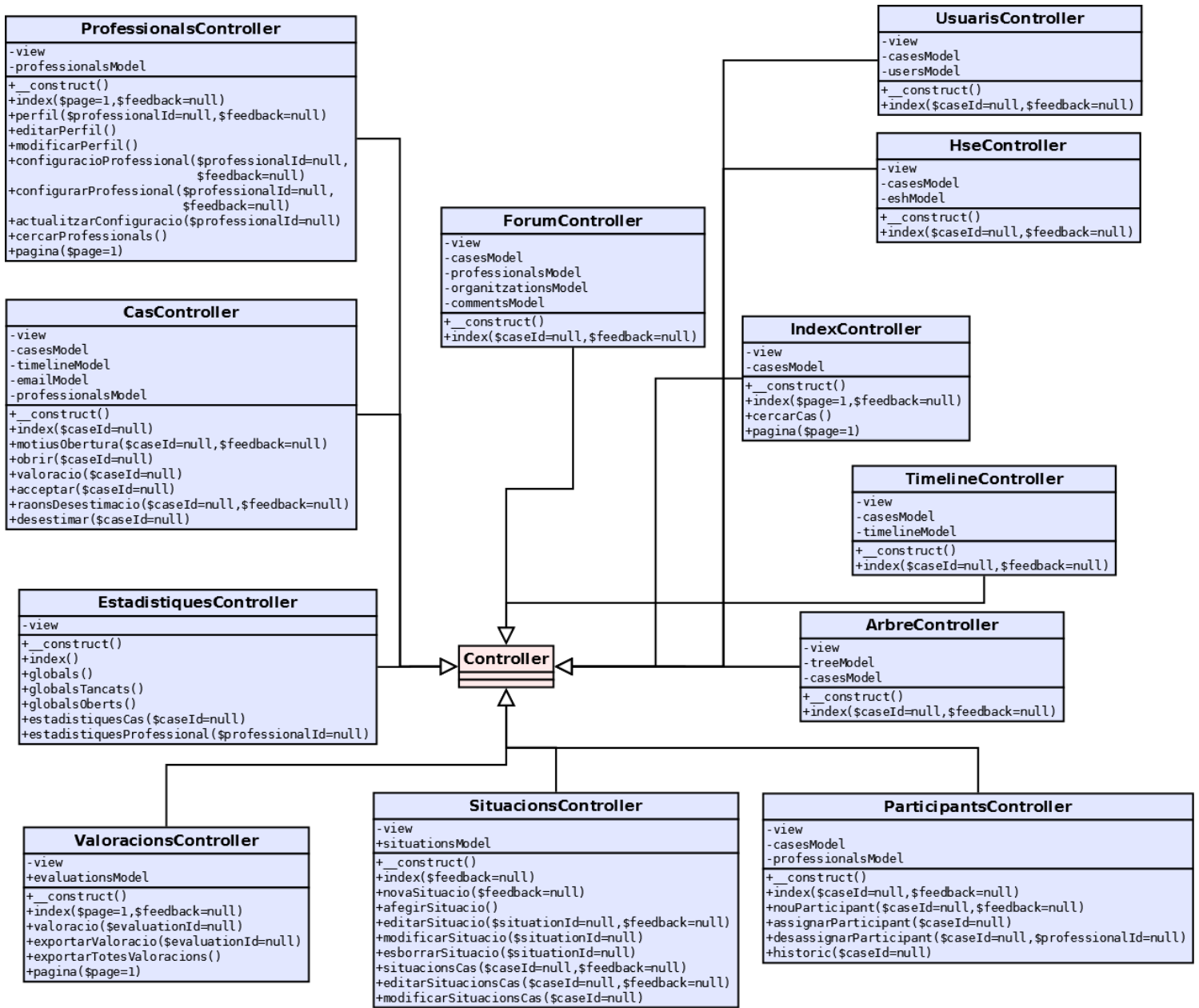


Figura 8. Controladors presents en el rol ASSB. Tots hereten de la classe *Controller*

Controladors del rol d'Administrador de recursos

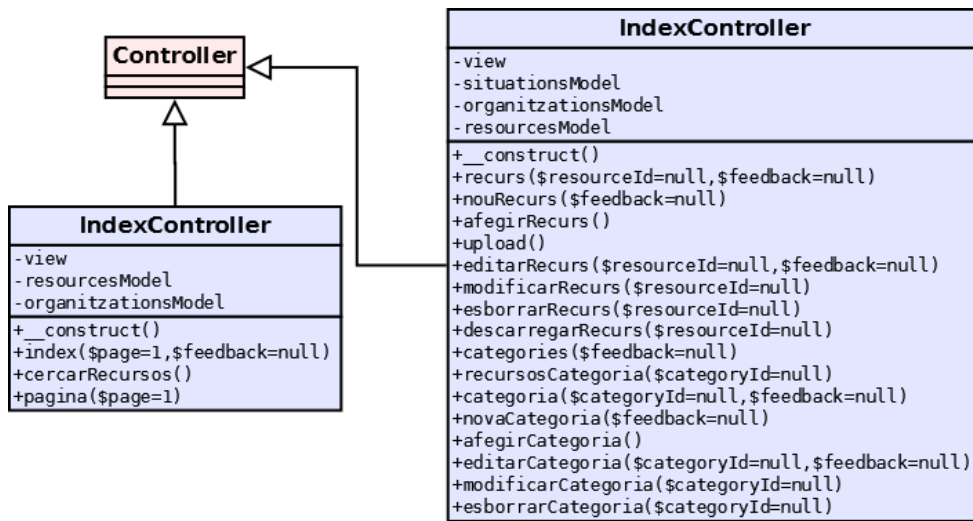


Figura 9. Controladors presents en el rol AREC. Tots hereten de la classe *Controller*

Controladors del rol d'Administrador de professionals

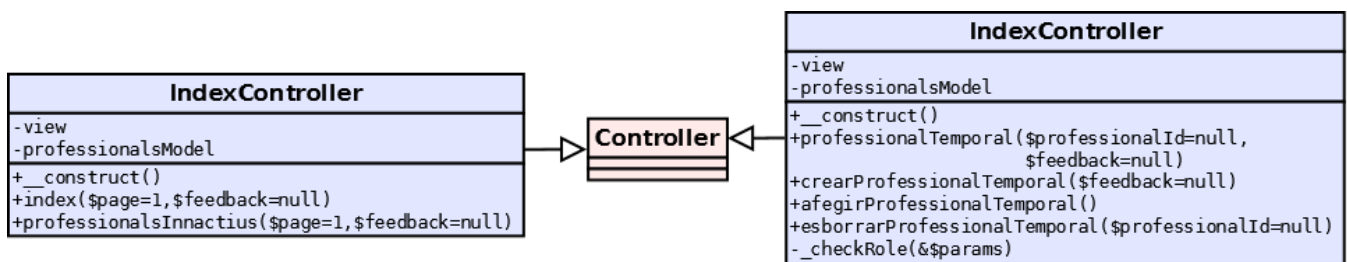


Figura 10. Controladors presents en el rol AREC. Tots hereten de la classe *Controller*

Controladors del rol d'assessor

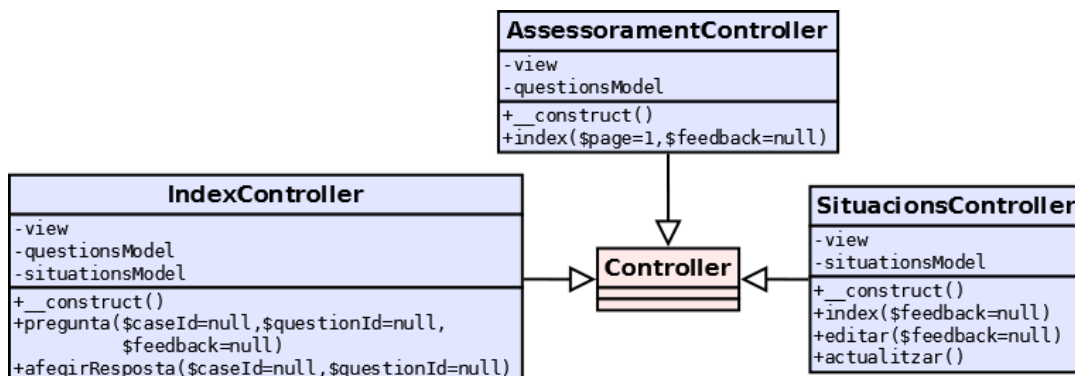


Figura 11. Controladors presents en el rol d'assessor. Tots hereten de la classe *Controller*

El rol d'administrador d'enquestes únicament utilitza els controladors Login i Email

3.5.3 Per rol i views

Com hem vist anteriorment, les vistes de les classes *Login* i *Email* són comunes per tots els rols de l'aplicació. Com que les funcionalitats no ho són, els controladors canvien, i per tant, les vistes també. Anem a veure amb més detall els diagrames de classes de cada rol.

Controladors dels rols SSB i ORG

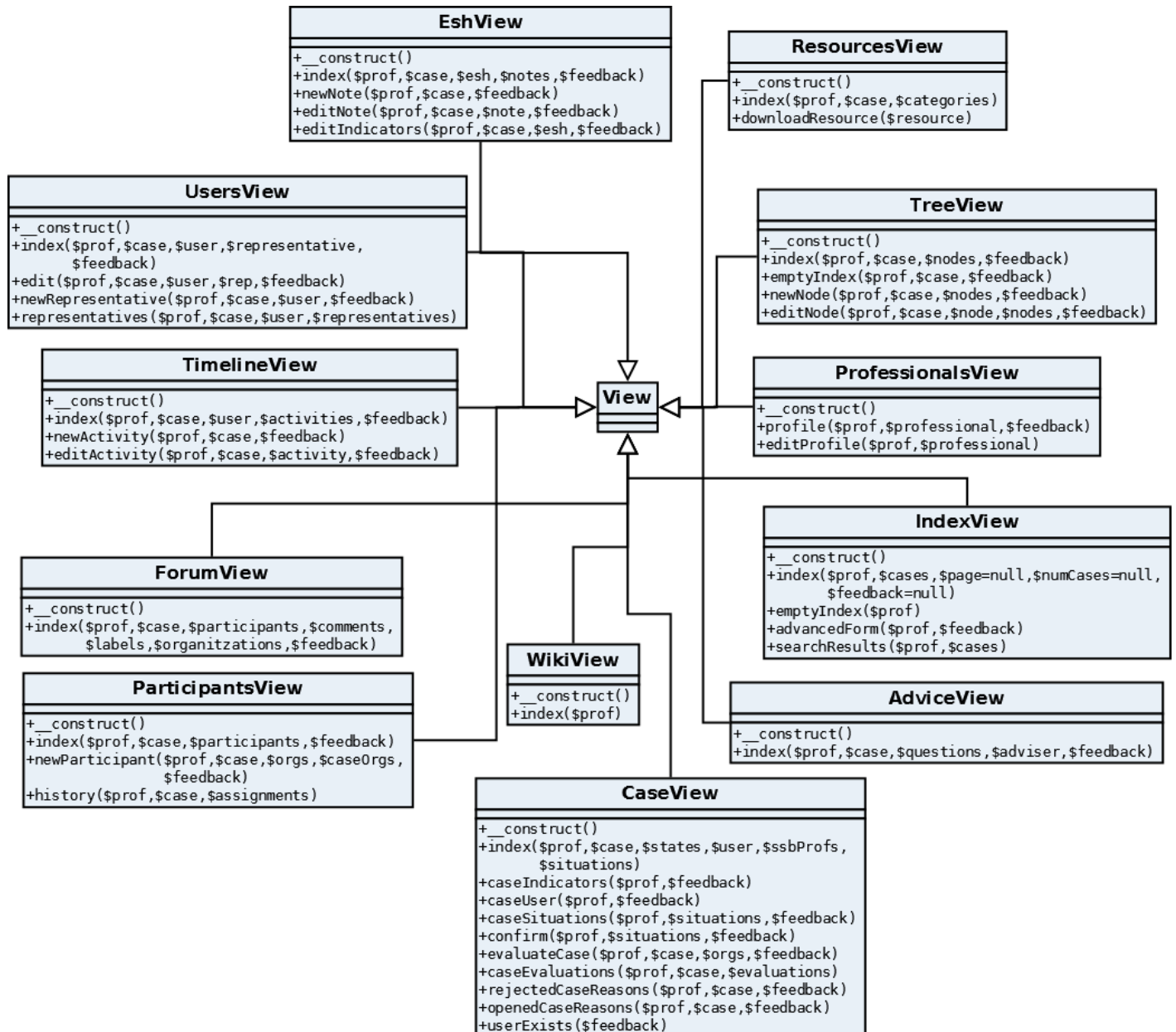


Figura 12. Vistes presents en els rols SSB i ORG. Tots hereten de la classe View

Controladors del rol ASSB

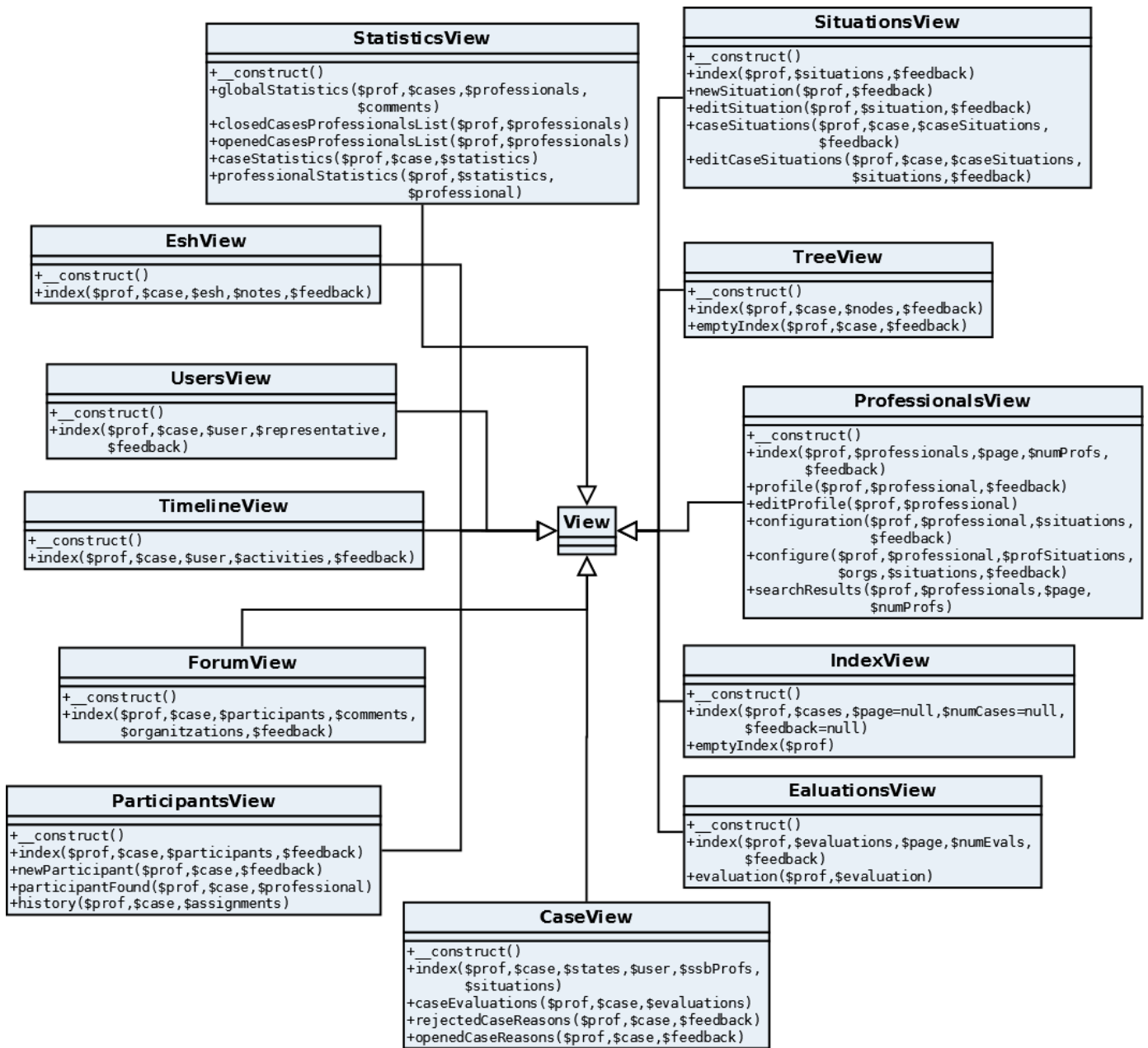


Figura 13. Vistes presents en el rol ASSB. Tots hereten de la classe View

Controladors del rol d'Administrador de recursos

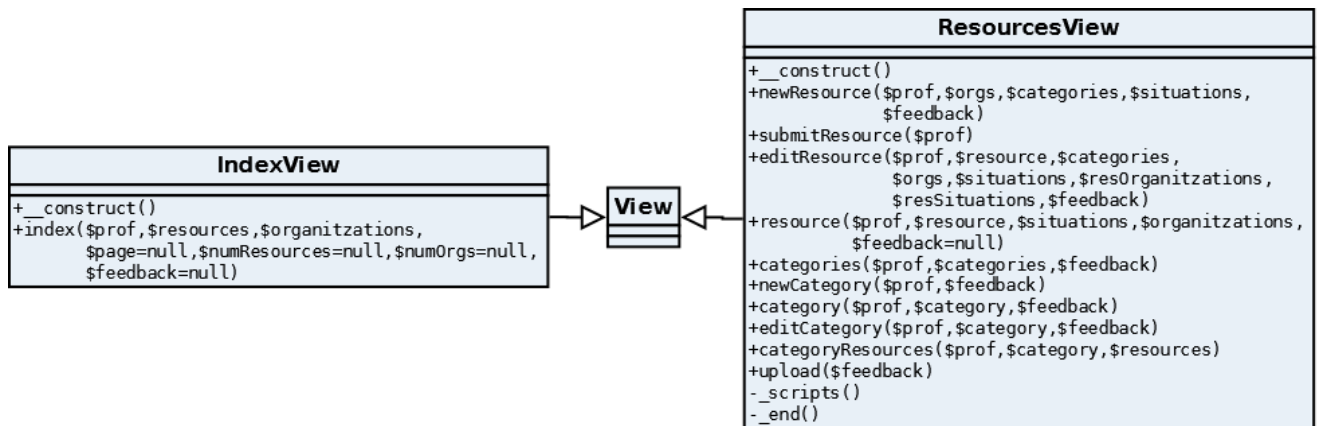


Figura 14. Vistes presents en el rol AREC. Tots hereten de la classe View

Controladors del rol d'Administrador de professionals

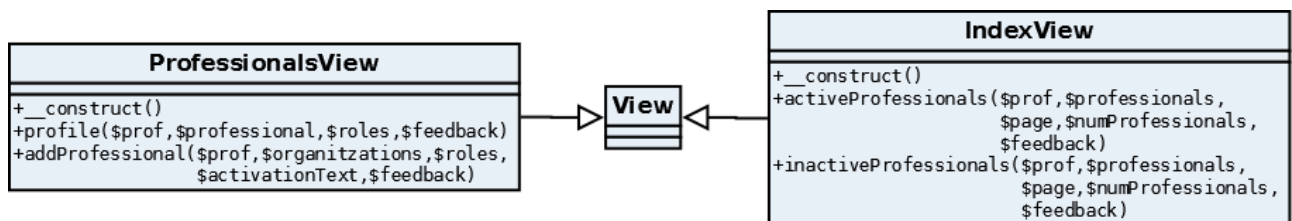


Figura 15. Vistes presents en el rol AREC. Tots hereten de la classe View

Controladors del rol d'assessor

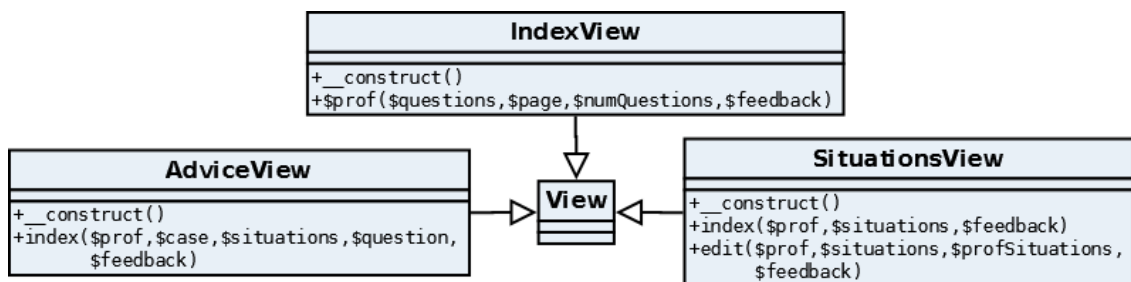


Figura 16. Vistes presents en el rol d'assessor. Tots hereten de la classe View

El rol d'administrador d'enquestes únicament utilitza les vistes Login i Email

Pel que fa als models i altres classes utilitzades com a *helpers*, s'ha decidit no incloure-les en els diagrames de classes.

3.6 Simulació d'un cas i diagrames de seqüència

3.6.1 Dades

Una simulació fictícia d'un cas, pot ser una manera de fer més comprensible tota la feina realitzada. A continuació, es mostrarà la introducció i evolució d'un cas i els actors que hi intervenen. Acompanyarem les tasques realitzades amb un conjunt de captures de pantalla.

Eduard Romeu (*e.romeu*): Professional d'organisme associat d'un centre escolar.

Josep Moya (*j.moya*): Professional de Serveis Socials Bàsics.

Carles Garcia Pérez: Noi de 15 anys que serà l'usuari del cas. Rep assetjament escolar.

3.6.2 Tasques

3.6.2.1 Afegir cas

Eduard Romeu és professor d'un institut. Dona classes a alumnes de 2n d'ESO. Detecta indicis de que un alumne pot estar rebent assetjament escolar, així que decideix informar. Després de logar-se a l'aplicació web, es dirigeix al menú principal (situat a la part esquerra) i prem l'opció d'*Afegir Cas*.

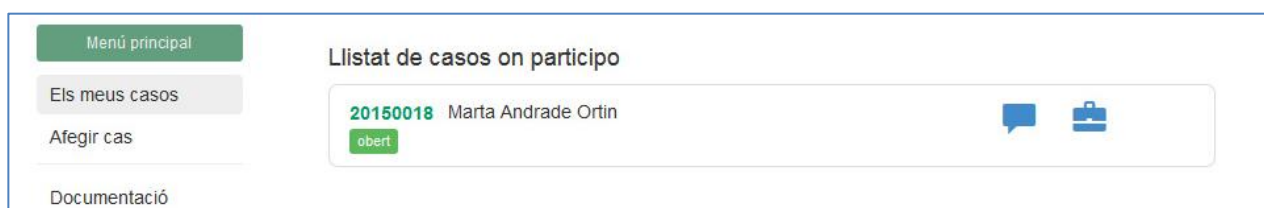
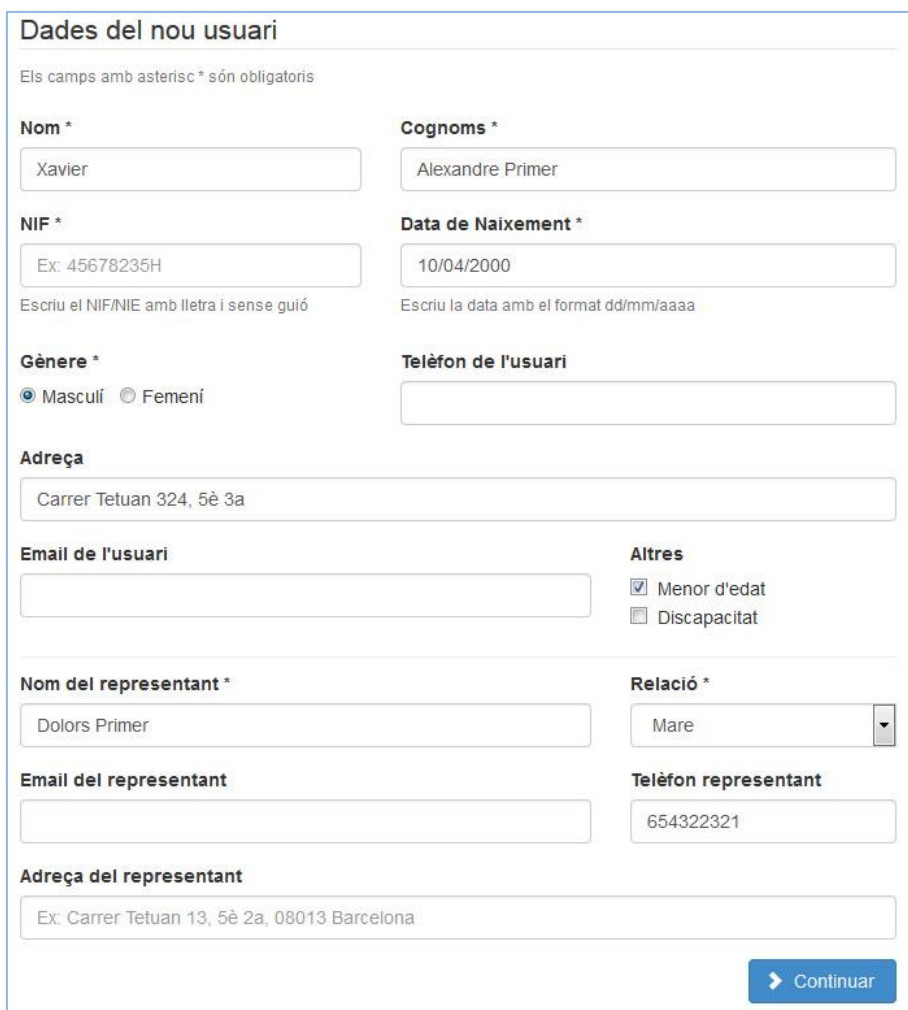


Figura 17. Menú principal de la web per a professionals d'organisme associat.

A continuació ha d'introduir els indicadors que associa a la problemàtica de l'alumne.

Figura 18. Formulari per introduir els indicadors del nou cas

Un cop emplenats, es prem el botó continuar i s'arriba al pas 2: introduir les dades de l'usuari.



Dades del nou usuari

Els camps amb asterisc * són obligatoris

Nom * **Cognoms ***

NIF * **Data de Naixement ***

Escriu el NIF/NIE amb lletra i sense guió Escriu la data amb el format dd/mm/aaaa

Gènere * ☒ Masculí ☐ Femení **Telèfon de l'usuari**

Adreça

Email de l'usuari **Altres** ☒ Menor d'edat ☐ Discapacitat

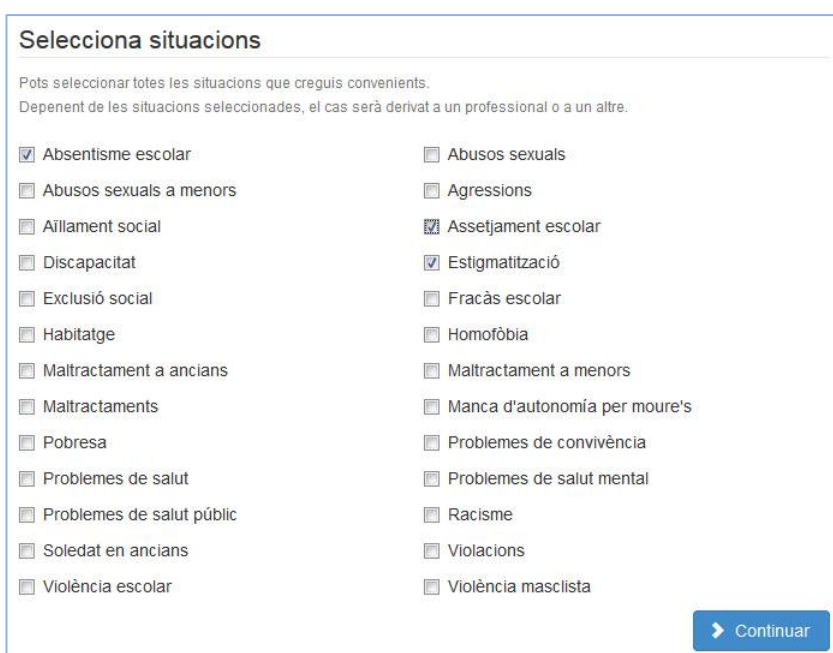
Nom del representant * **Relació ***

Email del representant **Telèfon representant**

Adreça del representant

[▶ Continuar](#)

Figura 19. Formulari per emplenar les dades del nou usuari



Selecciona situacions

Pots seleccionar totes les situacions que creguis convenientes.
Depenent de les situacions seleccionades, el cas serà derivat a un professional o a un altre.

☒ Absentisme escolar	☐ Abusos sexuals
☐ Abusos sexuals a menors	☐ Agressions
☐ Aïllament social	☒ Assetjament escolar
☐ Discapacitat	☒ Estigmatització
☐ Exclusió social	☐ Fracàs escolar
☐ Habitatge	☐ Homofòbia
☐ Maltractament a ancians	☐ Maltractament a menors
☐ Maltractaments	☐ Manca d'autonomia per moure's
☐ Pobresa	☐ Problemes de convivència
☐ Problemes de salut	☐ Problemes de salut mental
☐ Problemes de salut públic	☐ Racisme
☐ Soledat en ancians	☐ Violacions
☐ Violència escolar	☐ Violència masclista

[▶ Continuar](#)

Com que és menor d'edat, també emplena les dades del seu representant legal, en aquest la mare. En prémer continuar s'arriba al penúltim pas: seleccionar les situacions.

Figura 20. Situacions.

Confirmació del conjunt de dades

Indicadors

Socials

Indicis clars d'assetjament escolar per part de la resta d'alumnes del centre. A conseqüència d'aquest motiu he notat que les seves notes i autoestima han disminuït dràsticament.

Usuari

Nom

Xavier

Cognoms

Alexandre Primer

Data de naixement

10/04/2000

Sexe

Masculí

NIF

62655805X

Adreça

Carrer Tetuan 324, 5è 3a

Representant

Nom

Dolors Primer

Relació

Mare

Telèfon

654322321

Situacions

Absentisme escolar

Assetjament escolar

Estigmatització

Derivar a Serveis Socials Bàsics

Figura 21. Confirmació del conjunt de dades introduïdes.

Finalment, el professor del centre escolar té l'opció de revisar totes les dades que ha introduït. En cas que consideri que n'hi ha alguna que és incorrecte o que pot ser complementada amb més informació, pot anar al pas en qüestió amb la paginació situada a la part inferior de la pàgina.

Pas 1 – Indicadors

Pas 2 – Usuari

Pas 3 – Situacions

Pas 4 – Confirmar

Si vols actualitzar alguna dada, prem el botó del pas en qüestió

Figura 22. Paginació que possibilita actualitza informació de cada pas

L'última tasca de la derivació d'un cas, consistiria en derivar-lo a Serveis Socials Bàsics. Prement el botó verd aconseguiríem que el cas fes els primers passos per tirar endavant i poder resoldre'l.

Diagrama de seqüència

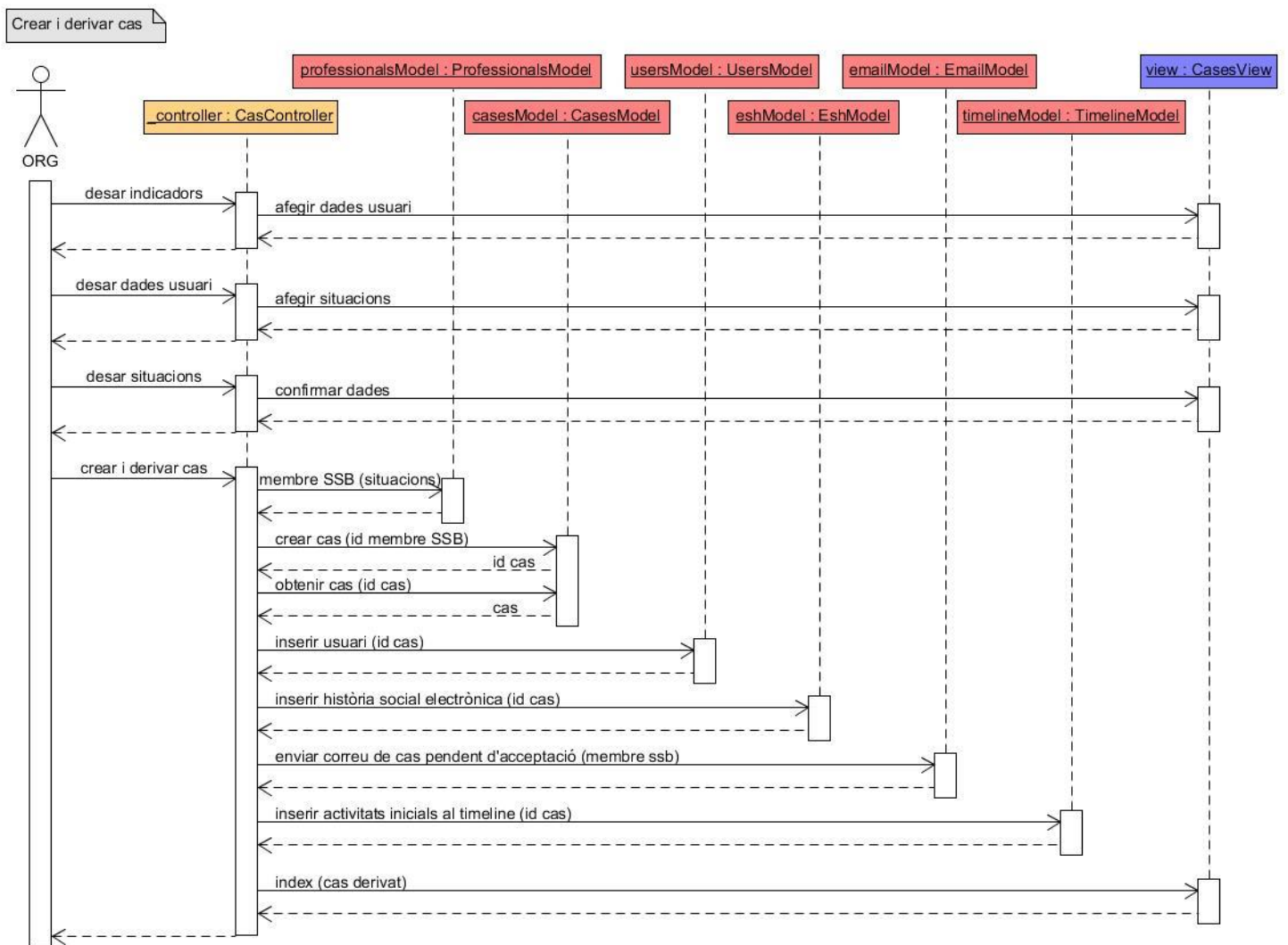


Figura 23. Diagrama de seqüència de “Crear i derivar un cas”

Tant els indicadors, com les dades de l’usuari com les situacions es guarden en les variables de sessió mentre el cas no és derivat en l’últim pas de la confirmació. És llavors quan es produeix la interacció amb els models per inserir les dades a la BD.

Per crear-lo, primer de tot obtindrem el membre més adequat de SSB gràcies a les situacions introduïdes. Posteriorment es crea el cas, s’introdueixen les relacions cas-professionals, etc...

Finalment es mostrarà un feedback a l’usuari informant-lo de que el cas ha estat derivat correctament i ara resta pendent d’acceptació.

Cal comentar que en el diagrama de seqüència no s'utilitzen els noms reals de les funcions ni tots els paràmetres utilitzats. Únicament els que es consideren que ajuden a comprendre el flux de les accions.

3.6.2.2 Acceptar el cas

Josep Moya que és el membre de Serveis Socials Bàsics que ha rebut el cas, rebrà un correu informant que té un cas pendent d'acceptació.

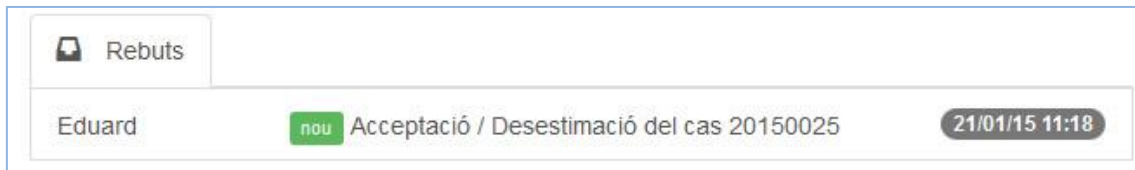


Figura 24. Nou correu rebut. Safata d'entrada.

En l'índex de la web, apareixerà en la part superior del llistat de casos d'un color taronja que el diferencia dels casos oberts i tancats.

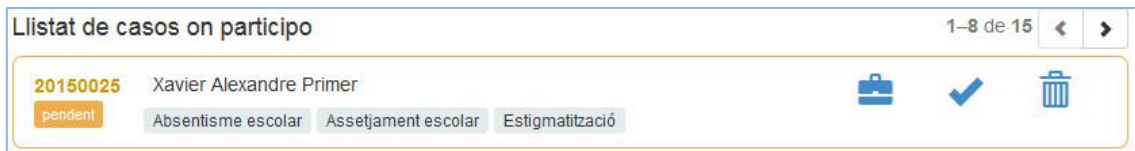


Figura 25. Llistat de casos d'un professional de SSB. Es mostra un cas pendent d'acceptació.

Després de consultar la informació que motiva l'enviament, en Josep pot decidir acceptar el cas.

Indicadors	
Salut	Sense informació
Salut mental	Sense informació
Socials	Indicis clars d'assejament escolar per part de la resta d'alumnes del centre. A conseqüència d'aquest motiu he notat que les seves notes i autoestima han disminuït dràsticament.
Risc	Sense informació

Figura 26. Història social electrònica de l'usuari d'un cas.

Per fer-ho premerà l'icona en forma de *tick* i es mostrarà un missatge modal de confirmació.

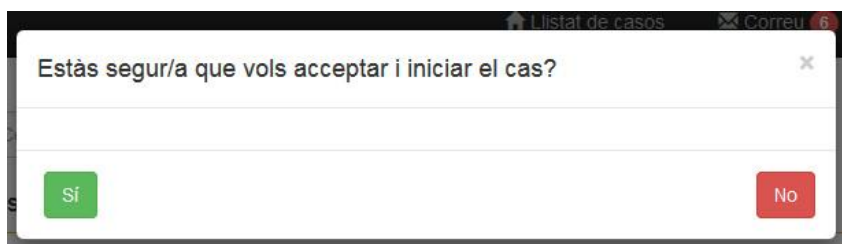


Figura 27. Missatge modal de confirmació d'acceptació del cas

Diagrama de seqüència

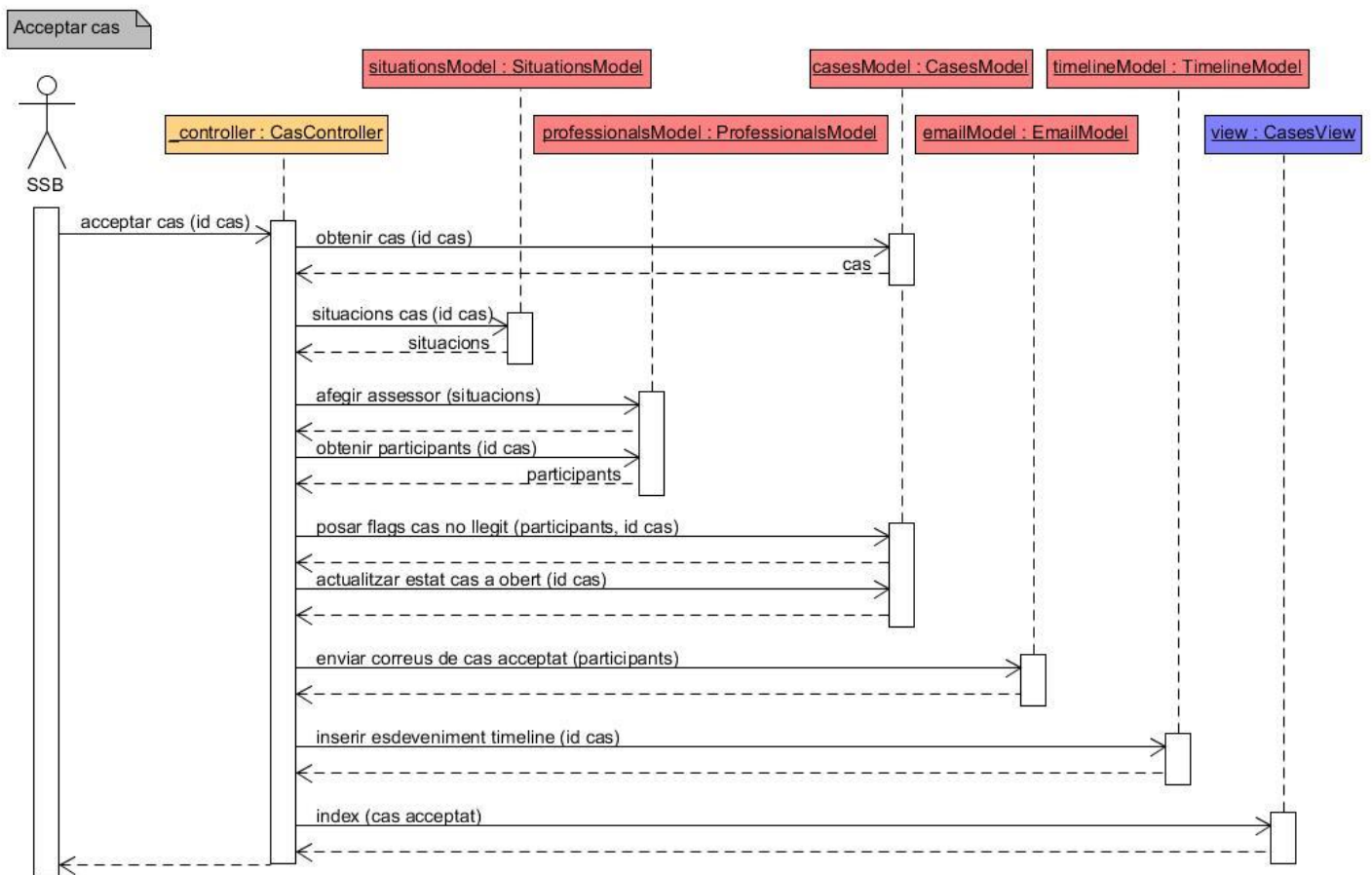


Figura 28. Diagrama de seqüència d'acceptació d'un cas per part de SSB

Primer de tot s'obté el cas per tal de fer les comprovacions oportunes (que realment el professional estigui assignat, cas en estat pendent..).

A continuació obtindrem les situacions per tal de poder assignar l'assessor que més s'ajusti a aquestes situacions. També obtindrem els participants per tal d'informar-los amb un correu que el cas ha estat obert i es produiran varis canvis en els estats de les assignacions professionals – cas i en el propi estat del cas.

Finalment deixarem constància de la data d'obertura del cas en el *timeline* i cridarem a la vista per redirigir-nos a l'índex tot mostrant el *feedback* corresponent.

3.6.2.3 Assignar participants i comentar en el fòrum d'un cas

Des de la pàgina inicial de la web (llistat de casos), podem accedir directament al *fòrum de treball* d'un cas prement l'ícono que té forma de comentari.

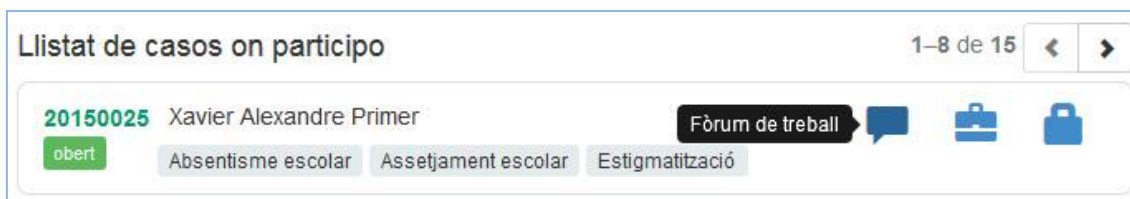


Figura 29. Selecció de l'ícono del fòrum d'un cas

Un cop dins del fòrum, i com que és el primer cop que hi accedim, no observarem cap comentari ni conversa. Per a enviar un comentari és necessari que el participant estigui assignat al cas.



Per tal d'anar a la pàgina d'assignació de participants ho podem fer de dues maneres. O bé a través del menú esquerre seleccionant l'opció "*Participants*", o bé a través de la barra lateral dreta de participants (present al fòrum), prement el botó "*Assignar*".

Figura 30. *Sidebar* dreta present dins del fòrum d'un cas, que permet visualitzar i accedir a la gestió de participants



Dins de la pàgina de participants, podem fer una assignació directa per organisme. El membre de SSB no coneix, a priori, quin és el membre d'aquell organisme que quedarà assignat al cas.

El membre que quedarà assignat, és aquell que té menys càrrega de feina (casos assignats) en aquell precís moment.

Així doncs, prement una de les opcions del *radio* i posteriorment clicant el botó "*Assignar al cas*" ja hauréu assignat un nou professional.

Figura 31. Pàgina d'assignació de participants a un cas

La pàgina redirigirà a la llista de participants assignats mostrant el feedback apropiat.



The screenshot shows a web interface for managing participants. At the top, a green notification bar with a checkmark icon and the text "Assignat al cas" (Assigned to the case) is displayed. Below this, there is a blue button labeled "+ Afegir nou participant" (Add new participant) and a link labeled "Veure històric" (View history). The main content is a table with three rows of assigned participants:

Participant Name	Assigned Location	Action
Elena Gómez	Centre de Salut Mental Infantil i Juvenil (CSMIJ)	[Trash icon]
Eduard Romeu Garcia	Centre Escolar (CE)	[Trash icon]
Marcos Catalán	Serveis Socials Bàsics (SSB)	

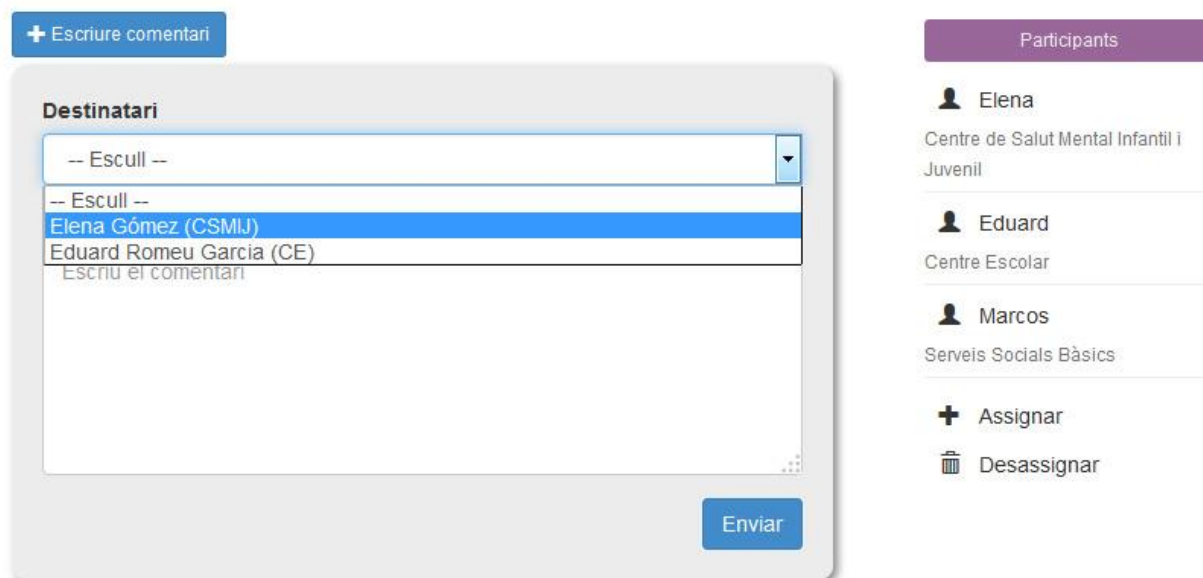
Figura 32. Llista de participants d'un cas. Es mostra el feedback de correcta assignació d'un nou participant.

Hem assignat una professional del *Centre de Salut Mental i Juvenil (CSMIJ)*. Podem de comprovar que s'anomena Elena Gómez.

El que volem, a continuació, és enviar un comentari a l'Elena tot preguntant-li sobre si el nostre usuari és present a la seva base de dades. Estem intentant recopilar informació per tal de poder començar a entendre les problemàtiques associades a l'usuari del cas.

Així doncs, tornem al fòrum del cas prement l'opció del menú esquerra.

Dins del fòrum seleccionem el botó "*Afegir comentari*" i ens disposem a escriure el missatge que voldrem enviar. Abans, però, seria adient seleccionar el destinatari del comentari, en aquest cas l'Elena Gómez.



The screenshot shows the "Afegir comentari" (Add comment) interface. On the left, there is a button labeled "+ Escriure comentari" (Write comment). Below it, a modal window titled "Destinatari" (Recipient) is open, displaying a dropdown menu with the following options: "-- Escull --", "Elena Gómez (CSMIJ)", and "Eduard Romeu Garcia (CE)". The "Elena Gómez (CSMIJ)" option is currently selected. Below the dropdown, there is a text area labeled "Escriu el comentari" (Write the comment) and a blue "Enviar" (Send) button. On the right side of the interface, there is a purple header labeled "Participants" and a list of participants: Elena (Centre de Salut Mental Infantil i Juvenil), Eduard (Centre Escolar), and Marcos (Serveis Socials Bàsics). Below the list, there are two buttons: "+ Assignar" (Assign) and "[Trash icon] Desassignar" (Unassign).

Figura 33. Llista de participants d'un cas. Es mostra el feedback de correcta assignació d'un nou participant.

Finalment, escriurem el comentari i li donarem a enviar. L'Elena, quan iniciï sessió, rebrà dues notificacions: assignació a un nou cas i un comentari no llegit. Restarem a l'espera de la seva resposta tot i que mentrestant podem establir una conversa amb un altre participant.

Diagrames de seqüència

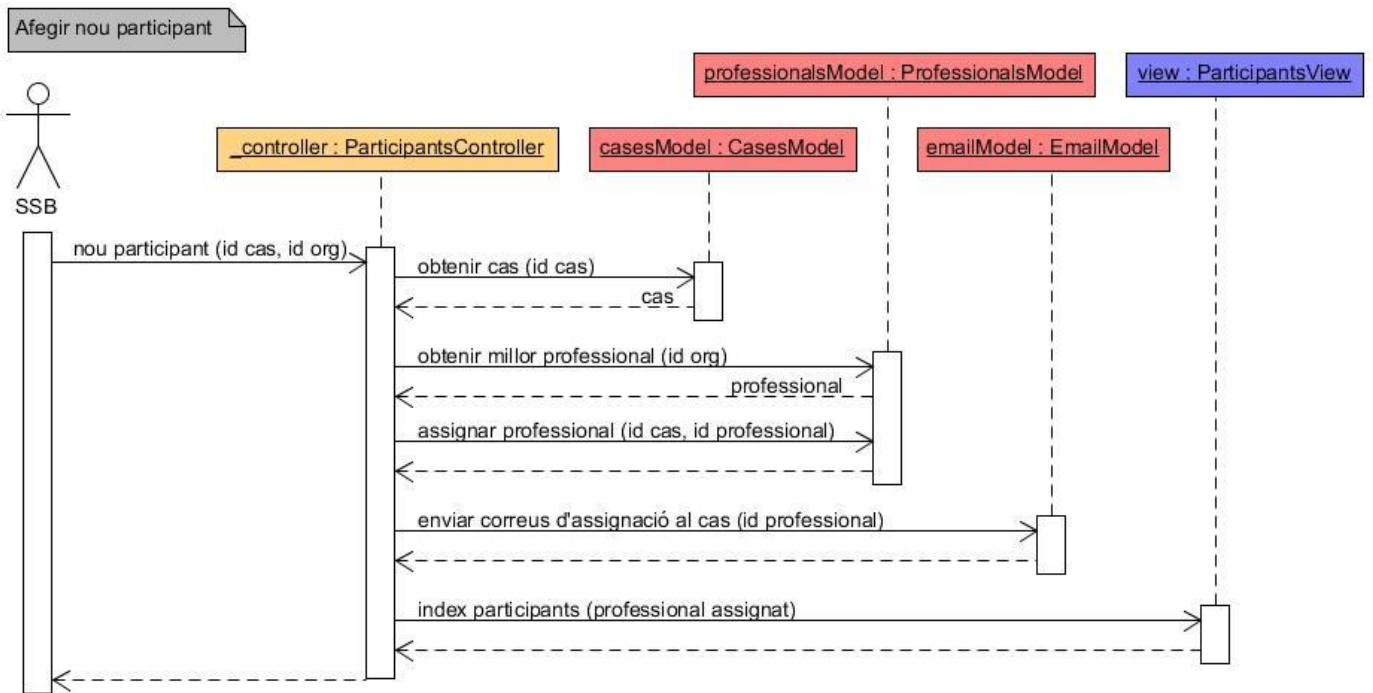


Figura 34. Diagrama de seqüència d'afegir un nou participant a un cas

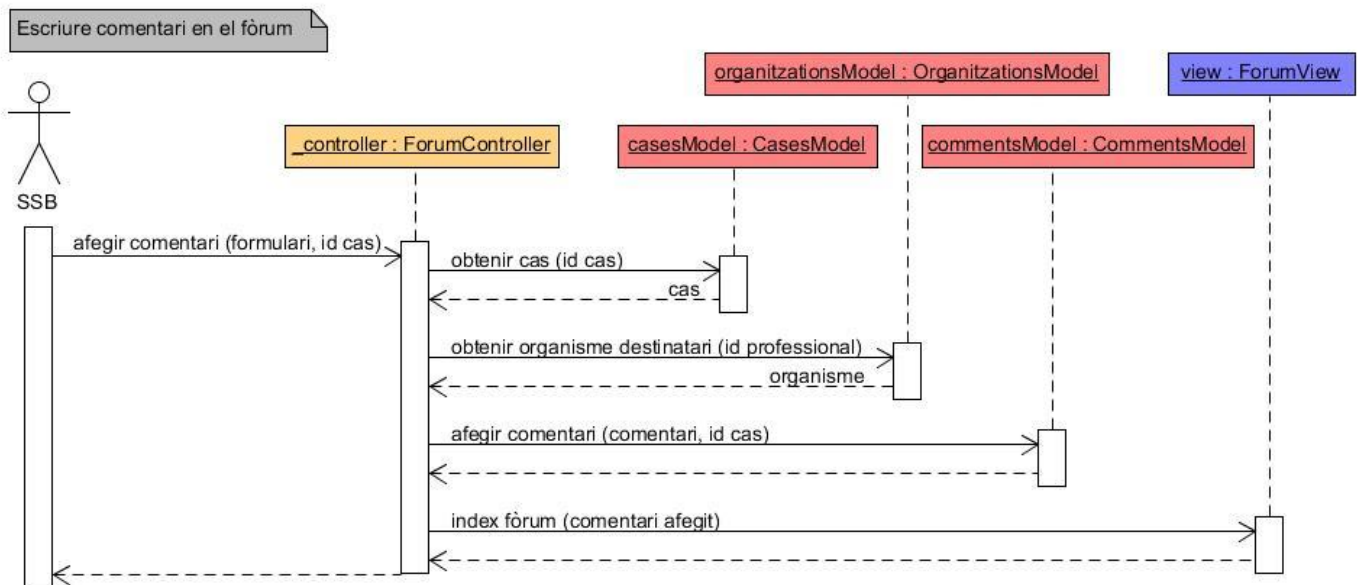


Figura 35. Diagrama de seqüència d'afegir comentari al fòrum d'un cas

3.6.2.4 Desenvolupar i tancar un cas

El transcurs natural del cas seria el següent: Josep Moya, membre de Serveis Socials Bàsics, obtindria informació de diversos organismes. Aquesta informació podria ser publicada en format de notes personals en l'història social electrònica.

A mesura que es detectessin fets especialment rellevants, podrien ser inclosos com a nous esdeveniments en el *timeline* del cas.

També l'arbre sociogenealògic jugaria un paper important, intentar esquematitzar les relacions familiars i socials de l'usuari per intentar veure de manera més clara quins factors han pogut determinar la problemàtica actual.

El desenvolupament del cas, però, és lliure i difícilment predictable. Cada professional haurà d'anar "improvitzant" a conseqüència de les situacions que se li presentin. Així, per tant, també se'm fa molt complicat establir quan es podria considerar que el cas està llest per ser tancat. Per la següent simulació, però, haurem de suposar aquest fet.

Doncs ara ens trobem en el moment en que Josep Moya vol tancar el cas. Per a fer-ho, haurà de seleccionar l'ícono de cademat present al llistat de casos principal del cas en qüestió.



Figura 36. Prement l'ícono de tancament del cas en el llistat general de casos.

A continuació s'obrirà una nova pàgina per tal d'indicar els motius de tancament, i per fer una valoració dels organismes implicats.

Valoració final del cas

Motiu de tancament *

El cas ha estat resolt satisfactòriament.

Valoració dels organismes implicats *

Centre de Salut Mental Infantil i Juvenil (CSMIJ)	M'ha servit totalment
Centre Escolar (CE)	-- Escull -- -- Escull -- No m'ha servit M'ha servit parcialment M'ha servit totalment

Figura 37. Formulari de tancament d'un cas

Com es pot observar, cada organisme pot ser puntuat respecte 3 opcions diferents: No m'ha servit, m'ha servit parcialment i m'ha servit totalment.

L'objectiu d'aquestes puntuacions és doble, detectar ràpidament errors de procediment i comunicació amb els diversos organismes, i també per poder fer una futura explotació de dades.

Un cop omplertes aquestes dades, la segona part del formulari ens demanarà d'incloure altres comentaris de manera opcional.

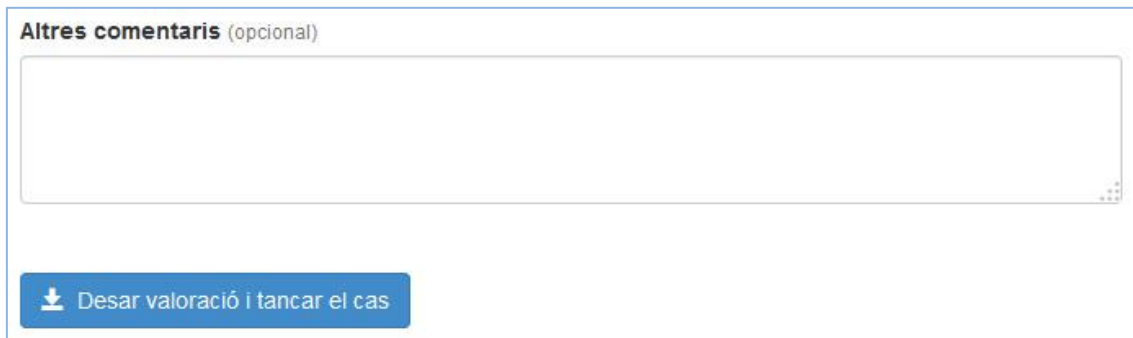
Aquesta imatge mostra una interfície web per a tancar un cas. A la part superior, hi ha el títol "Altres comentaris (opcional)" en negreta. Just a sota, hi ha un camp de text rectangular gran i buit per a introduir comentaris. A la part inferior del formulari, hi ha un botó blau amb el text "Desar valoració i tancar el cas" i un icona de descàrrega a l'esquerra.

Figura 38. Formulari de tancament d'un cas

Finalment haurem de prémer el botó "*Desar valoració i tancar el cas*" i la web redirigirà cap a la pàgina inicial tot mostrant-nos, ara, que el cas ha estat tancat.

Aquesta imatge mostra una barra d'informació d'un cas tancat. A l'esquerra, hi ha el número de cas "20150025" en vermell i el nom "Xavier Alexandre Primer". Just a sota, hi ha un botó vermell amb el text "tancat". A la dreta del botó, hi ha tres botons grisos: "Absentisme escolar", "Assetjament escolar" i "Estigmatització". A la part dreta de la barra, hi ha tres icones: un discurs, una llista i un triangle cap amunt.

Figura 39. Cas tancat present en el llistat de casos de la pàgina inicial.

Diagrama de seqüència

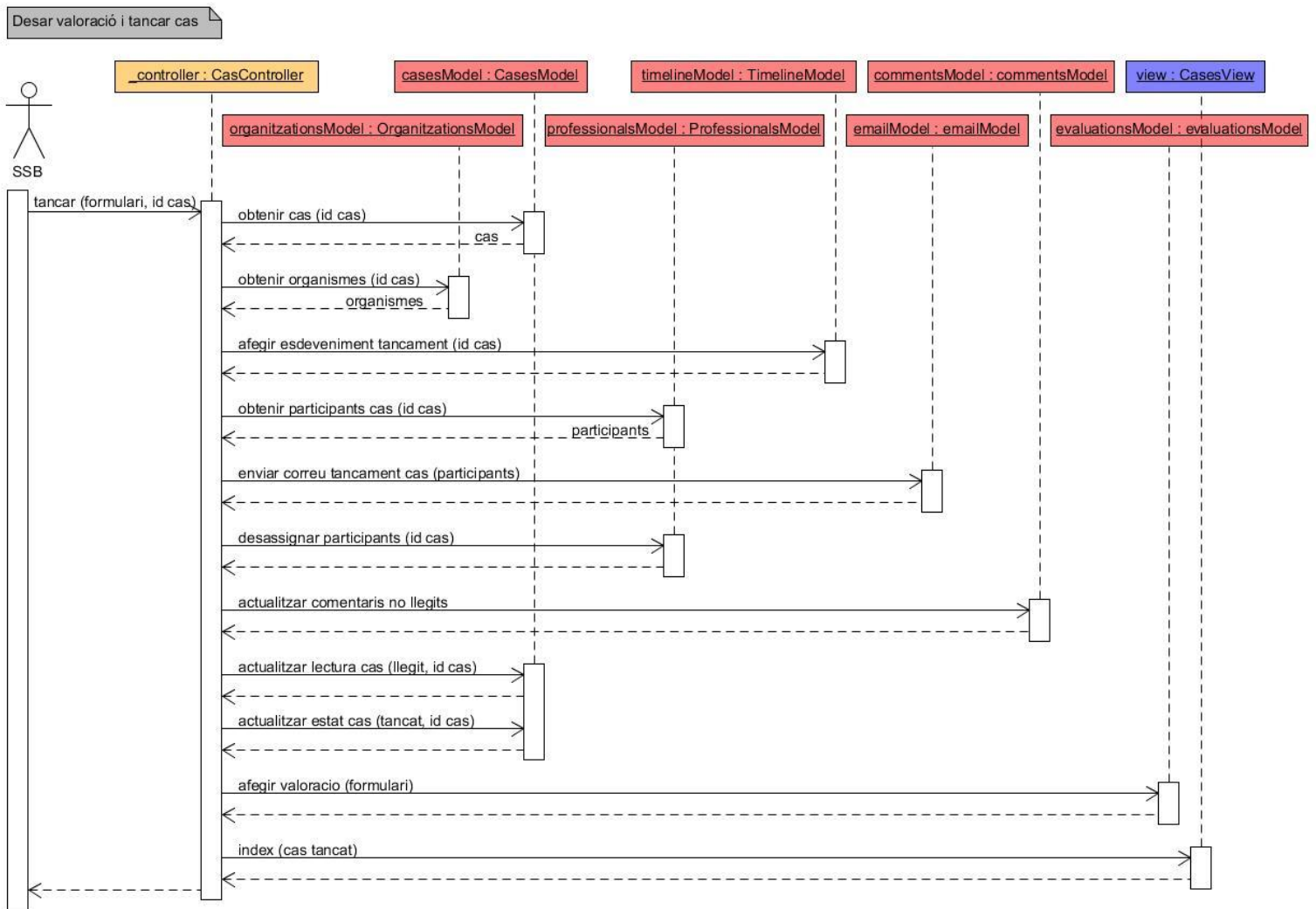


Figura 40. Diagrama de seqüència de tancament d'un cas

3.7 Seguretat i validació de dades

3.7.1 Introducció

En termes de seguretat, hem intentat solucionar els principals problemes d'injecció d'*SQL* i d'*HTML*. Anem a veure que signifiquen i com minimitzar-los/evitar-los.

Normalment, els principals forats de seguretat que pateixen les webs que realitzen comunicació amb els servidors és a causa dels formularis. És així perquè els formularis contenen dades introduïdes per l'usuari, no són fixes i per tant estem exposats a que algú pugui intentar fer accions lesives contra nosaltres. Com actuar en aquests casos? Sanejant, validant i acceptant les dades.

Anem a veure alguns exemples que mostraran més adequadament perquè és important prendre's un temps investigant això.

Suposem que tenim un formulari on introduïm l'*identificador* i *password* per tal de poder accedir a la web (un login bàsic). Una *query* comprovació seria la següent:

```
SELECT *  
FROM professionals  
WHERE identificador = '?' AND password = '?';
```

Ara pensem que passaria si algú col·loqués en el camp d'identificador el d'un altre professional i com a *password* ho següent ' OR '1' = '1

La *query* es transformaria en

```
SELECT *  
FROM professionals  
WHERE identificador = 'david' AND password = '' OR '1' = '1';
```

Què estaria succeint? Doncs que es podria accedir sense ser l'usuari. S'estaria produïnt una injecció d'*SQL* a partir de dades introduïdes per un formulari.

Per evitar-ho procedirem a sanejar i a validar tota dada que ens arribi des del costat del client.

3.7.2 Sanejament i validació de les dades dels formularis

3.7.2.1 Sanejament

El sanejament de dades es realitza únicament en el costat del servidor. Sanejar significa extreure tot tipus de caràcters o dades no esperades que puguin alterar el funcionament esperat.

Gràcies a les opcions de sanejament que conté *php*, podem esborrar tipus de caràcters que no esperem, o bé, modificar-los. Per exemple, en el cas de que estiguem esperant un enter, sanejarem les dades esborrant tot el que no sigui propi d'enters.

`FILTER_SANITIZE_NUMBER_INT` : Esborra tots els caràcters excepte dígit i els signes de suma i resta. (Exemple)

Aplicant diversos tipus de filtres en funció de la dada esperada, estarem assegurant que tot allò que ens arriba no conté res no esperat.

3.7.2.2 Validació

La validació consisteix en verificar que les dades tinguin sentit i coherència per a l'acció que es vol realitzar amb elles. La validació es realitza tant a costat de client com a costat de servidor.

Al costat del client podem aconseguir-ho mitjançant l'*html* i *javascript*. En els *input* dels formularis els hi afegirem l'atribut *required*, que impedeix que el formulari pugui ser pujat sense haver completat aquell camp de manera obligatòria. D'això se n'encarrega el navegador. També podem comprovar les dades abans de comunicar amb el servidor, i en el cas de detectar alguna cosa errònia, cancel·lar la pujada i informar a l'usuari perquè canviï la/les dada/es.

Per exemple, volem cercar un usuari pel seu NIF. Abans d'iniciar la cerca, podem evitar feina al servidor si comprovem que el NIF introduït no és correcte, és a dir, no compleix amb el format establert per tots els NIF de l'estat espanyol.

```
$(".form-search-user").on("submit", function(ev) {  
    var check = true;  
    var nif = $( "#inp-p-nif" ).val().trim(); //obtenim el nif  
  
    if (!isNif(nif)) {  
        showFeedback('#inp-p-nif', 'NIF / NIE incorrecte!');  
        check = false;  
    }  
    return check;  
})
```

Si no fos correcte, es mostraria un missatge d'error i es cancel·laria la pujada.

Què fem en el costat del servidor respecte a la validació? El mateix. El motiu és que les dades que ens arriben des del client no són fiables tot i haver fet les comprovacions abans d'enviar-les.

3.7.2.3 Dades requerides

Aquest pas és també important ja que és el tercer en discòrdia i el que finalment determina si s'ha d'enviar un missatge d'error al professional o bé es pot continuar amb el transcurs lògic de l'aplicació. Els formularis podran ser de tres tipus, amb tots els camps requerits, amb uns quants camps requerits o amb almenys un camp requerit.

Veiem com podem fem per aglutinar sanejament, validació i dades requerides.

Cada formulari té les seves regles, si es superen, es considerarà que es pot interactuar amb la base de dades. Per a obtenir les regles, m'he ajudat d'una classe estàtica externa, **Rules.php**.

Considerem el següent exemple: Volem comprovar les dades del formulari *d'edició de dades d'un professional*.

```
private static function _editProfessionalSanitation() {
    return array
    ( 'name'      => self::_stringSanitation(),
      'surnames' => self::_stringSanitation(),
      'email'     => self::_emailSanitation()
    );
}
```

En aquest mètode es sanejaran les dades cridant als *FILTER* adequats

```
private static function _editProfessionalValidation() {
    return array
    ( 'name'      => self::_regexpValidation(self::REGEXP_SIZE, 1, 50),
      'surnames' => self::_regexpValidation(self::REGEXP_SIZE, 1, 100),
      'email'     => self::_regexpValidation(self::REGEXP_EMAIL)
    );
}
```

En aquest mètode es validaran les dades amb expressions regulars, o bé, amb *FILTER* propis de php.

```
private static function _editProfessionalRequired() {
    return array ( 'name', 'surnames' );
}
```

Finalment aquest tercer mètode ens indica quins són els camps obligatoris

Des del controlador, llavors, faríem una crida d'aquest tipus:

```
Input::postParams('editProfessional', null, MULTIPLE_REQUIRED);
```

Ens estem ajudant d'una altre classe estàtica externa, **Input.php**, que és l'encarregada d'obtenir les *rules* correctes (primer paràmetre), i de gestionar les crides a sanejament, validació i dades requerides. Si el resultat de totes les comprovacions és correcte es retornen

les dades sanejades, en cas contrari es retorna *false* i s'iniciaria la procedimentació per tal d'avisar que ha succeït un error.

Aquí, però, no acaben les comprovacions, a l'hora d'inserir els paràmetres a les *queries*, els afegirem mitjançant sentències preparades. Bàsicament consisteix en indicar el tipus de paràmetre (enter, string, etc..), i si no coincideix, es cancel·larà la consulta.

D'aquesta manera, per tant, solucionem el problema que plantejàvem a l'inici d'aquest punt. S'impedirà que el ' *OR '1=1* pugui ser interpretat com a tal i no com a string, ja que "l'escapament" de cometes fet en el sanejament ho impedirà.

Aquest tipus de validacions també impedeix l'injecció d'*html*. En caixes de text (*textarea*) on es dona més llibertat a l'usuari, també es realitza una comprovació de tags d'*html* introduïts a partir d'una llista estàtica. Així per exemple no deixem introduir tags **<script>**, fàcilment susceptibles de ser utilitzats en contra nostre.

3.7.3 Encriptació hash

Les contrassenyes dels professionals són emmagatzemades com a *hash* d'una sola via. Això s'aconsegueix gràcies a la funció **crypt** de php (entre d'altres) per tal de generar un hash a partir de la contrassenya real del professional. En el cas de que es poguessin capturar les dades de la BD, les contrassenyes emmagatzemades serien els hash i no pas les contrassenyes reals, per la qual cosa no es podrien saber les claus per poder accedir al compte d'un usuari.

3.7.4 Patró Post - Redirect - Get

El patró post-redirect-get, impedeix alguns enviaments de formularis duplicats. Consultem la inserció sense encara utilitzar-lo:

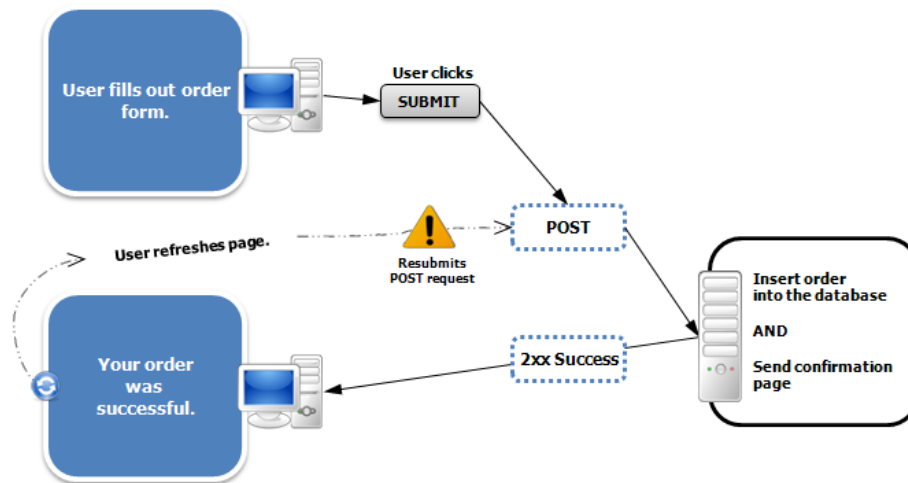


Figura 41. Inserció de dades a la BD sense el patró PRG

Usualment, es podria donar la situació en la qual un usuari de la web premés F5 després d'inserir dades a la BD. Què succeiria? Doncs que es tornarien a inserir les mateixes dades de forma errònia.

A nivell de servidor això es pot evitar amb aquest patró.

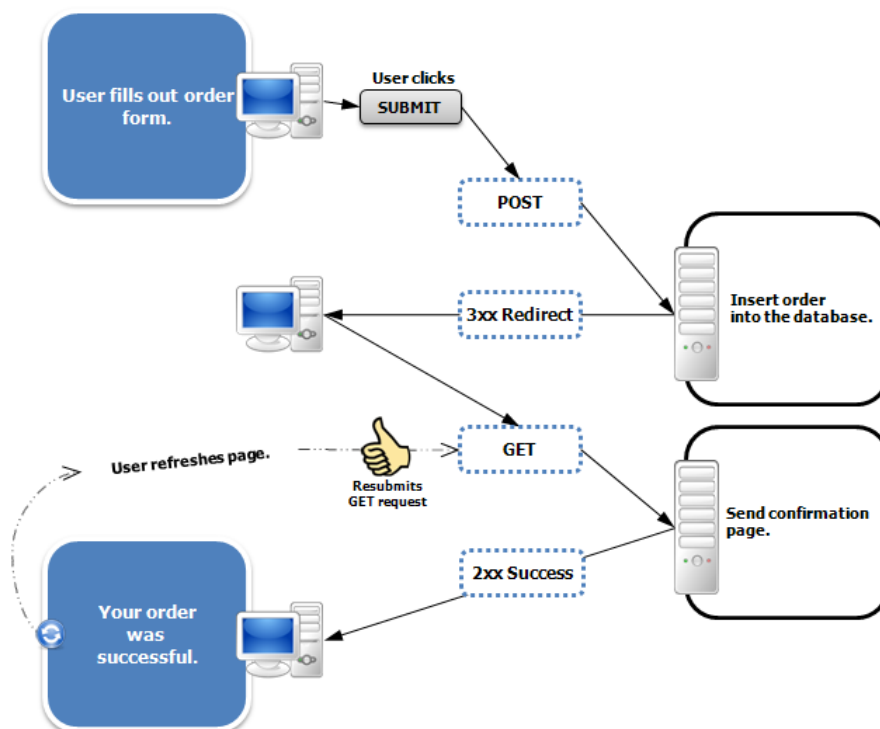


Figura 42. Inserció de dades a la BD amb el patró PRG

Simplement, després de fer una inserció, redirigirem al mateix mètode de la pàgina o a un altre, a través de l'*url*. És a dir, a partir de *get*. Ja no és possible que en actualitzar la pàgina es produeixi una duplicació de dades, ja que la funcionalitat d'inserció queda oculta entre les redireccions.

Gràcies a aquest patró també podem evitar errors comuns com el típic "Document expirat", que succeeix quan s'està intentant obtenir dades de *\$_POST* quan ja no hi són presents. Es pot donar aquest cas si un usuari seleccionés l'opció de "tirar enrere" que proporcionen els navegadors. Altres formes d'evitar això, és evitar els enviaments mitjançant POST i fer-ho amb GET. En el cas de les cerques, aquesta aplicació només n'utilitza d'aquest segon tipus.

3.7.5 Gestió de la informació. Visibilitat dels rols.

Acabem d'explicar els termes de seguretat presents en la web. Però, i si el problema no es centra en les dades, i sí en les visualitzacions de contingut i funcionalitats associades als rols?

Com fem que un rol determinat no pugui realitzar una funció que no li pertoca? Com evitem que es puguin consultar dades de casos que no es tenen assignats?

Rols

Segons el rol, cada un tindrà diferents controladors i diferents vistes. Com hem explicat anteriorment, en el dispatcher es selecciona un tipus de *path* en funció del rol, per tant ja estaríem evitant l'accés a mètodes de controladors no permesos.

En el cas de que dos rols puguin realitzar accions i funcionalitats molt semblants, no sembla lògic que s'hagi de produir una duplicació de codi amb tots i cada un dels mètodes. S'ha optat per crear un sol controlador per aquests 2 rols (Professional d'Organisme Associat i Professional de Serveis Socials Bàsics) i gestionar els mètodes de manera independent revisant sempre el rol del professional que tenim a les variables de sessió.

Informació permesa

Si observem una *url* típica dins del transcurs d'execució de la pàgina web podem observar alguna cosa semblant a això:

[.../uis/forum/index/9](#)

L'últim paràmetre, el "9", ens està indicant de quin cas es tracta. Aquesta informació és passada entre pàgines a partir de la pròpia *url* i obtinguda gràcies a GET. Ara bé, que succeiria si algun professional intentés canviar el "9" per un "10"? És a dir, com gestionem que cada professional pugui únicament consultar les dades que li pertocuen?

El que fem és una sèrie de *checks* en cada mètode de l'aplicació. Abans de realitzar cap tasca, com per exemple, mostrar els missatges del fòrum d'un cas, es comprova si el professional hi té accés. S'aconsegueix fent una *query* a la base de dades obtenint el cas en qüestió fent un *join* de l'*id* del cas amb l'*id* del professional. Si ens retorna les dades del cas és que efectivament aquell professional està assignat al cas, i podem continuar fent altres tasques.

En cas contrari, s'informarà amb un missatge d'error i es retornarà a la pàgina prèvia.



Figura 43. Missatge d'accés denegat si s'intenta accedir a alguna funcionalitat o informació que no pertoca

De la mateixa manera, també es fa la gestió de funcionalitats. Si un cas és tancat, no es podran afegir nous participants, etc ...

3.8 Maquetació i disseny

3.8.1 Maquetació

En aquest apartat ens disposem a explicar les diferents parts estructurals que conformen la nostra web.

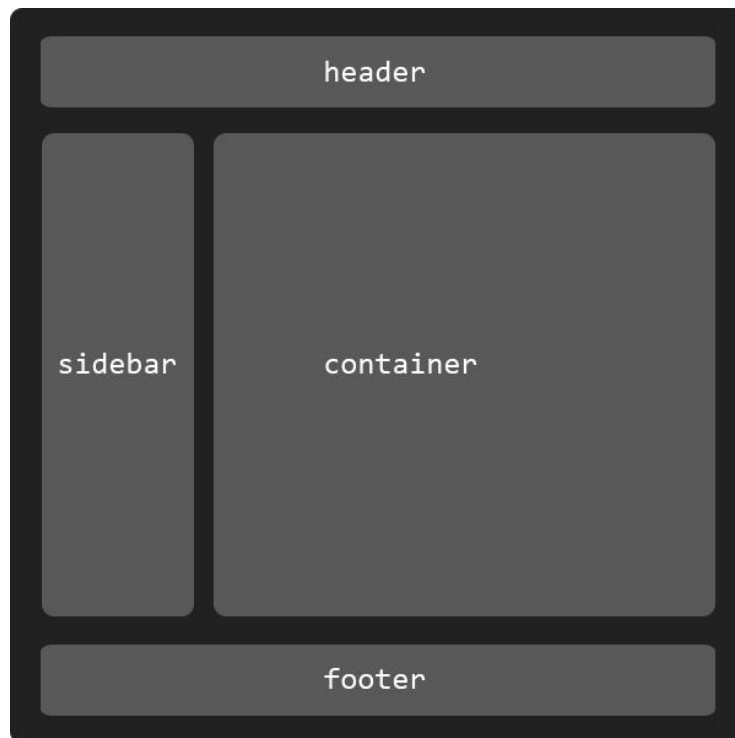


Figura 44. Mostra l'estructura més habitual que segueixen les pàgines de la web

Com es pot comprovar en la figura anterior, s'ha seguit l'esquema clàssic de disseny de pàgines web. A continuació aprofundim en cada una de les zones i expliquem els motius de fer-ho així i les variacions que es produeixen.

Un dels principals motius d'escollir aquest tipus de format és la usabilitat. L'usuari que es disposi a utilitzar-la acostuma a entendre aquest esquema simplement amb mirar-lo.

El **header o capçalera** contindrà el títol i serà el principal focus de navegació de la pàgina. Des d'ell ens podem moure entre els diferents grans blocs que formen la web.

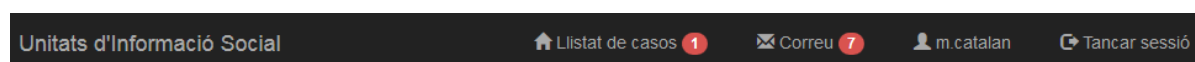


Figura 45. Mostra la barra de navegació principal

El llistat de casos és la pàgina inicial, l'índex. Si tenim algun comentari d'un cas no llegit, ens apareixerà una notificació per avisar-nos d'aquest fet. De la mateixa forma, el correu extern de l'aplicació també és accessible de d'aquí i també mostra el número de correus no llegits.

Les altres dues opcions corresponen al propi perfil de l'usuari i a tancar la sessió.

El **sidebar o barra lateral** contindrà el menú de cada apartat on ens trobem, així per exemple, si estem consultant el correu, la barra lateral serà la següent:

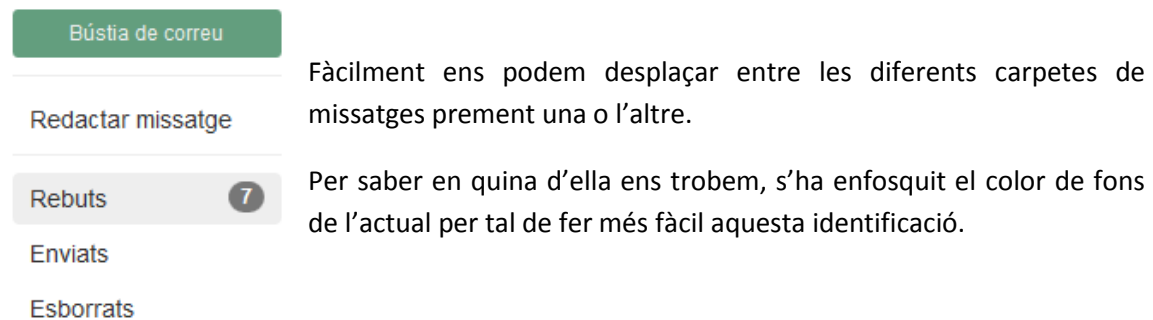


Figura 46. Mostra la barra lateral en el controlador dels correus.

El **container o contenidor principal** és qui conté tota la informació i opcions del lloc on ens trobem. És la part central de la pàgina web i on es produeixen la majoria d'interaccions amb l'usuari de l'aplicació.

El **footer o peu de pàgina** acostuma a contenir informació de contacte, sobre les institucions responsables, etc..

Ara bé, aquesta disposició no és estàtica. Com havíem comentat anteriorment, cada rol realitza unes funcionalitats, i visualitza un tipus d'informació o una altre. Això es superposa a l'estructuració bàsica de la web per tal d'afegir variacions. Com que les vistes realitzen *require* de diversos *templates*, no hi ha cap problema en modificar la disposició, sempre i quan tinguem en compte les amplades corresponents de cada bloc.

Es pot observar, per exemple, en els fòrums dels casos. Allà hi tenim també una segona barra lateral disposada a la part dreta del contenidor principal. Aquesta barra l'utilitzem per poder observar els participants presents en el cas i per filtrar comentaris en funció de l'organisme desitjat.

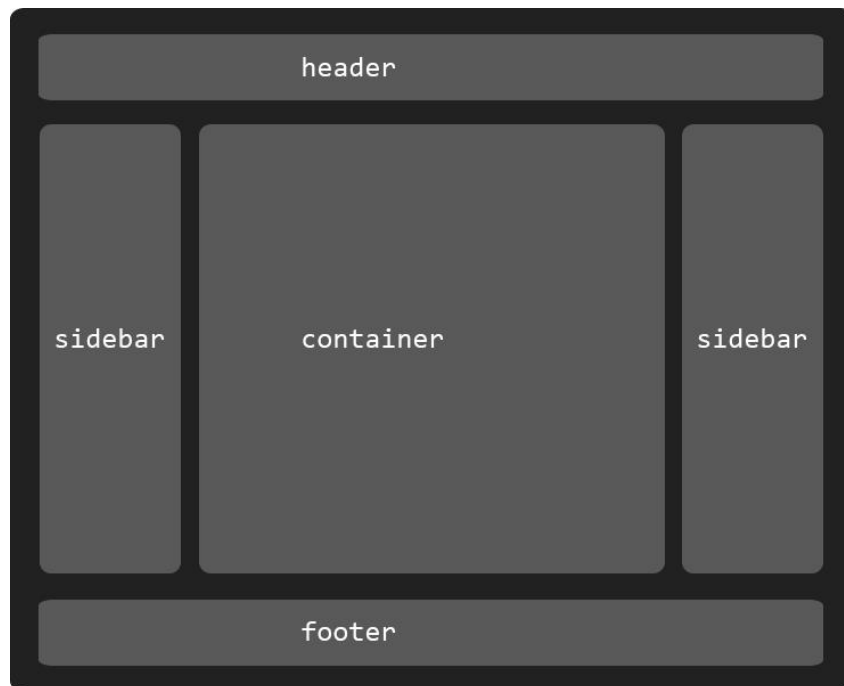


Figura 47. Mostra l'estructura de les pàgines que contenen 2 barres laterals

3.8.2 Disseny adaptatiu

Respecte al codi utilitzat a nivell de client (*html*, *css* i *javascript*), m'he ajudat d'un *framework* que permet crear interfícies web adaptant-les a tot tipus de dispositius. Era, a més, un dels requisits del projecte. S'utilitza **Bootstrap**. Però de què serveix exactament aquest *framework*? Quin tipus d'ajuda ens proporciona? Vegem-ho més clarament.

Suposem que volem crear una taula amb 2 columnes. Cada una d'aquestes 2 columnes disposarà d'un cert tipus d'informació. Ho més senzill seria crear una `<table>` i anar creant files amb `<tr>` i columnes amb `<td>`. Quin és el problema? En dispositius de pantalla gran, no n'hi ha cap. Però que succeirà quan un dispositiu mòbil intenti visualitzar aquesta taula? Doncs que no serà possible ja que l'amplada en píxels que formen les columnes és superior als píxels d'amplada de la pantalla.

Bootstrap ofereix una solució a aquesta i d'altres problemàtiques. Estableix 4 tipus de dispositius i els assigna un tipus de mida fixe:

- Dispositius molt petits (smartphones): menys de 768 px, prefix *col-xs*
- Dispositius petits (tablets): entre 768 i 992 px, prefix *col-sm*
- Dispositius mitjans (pantalles pc): entre 992 i 1200 px, prefix *col-md*
- Dispositius grans (pantalles pc allargades): Més de 1200 px, prefix *col-lg*

Cada un d'aquests grups té una sèrie de prefixos per tal de ser utilitzats en les classes dels elements *html*. Continuem amb l'exemple anterior per acabar d'entendre-ho.

Ara ja no es necessari crear cap `<table>`, sinó que ho podem fer amb `<div>` (contenidors) dotant-los d'una mida variable en funció del dispositiu que estigui accedint a la pàgina.

El codi resultant seria el següent:

```
<div class="taula">  
  <div class="fila1 col-lg-6 col-md-6 col-sm-6 col-xs-12"></div>  
  <div class="fila2 col-lg-6 col-md-6 col-sm-6 col-xs-12"></div>  
</div>
```

Bootstrap estableix que un 100% d'amplada correspon a 12 columnes, per tant *col-lg-12* seria el mateix que fer *width:100%* en *css*. En l'exemple estem dient que volem que cada columna ocupi la meitat de l'espai disponible (*col-lg-6*) en dispositius grans, mitjans i petits. Però perquè els dispositius molt petits tenen un *col-xs-12*? Doncs precisament per evitar el problema inicial. Quan les *media queries* detectin que es tracta d'un *smartphone* les columnes passaran d'ocupar la meitat de l'espai disponible a ocupar-lo tot. Els contenidors cauran un sota l'altre i ja no apareixeran al costat, impossibilitant així que la informació mostrada (en amplada) sigui més gran que la pantalla del dispositiu.

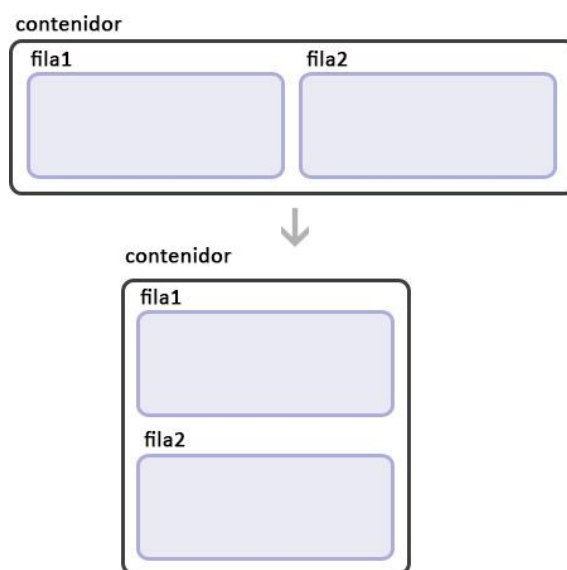


Figura 48. Caiguda de files (caixes) en el disseny adaptatiu realitzat

Aquesta és la tècnica que s'ha escollit per fer la web *responsive*. Jugant amb les classes proporcionades per *Bootstrap* es pot aconseguir que totes i cada una de les parts de la web sigui correctament visualitzada en totes les mides de pantalles disponibles. Bootstrap, a més, ofereix un munt de classes correctament estilitzades en *css*, i de les quals ens hem ajudat per fer que l'aparença sigui ho més maca possible.

3.9 Usabilitat

3.9.1 Criteris d'usabilitat

S'han seguit alguns criteris d'usabilitat a l'hora de dissenyar la interfície de la pàgina web. Bona part de les accions efectuades formen part les recomanacions realitzades a l'assignatura de Factors Humans i Computació. Anem a veure-les.

Un dels primers criteris en que es basa en la usabilitat, és en la facilitat d'aprendre i memorització de les diferents funcionalitats presents en la web. La facilitat d'aprenentatge es pot aconseguir gràcies a tècniques com les *affordances*. En aquesta aplicació, sovint es mostren icones que tendeixen a fer comprendre a l'usuari, la possible funcionalitat derivada de prémer o no un botó. S'intenta associar un objecte de la vida real a un tipus de funcionalitat. Podem observar alguns exemples.

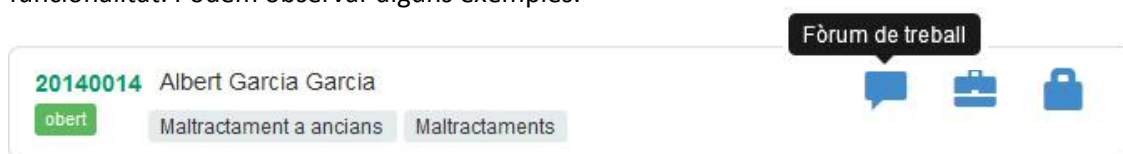


Figura 49. Cas present en la pàgina d'inici de l'aplicació

A la figura anterior podem observar com hi ha diverses *affordances*. Cada icono representa un tipus d'acció diferent. L'icono de missatge denota que en prémer-la s'anirà directament al Fòrum del cas (o almenys això s'ha intentat). Aquesta associació es reforça amb la inclusió de *tooltips* per tal de que l'usuari pugui estar convençut de que la acció que vol realitzar és la correcta. S'han inclòs, a més, elements que facin més fàcilment identificable un cert tipus de conjunt. Tornant a aquest exemple, les etiquetes de les situacions (*Maltractaments a ancians* i *Maltractaments*) poden fer més fàcilment identificable un cas que no pas el nom.

També millora l'aprendibilitat la inclusió de pistes a l'hora de realitzar una tasca. En podem veure en gairebé tots els camps d'entrada dels formularis, gràcies a l'atribut *placeholder* que permet explicar a l'usuari (o mostrar un exemple) que és exactament el que es demana en aquell camp. Els camps que poden resultar més confusos o complexos, disposen a més d'una petita explicació situada just sota seu.

No podem deixar d'explicar el feedback ni la visibilitat si parlem de camps de formularis.

Com pot un usuari adonar-se de que està omplint un camp de manera errònia? Si no s'ha fixat en els *placeholders* ni en les indicacions, se li haurà de mostrar un missatge, un **feedback** que li mostri en què s'està equivocant. Això s'aconsegueix mitjançant les comprovacions al costat del client. Com que anteriorment ja les hem explicat, passem ara a veure les conseqüències de la detecció d'un camp erroni

Nom *
Escriu el nom

Cognoms *
Escriu els cognoms

NIF *
478854586N
NIF / NIE incorrecte!
Escriu el NIF/NIE amb lletra i sense guió

Data de Naixement *
Ex: 23/08/1999
Escriu la data amb el format dd/mm/aaaa

Figura 50. Cas present en la pàgina d'inici de l'aplicació

S'hi pot veure com es mostra un *feedback* en color vermell informant de que el NIF introduït no és vàlid (la lletra no correspon). Per cada camp, s'ha intentat que es mostrés una petita indicació molt visual que denoti un error.

Com veiem en la figura 48, el feedback després de cada acció és bàsic. He establert quatre tipus de feedbacks: d'informació (color blau), èxit (verd), avís (taronja), error (vermell).

Altres accions tingudes en compte a l'hora de fer els formularis és la inclusió d'asteriscs en camps obligatoris, l'ús de **l'espai en blanc** per separar conjunts de camps i la **alineació** dels camps per mantenir una consistència estructural, aconseguint així, simplicitat en la visualització.

També les variables visuals ajuden a millor l'experiència de l'usuari. És força comú que durant l'aplicació trobar-se amb *labels* que ajuden a identificar característiques d'una manera ràpida (estats del cas, participants assignats o no..).

Respecte a la disminució d'errors, s'han col·locat passos previs de confirmació per tal de realitzar accions que es podrien considerar "més importants" que algunes d'altres. No és el mateix enviar un comentari que tancar un cas. En aquesta darrera situació haurem de fer passos previs i confirmacions. Això s'aconsegueix amb els anomenats *modals* de *Bootstrap*.

Gràcies a una llibreria externa, *bootbox*, es pot aconseguir que aquests missatges modals tinguin la formatació adequada. Es vol des assignar un professional d'un cas? Primer apareixerà una "caixeta" que preguntarà si realment és així. En cas de que s'hagués premut de forma errònia, es procedirà a la cancel·lació de l'acció.

En el marc de la usabilitat, s'han considerat diferents tipus de pautes més, com per exemple la utilització de pocs colors i no saturats dins d'una mateixa pàgina, la inclusió de sistemes de paginació, la pròpia estructura dels *templates* que permet sempre tornar a l'inici... Tota la pàgina, a més, és adaptable a qualsevol dispositiu mòbil (tot i que algunes de les funcionalitats s'han ocultat).

Per a la creació dels *templates*, no s'han seguit els procediments estrictes i incrementals com són l'*storyboard* ni el *prototipat*. La raó, és la pròpia envergadura del projecte, que en molts casos ha limitat la procedimentació. He hagut de prioritzar, i tot i així, n'hi ha parts que són fàcilment millorables (les explico més endavant).

3.9.2 Avaluació de la usabilitat

L'avaluació de la interfície gràfica per tal de millorar el disseny, s'ha realitzat mitjançant mètodes de testeig. S'han fet tests sumatius (o de valoració) per tal que l'usuari interactués amb la web fent tasques, recollint dades qualitatives i quantitatives. També els tests de validació i verificació han format part de l'avaluació.

Cal dir que aquestes observacions de tasques s'han realitzat la majoria de cops amb usuaris de diferents edats, i amb diferents facilitats per la tecnologia, que desconeixien totalment l'aplicació. Els motius han sigut 2. Si aquests usuaris eren capaços de completar certes tasques sense saber del tot que signifiquen, sembla lògic creure que un professional del sector que en conegui els termes, podrà també fer-ho. El segon motiu ha estat la dificultat d'establir avaluacions presencials amb els professionals reals que l'utilitzaran.

Moltes variacions presents en la interfície, han estat gràcies a errors que s'han anat veient que es cometien durant la realització de tasques, així com consells i feedback proporcionat.

3.10 Base de dades

En aquest apartat ens centrarem en explicar el disseny de la base de dades i a explorar les taules per tal de poder comprendre quin informació emmagatzema cada una d'elles. Com que

3.10.1 Disseny

Com que s'ha seguit un model evolutiu de desenvolupament de software, la creació de les taules necessàries es produïa en paral·lel un cop eren necessàries.

S'ha seguit l'esquema clàssic de model de bases de dades relacional: entitats diferenciades en taules i relacionades entre si mitjançant claus forànies. Anem ara a veure en més detall quines són les taules que s'han utilitzat i quin és la informació que emmagatzema cada una d'elles.

3.10.2 Taules

Primer de tot intentaré explicar les taules que tenen un grau de dependències més elevat, aquestes no formen part de cap subgrup però són presents en la majoria de subgrups. Dit d'una altra manera: són les que tenen més *foreign keys* apuntant cap a elles.

Els atributs subratllats de les taules indiquen que són una clau primària. Totes les *primary keys* i les *foreign keys* tenen el seu índex associat, per tant a partir d'ara, no s'enunciarà més aquests índexs en aquelles taules on siguin presents.

roles
value
name

Conté tots els rols d'usuari existents dins l'aplicació.

value: És la constant associada al rol.

name: És el nom del rol.

organitzacions
<u>organization_id</u>
name
acronym
label
role

Conté les dades dels organismes.

organization_id: És la clau primària.

name: És el nom real de l'organisme.

acronym: És l'acrònim de l'organisme.

identifier: És l'etiqueta que s'ha associat a cada organisme.

role: És la constant del rol al que pertany (no és una clau forània).

professionals

professional_id
name
surnames
identifier
role
email
registration_date
hash_password
unread_messages
unread_comments
organization_id

Conté les dades dels professionals que interactuen amb l'aplicació.

professional_id: És la clau primària.
name: És el nom real del professional.
surnames: Són els cognoms reals del professional.
identifier: És el "nom d'usuari".
role: És la constant del rol al que pertany (no és una clau forània).
email: És el correu electrònic real.
registration_date: És la data en que va ser donat d'alta a l'aplicació.
hash_password: És la contrasenya encriptada.
unread_messages: Són el número de missatges no llegits.
unread_comments: Són el número de comentaris no llegits.
organization_id: És la clau forana que apunta a un organisme.

Índexs: L'identificador, el nom, els cognoms i restricció d'identificador únic.

cases

case_id
date
state
identifier

Conté la informació bàsica i identificadora d'un cas.

case_id: És la clau primària.
date: És la data en la qual s'ha produït l'últim canvi d'estat del cas.
state: És l'estat del cas.
identifier: És l'identificador que cada cas té associat.

Índexs: L'identificador i restricció d'identificador únic.

Respecte a l'usuari dels casos

users

user_id
name
surnames
birthdate
sex
nif
address
phone
email
disability
case_id

Conté les dades de l'usuari en que es centra el cas.

user_id: És la clau primària.
name: És el nom de l'usuari.
surnames: Són els cognoms de l'usuari.
birthdate: És la data en que va néixer.
sex: És el gènere de l'usuari.
nif: És el *nif* / *nie* identificador de l'usuari.
address: És l'adreça on viu l'usuari.
phone: És el telèfon de l'usuari.
email: És el correu de l'usuari.
disability: Per saber si l'usuari és o no discapacitat.
case_id: És la clau forana que apunta al cas al que pertany.

Índexs: El *nif*, i restriccions de *nif* i id del cas únics.

representatives

representative_id
fullname
relation
address
email
phone
date
state
user_id

Conté les dades dels representants dels usuaris.

representative_id: És la clau primària.
fullname: És el nom i cognoms del representant.
relation: És la relació representant-usuari (pare, mare, àvia, tutor legal..).
address: És l'adreça física on viu el representant.
email: És el correu del representant.
phone: És el telèfon del representant.
date: És la data en la qual es realitza l'assignació representant-usuari.
state: És el *flag* que ens indica si el representant és l'actual o no.
user_id: És la clau forana que apunta a l'usuari del qual és o era representant.

Respecte als professionals i la seva assignació als casos

cases_professionals

case_professional_id
read_flag
unread_comments
case_id
professional_id

Conté les assignacions dels professionals als casos.

case_professional_id: És la clau primària.

read_flag: Determina si el professional encara no ha visitat el cas per 1r cop.

unread_comments: Conté el número de comentaris no llegits del fòrum.

case_id: És la clau forana que apunta al cas on està assignat.

professional_id: És la clau forana que apunta al professional assignat.

Índexs: Restricció de relació única cas-professional.

Disparadors: Duplicació de dades a la taula *cases_professionals_history* quan s'assigna o es desassigna un professional. També s'actualitza *unread_comments* de la taula *professionals* quan s'actualitza o s'esborra una assignació.

cases_professionals_history

case_professional_history_id
state
date
case_id
professional_id

Conté l'històric d'assignacions de professionals a un cas.

case_professional_history_id: És la clau primària.

state: És l'estat de l'assignació (assignat o des assignat).

date: És la data en la qual es va realitzar l'assignació o des assignació.

case_id: És la clau forana que apunta al cas.

professional_id: És la clau forana que apunta al professional.

Respecte als comentaris d'un cas

comments

comment_id
body
creation_date
read_flag
reading_date
professional_id_from
professional_id_to
case_id
thread_organization_id

Conté els comentaris del fòrum d'un cas.

comment_id: És la clau primària.

body: És el missatge, el comentari en si enviat.

creation_date: És la data en la qual es va enviar.

read_flag: Indica si el receptor del missatge l'ha llegit o no.

reading_date: És la data en la qual el comentari va ser llegit.

professional_id_from: És la clau forana que apunta a qui l'ha enviat.

professional_id_to: És la clau forana que apunta al professional destinatari.

case_id: És la clau forana que apunta al cas.

thread_organization_id: És la clau forana que apunta al fil de la conversa dels professionals (Exemple: Si un membre de SSB està parlant amb el CSMA, aquest id farà referència a l'organisme CSMA).

Disparadors: Incrementa i decrementa *unread_messages* en el destinari del comentari en funció de si el comentari es crea nou, s'esborra, canvia d'estat..

Respecte a l'espai d'assessorament d'un cas

questions

question_id
body
creation_date
read_flag
reading_date
answered
answer
answer_date
professional_id_from
professional_id_to
case_id

Conté les preguntes (i respostes) fetes en l'espai d'assessorament.

question_id: És la clau primària.

body: És el text de la pregunta.

creation_date: És la data en la qual es va enviar.

read_flag: Indica si l'assessor l'ha llegit o no.

reading_date: És la data en la qual la pregunta ha estat.

answered: És el flag que indica si la pregunta ha estat contestada.

answer: És el text de la resposta.

answer_date: És la data en que ha estat contestada.

professional_id_from: És la clau forana que apunta a qui l'ha enviat.

professional_id_to: És la clau forana que apunta a l'assessor destinatari.

case_id: És la clau forana que apunta al cas.

Respecte a la història social electrònica d'un cas

electronic_social_histories

esh_id
creation_date
health_indicators
mental_health_indicators
social_indicators
risk_indicators
case_id

Conté la informació relacionada amb la història social electrònica.

esh_id: És la clau primària.

creation_date: És la data en la qual va ser escrita.

health_indicators: Són els indicadors de salut.

mental_health_indicators: Són els indicadors de salut mental.

social_indicators: Són els indicadors socials.

risk_indicators: Són els indicadors de risc.

case_id: És la clau forana que apunta al cas.

Índexs: Restricció de *id* del cas únic.

notes

note_id
type
body
creation_date
professional_id
esh_id

Conté les notes de seguiment de cada professional de l'esh.

note_id: És la clau primària.

type: És el tipus de nota (de seguiment, informativa, etc.).

body: És el text escrit, el contingut de la nota.

creation_date: És la data en la qual va ser escrita.

professional_id: És la clau forana que apunta al professional que l'ha escrit.

esh_id: És la clau forana que apunta al cas.

Respecte a la història social electrònica d'un cas

electronic_social_histories

esh_id
creation_date
health_indicators
mental_health_indicators
social_indicators
risk_indicators
case_id

Conté la informació relacionada amb la història social electrònica.

esh_id: És la clau primària.

creation_date: És la data en la qual va ser escrita.

health_indicators: Són els indicadors de salut.

mental_health_indicators: Són els indicadors de salut mental.

social_indicators: Són els indicadors socials.

risk_indicators: Són els indicadors de risc.

case_id: És la clau forana que apunta al cas.

Índexs: Restricció de *id* del cas únic.

Respecte al *timeline* d'un cas

activities

activity_id
type
title
date
body
editable
editable_date
case_id
professional_id

Conté les activitats (o esdeveniments) del *timeline* d'un cas.

activity_id: És la clau primària.

type: És el tipus d'esdeveniment (de seguiment, informativa, de risc, etc..)

title: És el títol de l'esdeveniment.

date: És la data en la qual va ser creat l'esdeveniment.

body: És el contingut de l'esdeveniment.

editable: És el *flag* que determina si l'esdeveniment pot ser editat.

editable_date: Ens diu si la data de l'esdeveniment pot ser editada.

case_id: És la clau forana que apunta al cas.

professional_id: És la clau forana que apunta al professional que l'ha creat.

Respecte a l'arbre sociogenealògic d'un cas

nodes

node_id
name
surnames
specific_relation
relation
state
annotations
parent_id
case_id

Conté els nodes (persones) de l'arbre sociogenealògic.

node_id: És la clau primària.

name: És el nom de la persona.

surnames: Són els cognoms.

specific_relation: És la relació específica (familiar, amic..).

relation: És la relació entre l'usuari del cas i la persona de l'arbre (tiet, germà..)

state: Ens diu si el node és important o no.

annotations: Són les anotacions introduïdes que fan referència a la persona.

parent_id: És la clau forana que apunta al node pare.

case_id: És la clau forana que apunta al cas.

Respecte a les avaluacions d'un cas

evaluations

evaluation_id
reasons
comments
date
professional_id
case_id
case_state_id

Conté les avaluacions o motius de desestimació, obertura.. d'un cas.

evaluation_id: És la clau primària.

reasons: És el motiu en qüestió.

comments: Són comentaris associats a l'avaluació final del cas.

date: És la data en la qual s'ha inserit l'avaluació.

professional_id: És la clau forana que apunta al professional que l'ha creat.

case_id: És la clau forana que apunta al cas.

case_state_id: La clau forana que apunta a l'estat del cas associat a l'avaluació.

Disparadors: Actualitza l'estat de *cases* en cada inserció d'una avaluació.

rates

rate_id
rate
organization_id
evaluation_id

Són les puntuacions dels organismes, fetes en les avaluació final del cas.

rate_id: És la clau primària.

rate: És la puntuació (sense utilitat, parcialment útil o útil).

organization_id: És la clau forana que apunta a l'organisme.

evaluation_id: És la clau forana que apunta a l'avaluació final.

Respecte als estats dels casos

cases_states	
<u>case_state_id</u>	
state	
date	
case_id	
professional_id	

Conté l'historial d'estats que ha tingut un cas al llarg del temps.

case_state_id: És la clau primària.

state: És l'estat del cas (obert, tancat, desestimat..)

date: És la data on es va produir aquest canvi d'estat.

case_id: És la clau forana que apunta al cas.

professional_id: És la clau forana que apunta al professional que ho ha propiciat.

Respecte a les situacions

situations	
<u>situation_id</u>	
name	

Conté les dades de les situacions.

situation_id: És la clau primària.

name: És el nom de la situació.

cases_situations	
<u>case_situation_id</u>	
case_id	
situation_id	

Conté la informació relacionada amb la història social electrònica.

case_situation_id: És la clau primària.

case_id: És la clau forana que apunta al cas.

situation_id: És la clau forana que apunta a la situació.

Respecte a la missatgeria externa als casos

messages	
<u>message_id</u>	
header	
body	
creation_date	
professional_id_from	
professional_id_to	

Conté les dades dels missatges personals escrits entre professionals.

message_id: És la clau primària.

header: És la capçalera del missatge.

body: És el cos del missatge.

creation_date: És la data en la qual va ser enviat.

professional_id_from: És la clau forana del professional que envia el missatge.

professional_id_to: És la clau forana del professional que rep el missatge.

Disparadors: Incrementa *unread_messages* en el destinari del missatge.

messages_status	
<u>message_status_id</u>	
delete_flag	
delete_date	
read_flag	
reading_date	
message_id	
professional_id	

Conté l'estat d'un missatge respecte la interacció amb un professional.

message_status_id: És la clau primària.

delete_flag: És el *flag* que indica si el missatge està esborrat.

delete_date: És la data en la que es va esborrar el missatge.

read_flag: És el *flag* que indica si el missatge ha estat llegit.

reading_date: És la data en la que el missatge va ser llegit.

message_id: És la clau forana del missatge al que pertany.

professional_id: És la clau forana del professional.

Disparadors: En té dos que decrements en 1 *unread_messages* quan s'esborra un missatge no llegit o bé quan es llegeix.

contacts

contact_professional_id
date
contact_id
professional_id

Conté els contactes de cada professional en la missatgeria externa.

contact_professional_id: És la clau primària.

date: És la data en la qual el contacte va ser afegit.

contact_id: És l'id del professional que és un contacte de *professional_id*.

professional_id: És l'id del professional que té com a contacte *contact_id*.

messages_templates

message_template_id
header
body
fullname_from
professional_id_from

Conté les plantilles per a l'enviament de missatges automàtics.

message_template_id: És la clau primària.

header: És la capçalera del missatge.

body: És el cos del missatge.

fullname_from: És el nom complet del professional que l'envia.

professional_id_from: És la clau forana del professional que l'envia.

Respecte als recursos

categories

category_id
title
date

Conté les categories on s'agrupen els recursos.

category_id: És la clau primària.

title: És el títol de la categoria.

date: És la data en la qual va ser creada o modificada.

files

file_id
file_hash_name
file_name
file_size
file_type
file_extension

Conté les especificacions dels arxius pujats com a recursos.

file_id: És la clau primària.

file_hash_name: És el *hash* creat a partir de l'arxiu.

file_name: És el nom amb el qual l'arxiu va ser pujat al servidor.

file_size: És el nombre de bytes que conformen l'arxiu.

file_type: És el format de l'arxiu detectat.

file_extension: És el format real de l'arxiu.

resources

resource_id
title
url
date
category_id
file_id

Conté les dades dels recursos.

resource_id: És la clau primària.

title: És el títol o nom establert per identificar el recurs.

url: En el cas que sigui un enllaç, és la url.

date: És la data en la qual va ser creat.

category_id: És la clau forana de la categoria a la que pertany.

file_id: Si és un arxiu, és la clau forana que apunta a les seves dades.

resources_situations

resource_situation_id
resource_id
situation_id

Conté les assignacions de les situacions a cada recurs.

resource_situation_id: És la clau primària.

resource_id: És la clau forana del recurs.

situation_id: És la clau forana de la situació.

Índexs: Restricció única del parell de claus foranes.

organizations_resources
<u>organization_resource_id</u>
organization_id
resource_id

Conté les assignacions de les situacions a cada organisme.

organization_resource_id: És la clau primària.
organization_id: És la clau forana de l'organisme.
resource_id: És la clau forana del recurs.

Índexs: Restricció única del parell de claus foranes.

Respecte a les situacions que té associat cada professional

professionals_situations
<u>professional_situation_id</u>
professional_id
situation_id

Conté les situacions de cada professional de SSB.

professional_situation_id: És la clau primària.
professional_id: És la clau forana del professional.
situation_id: És la clau forana de la situació.

Respecte als professionals inactius

professionals_tmp
<u>professional_tmp_id</u>
identifier
email
activation_text
date
role
name
surnames
organization_id

Conté els professionals que s'han de donar d'alta a l'aplicació.

professional_tmp_id: És la clau primària.
identifier: És l'identificador del professional.
email: És el correu personal del professional.
activation_text: És el text d'activació utilitzat per poder donar-se d'alta.
date: És la data de creació del nou professional.
role: És el rol del professional.
name: És el nom del professional.
surnames: Són els cognoms del professional.
organization_id: És la clau forana de l'organisme al que pertany.

Índexs: Restricció d'identificador únic.

**Per a consultar d'una manera més exacte els atributs per tal de saber, per exemple, el tipus de camp (smallint, text, etc..) així com les diverses restriccions, cal observar el codi font sql.*

3.10.3 Querys importants

Durant l'obtenció de dades de la BD, s'han hagut de realitzar diverses *querys* que es consideren prou complexes com perquè sigui necessari explicar-les. Com s'ha explicat anteriorment, no s'ha seguit un sistema d'associació de model com a entitat (taula) de la BD, per tant, és comú que algunes *querys* obtinguin informació de taules que usualment són accedides des d'altres models.

A continuació les veiem:

- Obtenir els casos en l'índex de l'aplicació (per a professionals de SSB i d'organisme associat).

```
SELECT
    *,
    GROUP_CONCAT(s.name ORDER BY s.name ASC SEPARATOR ',') as situations
FROM (cases c)
JOIN cases_professionals cp ON (cp.case_id = c.case_id)
JOIN users u ON (u.case_id = c.case_id)
JOIN cases_situations cs ON (cs.case_id = c.case_id)
JOIN situations s ON (s.situation_id = cs.situation_id)
WHERE cp.professional_id = ?
GROUP BY c.case_id
ORDER BY cp.read_flag ASC, cp.unread_comments DESC, c.date DESC
LIMIT ?, ?
```

Anem a observar-la amb tranquil·litat. Com es pot veure, aquesta *query* determinarà quina informació es veu dels casos un cop s'inicia a l'aplicació, per això es tant important. S'obté un popurri d'informació ja que és necessari mostrar dades de diversos llocs, per tal de que l'usuari no hagi d'entrar dins del cas per comprovar que efectivament es tracta del que intenta cercar. Amb els diversos *joins* aconseguim relacionar la informació que després haurem d'agrupar per l'id del cas.

També destaca el *group_concat*. Bàsicament aquesta instrucció s'utilitza per concatenar amb un separador, tots els registres afectats per un *group by* en un camp de sortida. Estem, per tant, obtenim totes les situacions separades per comes. Després explotarem aquesta informació per obtenir-les individualment.

Finalment cal comentar l'ordre en que es mostren els casos. La prioritització escollida és la següent: primer els casos no llegits (nous), després els que tenen comentaris nous sense llegir i finalment per data (que s'actualitza en els canvis d'estat del cas). La instrucció *límit* ens serveix per crear la paginació quan es sobrepassen el número de casos permesos en cada pàgina.

- Realitzar la cerca avançada de casos per part de Serveis Socials Bàsics.

```
SELECT
    *,
    (SELECT MAX(ct.date) FROM cases_states ct WHERE ct.case_id = c.case_id AND
    ct.state = ?) as open_date,
    (SELECT MAX(ct.date) FROM cases_states ct WHERE ct.case_id = c.case_id AND
    ct.state = ?) as close_date,
    CONCAT(u.name, ' ', u.surnames) as fullname,
    GROUP_CONCAT(s.name ORDER BY s.name ASC SEPARATOR ',') as situations
FROM (cases c)
JOIN cases_professionals cp ON (cp.case_id = c.case_id)
JOIN users u ON (u.case_id = c.case_id)
JOIN cases_situations cs ON (cs.case_id = c.case_id)
JOIN situations s ON (s.situation_id = cs.situation_id)
```

D'aquesta *query* ens pot sorprendre que no s'estableix cap restricció. No hi ha cap *where* i per tant sempre s'obtindrien tots els casos, amb la conseqüent ineficiència que això suposaria. Però aquest fet té una explicació: aquesta consulta és parcial i es completa a partir dels camps introduïts per l'usuari; és una cerca que pot variar en funció del que es vulgui cercar. Com que fer una *query* per cada combinació de camps comportaria una quantitat de codi enorme i dificultaria l'escalabilitat i mantenibilitat, es va optar per construir-la de mica en mica:

```
if ( array_key_exists('case-open-first', $params)
    && array_key_exists('case-open-last', $params) )
{
    $havings[] = 'open_date BETWEEN ? AND ?';
    $first = $this->convertDate($params['case-open-first']) . ' 00:00:01';
    $last = $this->convertDate($params['case-open-last']) . ' 23:59:59';
    array_push($values, $first, $last);
    $types .= 'ss';
}
```

És complicat entendre el codi amb fragments que no mostren la seva totalitat, però l'objectiu d'aquesta explicació no es centra en fer entendible aquest tros, sinó en intentar comprendre la metodologia de creació de la *query*.

Per cada paràmetre s'intentarà comprovar si existeix. En cas afirmatiu s'aniran complementant *arrays* i *strings*. Un cop revisats tots els possibles paràmetres s'hauran d'anar creant els condicionals formant, així, una *query* completa. En l'exemple de codi següent es mostra com es crearien els *wheres* (creant una condició seguida d'*AND* per cada element present dins la llista).

```
$query .= ' WHERE ' . implode(' AND ', $wheres);
```

- Assignar el millor professional de Serveis Socials Bàsics en funció de les situacions i del número de casos que té assignats.

```

SELECT
  (
    (
      ( SELECT COUNT(*)
        FROM professionals_situations ps
        WHERE ps.professional_id = p.professional_id
          AND ps.situation_id IN (?)
      )
    /
      ( SELECT COUNT(*)
        FROM professionals_situations ps2
        WHERE ps2.professional_id = p.professional_id
      )
    ) * 100
  ) as percen,
  (
    SELECT COUNT(*)
    FROM cases_professionals cp
    WHERE cp.professional_id = p.professional_id
  ) as num_cases,
  *
FROM (professionals p)
JOIN organizations o ON (o.organization_id = p.organization_id)
WHERE p.role = ?
GROUP BY p.professional_id
ORDER BY percen DESC, num_cases ASC
LIMIT 1;

```

En aquesta *query* s'intentarà trobar el membre de SSB que més situacions té en comú amb les del nou cas que acaba de ser derivat. Però el número de situacions també és important.

Posem l'exemple d'un cas on s'han seleccionat 3 situacions: *Pobresa*, *maltractaments* i *aïllament social*. Si només miréssim per correspondència de situacions amb els professionals que també les tenen assignades, trobaríem massa resultats i poc rellevants. Com acotar la cerca? Obtenint el número total de situacions del professional i dividint el total de coincidències amb aquest número, obtenint així l'afinitat desitjada del professional amb el cas.

Com que és probable que aquest número sigui igual per diversos professionals, del que es tractarà ara és de repartir feina de manera equitativa, així que obtindrem el número d'assignacions a casos que té cada professional i ordenarem primer per afinitat de manera descendent (major afinitat) i per número de casos de manera ascendent (menys assignacions), obtenint, ara, el professional adequat.

3.11 Plugins i llibreries externes

Durant la programació de l'aplicació web, m'he trobat en situacions en les quals dubtava de programar per mi mateix una extensa part de codi, o bé cercar alguna alternativa *open source*.

El fet de cercar alternatives, ens allunya del fenomen "reinventar la roda". Estarem intentant trobar una solució a alguns problemes que segurament ja han estat resolts per algú, ja que en el seu moment, aquesta persona, es va trobar amb la mateixa problemàtica.

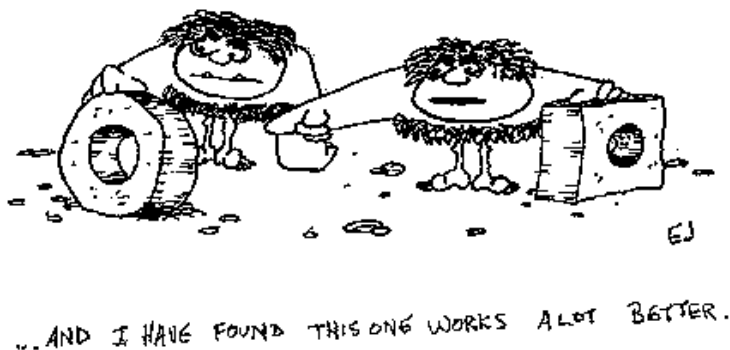


Figura 51. Expressa gràficament l'expressió "reinventar la roda".

Actualment la gran quantitat de material lliure (software) que es pot trobar a la xarxa, ens ajuda enormement a millorar i completar els nostres treballs. I així ha estat també en aquest cas. Una comunitat que comparteix coneixement, és una comunitat en constant evolució, i que és capaç d'assolir objectius impossibles de pensar si es treballa de manera individualitzada.

Sincerament, no hagués pogut programar tot el material que he utilitzat. *Bootstrap* per exemple, ajuda moltíssim en la maquetació i en l'estilització dels elements *html*. Aquestes hores evitades de programació en full d'estils han pogut ser dedicades a altres feines.

Això, per suposat, té contrapartides. El fet de no conèixer d'arrel en el que es treballa, converteix al programador (en aquest cas jo) en un engranatge que domina superficialment diverses parts de la producció. Centrant-nos en aquest treball final de grau, el volum de feina amb el que em vaig trobar des d'un inici va canviar les intencions que tenia a priori. Si bé jo volia conèixer de primera mà algun aspecte concret de la programació web, va quedar descartat en visualitzar les hores i hores de feina que venien per endavant.

Tot i així, cada cop més en el món professional es tendeix a modularitzar els components ja que no és possible programar-ho tot des de zero. La qual el desenvolupament de software evoluciona cada cop (i de manera positiva) a tenir l'habilitat d'incloure llibreries i adaptar-les al software propi.

Finalment, cal comentar que no s'ha utilitzat cap *framework* de model-vista-controlador però sí llibreries *javascript* i d'altres tipus. He volgut trobar un equilibri entre reinventar la roda i no reinventar-la, tot intentant passar per alt les coses que no m'atreien i centrant-me en un nivell

més baix en les que si que ho feien. Anem a veure de quins ajuts parlo (excloc *bootstrap* ja que ha estat explicat anteriorment).

3.11.1 TinyMce



Figura 52. Caixa d'opcions del plugin TinyMce

“ TinyMce és un editor de text per a HTML de codi obert que funciona en JavaScript i que es distribueix gratuïtament sota llicència LGPL. Té l'habilitat de convertir un camp del tipus textarea, o d'altres elements html, en instàncies de l'editor. ”

La definició que mostra la *wikipedia*, és doncs, molt clara. Un camp de text que permet formatar el contingut, agruparà informació i permetrà prioritzar-la, millorant així la comprensió lectora del destinatari del missatge.

Aquest *plugin* s'utilitza exclusivament en el correu extern de l'aplicació i funciona de la següent manera:

Primer de tot caldrà configurar els paràmetres d'inicialització. N'hi ha un munt, per destacar-ne uns quants, jo per exemple he fet que la caixa de text sigui *autoresize*, és a dir, que es vagi fent més gran cada cop que escrus, i he definit un *toolbar* amb les opcions de negreta, cursiva, subratllat, alienació i indentació del text i poder afegir enllaços.

A continuació, el professional escriurà i formatarà (si vol) el missatge i l'enviarà. El servidor rebrà el text amb les etiquetes *html* ja disposades i només caldrà realitzar algunes comprovacions tals com esborrar aquelles etiquetes no esperades. El pas final consisteix en inserir de forma rutinària aquestes dades a la taula corresponent.

3.11.2 Typeahead

És un *plugin* de *javascript* realitzat per *Twitter* per tal de poder realitzar un autocompletat. És utilitzat únicament quan s'intenta cercar el destinatari d'un missatge en el correu extern de l'aplicació.

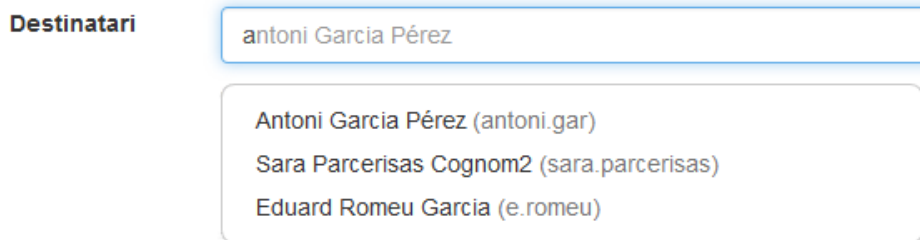


Figura 53. Autocompletat de contactes en afegir el destinatari del correu

Els destinataris formen part d'una llista de contactes que prèviament ha estat creada, i a l'hora de fer la cerca es fa únicament contra aquests. No tindria sentit (i seria molt costós computacionalment parlant) realitzar la cerca contra tots els professionals existents. Existint una llista limitada i personal, evitem que es pugui contactar amb qualsevol persona no coneguda.

Quan s'escriu una lletra, automàticament el plugin s'activa i es dirigeix de forma asíncrona a obtenir les dades del servidor. Es cerquen tant nom, cognoms com l'identificador dels professionals. En mostrar els resultats i seleccionar el destinatari, internament ens guardem l'id del professional escollit en un camp ocult.

3.11.3 JQuery Upload File

Ens permet pujar arxius al servidor mitjançant *ajax*.

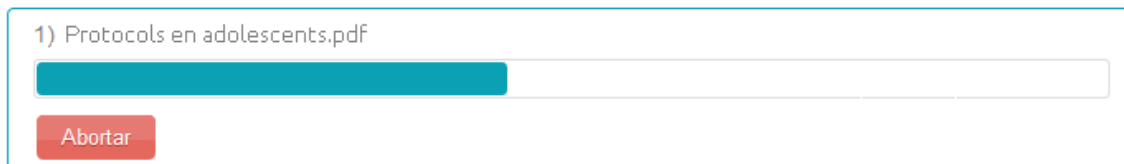


Figura 54. Barra de progrés del plugin *Jquery Upload File*

En prémer l'opció de pujar un nou arxiu, el mateix plugin ens obrirà una finestra per tal de que seleccionem el fitxer que volem pujar. Ell mateix s'encarrega de tallar-lo a trossos i pujar-lo de forma asíncrona. Mentre es produeix aquesta acció, una barra de progrés ens ho mostra de forma visual.

En finalitzar la pujada, un mètode prèviament definit del nostre controlador capturarà la informació de l'arxiu pujat en la *superglobal* de *php* `$_FILES`. Després de fer els *checks* oportuns, crearem un *hash* a partir de les dades de l'arxiu per tal de guardar-lo amb aquest nom. A la corresponent taula de la base de dades associarem tota aquesta informació en un únic registre per tal d'evitar que el fitxer quedi perdut.

Finalment, si tot ha succeït correctament, s'envia un feedback intern al client per a que mostri el missatge d'èxit, així com l'identificador del registre que s'acaba d'inserir.

En completar la resta del formulari i enviar-lo, totes les dades quedaran connectades en la nova inserció d'un registre a la taula de recursos.

3.11.4 Altres

També s'han fet servir altres llibreries menys complexes però no menys necessàries i útils. Per exemple: *datepicker* (mostra un desplegable del calendari per tal de seleccionar una data), *bootbox* (mostra un caixa modal de confirmació, abans de realitzar una acció que es considera habitualment perillosa o irreversible).

3.12 Planificació

3.12.1 Diagrama de Gantt

Anem a mostrar un diagrama de Gantt per tal d'expressar el temps que s'ha dedicat a la realització de les tasques principals al llarg de tot el projecte.

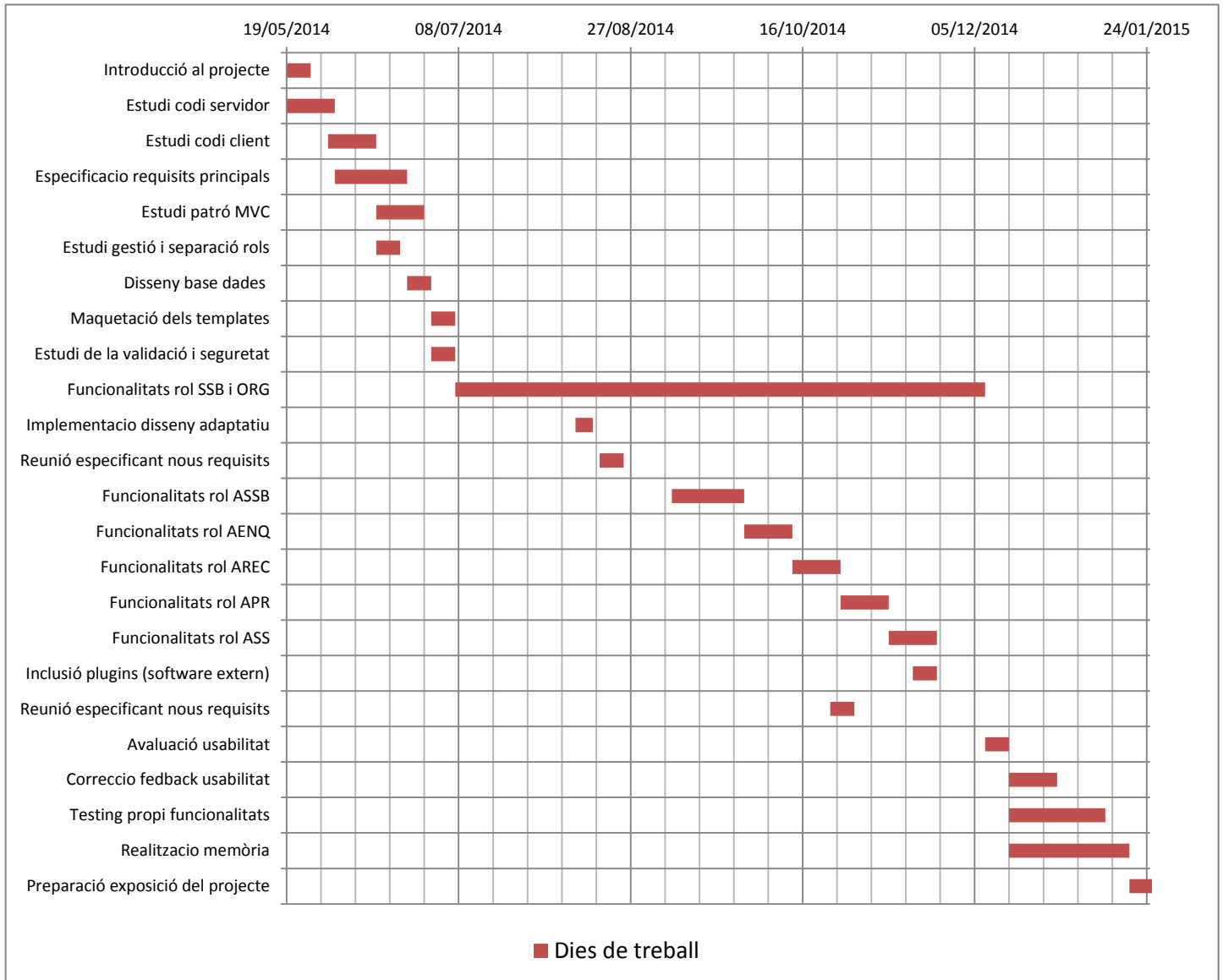


Figura 55. Diagrama de Gantt del projecte

A continuació es mostren les dades numèriques utilitzades per a la construcció del diagrama. S'ha decidit establir una duració mínima d'una tasca en un espai temporal corresponent a una setmana (7 dies).

Tasca	Data d'inici	Dies de treball	Data de finalització
Introducció al projecte	19/05/2014	7	25/05/2014
Estudi codi servidor	19/05/2014	14	01/06/2014
Estudi codi client	31/05/2014	14	15/06/2014
Especificació requisits principals	02/06/2014	21	22/06/2014
Estudi patró MVC	14/06/2014	14	29/06/2014
Estudi gestió i separació rols	14/06/2014	7	23/06/2014
Disseny base dades	23/06/2014	7	29/06/2014
Maquetació dels templates	30/06/2014	7	06/07/2014
Estudi de la validació i seguretat	30/06/2014	7	06/07/2014
Funcionalitats rol SSB i ORG	07/07/2014	154	07/12/2014
Implementació disseny adaptatiu	11/08/2014	5	14/09/2014
Reunió especificant nous requisits	18/08/2014	7	24/08/2014
Funcionalitats rol ASSB	08/09/2014	21	28/09/2014
Funcionalitats rol AENQ	29/09/2014	14	12/10/2014
Funcionalitats rol AREC	13/10/2014	14	26/10/2014
Funcionalitats rol APR	27/10/2014	14	09/11/2014
Funcionalitats rol ASS	10/11/2014	14	23/11/2014
Inclusió plugins (software extern)	17/11/2014	7	23/11/2014
Reunió especificant nous requisits	24/10/2014	7	30/10/2014
Avaluació usabilitat	08/12/2014	7	14/12/2014
Correcció feedback usabilitat	15/12/2014	14	28/12/2014
Testing propi funcionalitats	15/12/2014	28	11/01/2015
Realització memòria	15/12/2014	35	18/01/2015
Preparació exposició del projecte	19/01/2015	7	25/01/2015

3.12.2 Estimació dels costos de producció

Suposant que el següent treball hagués estat desenvolupat dins de l'àmbit corporatiu, podríem establir un petit esquema que ens permeti fer entendre els possibles costos estimats del projecte.

A les pròpies eines de desenvolupament, els hi establirem un cost de 0, ja que totes són programari lliure. Els factors externs tampoc són contemplats. Així per exemple, el fet que s'hagin d'utilitzar elements tangibles com ordinadors, material d'oficina, etc.. no queden reflectits en aquesta estimació.

He considerat que en una empresa real, aquest projecte hauria estat desenvolupat per diversos programadors i actors.

- Durada estimada del projecte: 6 mesos
- Número de programadors a jornada completa: 2.
- Sou dels programadors: 1500€ mensuals.
- Preu de les llicències del software utilitzat (open source): gratuït.
- Avaluació de la usabilitat durant una setmana: 350€.
- Desplaçament per les reunions en l'àmbit sociosanitari: 300€.

Fent el càlcul total podríem obtenir el següent cost:

<i>Sou programadors</i>	$1.500 * 2 * 6 = 12.000€$
	+
<i>Avaluació de la usabilitat</i>	350€
	+
<i>Desplaçaments</i>	300€
	=
	12.650€

3.13 Eines utilitzades



PHP és un llenguatge de programació interpretat que s'utilitza per a generar pàgines web de forma dinàmica. S'executa al cantó del servidor, per aquest motiu al navegador web ja l'hi arriba la pàgina en format HTML, no podent visualitzar-ne el codi.

L'aplicació ha estat desenvolupada amb PHP 5.3. Els motius d'haver escollit aquest llenguatge són els següents: és programari lliure, té una gran comunitat que hi dona suport, disposa d'un munt de llibreries, suporta una gran varietat de BD, té un bon rendiment i és un llenguatge senzill.



HTML 5 és la cinquena gran revisió del llenguatge bàsic de la World Wide Web. És una col·lecció estandarditzada pel desenvolupament i creació de pàgines web. Aquesta col·lecció representa la manera en que es presenta la informació en l'explorador d'internet i la manera d'interactuar amb ella.



Per tant, HTML5 ha de ser entès com un tot on hi és present els nou llenguatges d'*html*, javascript i css.



S'ha decidit utilitzar aquest llenguatge perquè, com acabem de comentar, és el més estandarditzat i amb possibilitats de futur.



MySQL és un sistema de gestió de bases de dades relacional multi-fil i multiusuari, que usa el llenguatge SQL. Els motius d'utilitzar-lo han estat els següents: és open source, té una gran escalabilitat i flexibilitat i un alt rendiment. El motor utilitzat ha estat *InnoDB* (suport transaccions ACID).



NetBeans

NetBeans és un entorn de desenvolupament integrat lliure i gratuït sense restriccions d'ús. Té molt d'èxit i una gran base d'usuaris, amb una comunitat en constant creixement

He utilitzat Netbeans IDE 8.0 Versió PHP.

4. Conclusions

L'atenció sociosanitària és primordial. Tota ajuda que serveixi per millorar aquesta atenció, és suficient per ser considerada la principal motivació que he tingut per desenvolupar aquest treball.

Respecte a conceptes purament informàtics, puc explicar que he après a realitzar pàgines web, que he entès el patró model – vista – controlador, m'he endinsat en les bases de dades relacionals, he utilitzat *frameworks* que mai havia tocat fins ara i he après molts i molts altres nous conceptes difícilment explicables.

Respecte a la valoració de l'àmbit sociosanitari que m'ha atès i assessorat, no puc estar més que agraït. Alguns dels principals inconvenients respecte a la creació de software, han estat l'especificació de requisits. Un cop avançada la realització d'un projecte, la seva exposició als canvis es torna complicada i difícilment tractable. Això, ha dificultat en alguns moments l'avanç del projecte tal i com jo esperava, però han estat únicament moments puntuals.

La realitat és que la informació proporcionada des d'un inici ha estat de molta utilitat. La inclusió d'una demostració amb pur *html*, em va permetre reflexionar i imaginar com havia de ser el producte final. Aquesta feina ha de ser valorada tant o més com la que he fet jo.

Per tant, puc expressar que la meva satisfacció per la realització d'aquest treball és la que en un inici tenia en ment. El fet d'haver realitzat un treball que es basa en la millora de la qualitat de vida de les persones, és el que m'anima a reconsiderar aspectes del meu futur professional.

5. Bibliografia

Llibres

Físics

José López Quijado. *Domine PHP y MySQL*. 2a Edición. Madrid: Editorial RA-MA, 2010.

Juan Diego Gauchat. *El Gran Libro de HTML5, CSS3 & Javascript*. 2a Edición. España: Editorial Marcombo, 2013.

En format digital

Eugenia Bahit @eugeniabahit. *POO y MVC en PHP*.

<http://www.bubok.es/libros/205199/POO-y-MVC-en-PHP>

Webs

<https://developers.google.com/web/fundamentals/>

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

<http://dev.mysql.com/doc/>

<http://librosweb.es>

<http://book.cakephp.org>

<http://stackoverflow.com/>

<http://www.w3c.es/>

I moltes i moltes més pàgines de gent anònima (i no) que m'han ofert coneixement de forma altruista.

6. Agraïments

- Als meus pares i germana per aguantar tots els meus maldecaps.
- Als meus amics i sobretot a en Joan, la Mònica i la Nina per realitzar les proves d'usabilitat.
- A en Marcos Catalán, per explicar-me el projecte amb tant de detall, per oferir-me recursos i per escoltar les meves inquietuds i constants preguntes.
- A en Sergio Escalera i en Marc Oliu per tutoritzar el projecte i respondre'm a totes les qüestions realitzades.
- A en Josep Moya i l'Ester Fornells per oferir-me el seu coneixement i experiència en el món social, i per intentar traslladar aquests coneixements a l'àmbit informàtic.
- A tots els programadors que han dedicat hores i hores de la seva vida per tal de facilitar-nos-la als demés.

ANNEX

En l'annex d'aquesta memòria ens centrem en explicar els passos a seguir per a realitzar diverses accions. Aquesta informació intenta que els professionals que vagin a utilitzar l'eina (l'aplicació web), puguin dur a terme les tasques que es proposin sense entrebancs.

Aquesta espècie de wiki, és informativa i està pensada a mode de recomanació. En cap cas es vol obligar a l'usuari final a seguir fil per randa tots els passos a seguir, sinó que el propòsit és simplement recomanador.

Índex de l'Annex

Casos

Què és un cas i com funciona?	1
Crear un nou cas	1
Fòrum de treball	3

Correu

Contactes	4
Enviar un correu	5

Casos

Què és un cas i com funciona?

Un cas és un sistema de centralització d'informació d'un usuari. La línia de la vida habitaul del cas és la següent: El cas és creat per un membre d'un organisme associat i derivat a Serveis Socials Bàsics. Aquest, revisa la informació proporcionada i ha de decidir si accepta tirar-lo endavant. En cas afirmatiu, el cas queda obert i es comença a realitzar la procedimentació per tal de resoldre'l.

Aquests procediments es basen fonamentalment en l'interacció amb un conjunt de professionals per tal d'esbrinar fets i situacions que ajudin a entendre la problemàtica a la que s'enfronta l'usuari.

El cas és tancat un cop Serveis Socials Bàsics ho considera apropiat.

Crear un nou cas

Els casos poden ser creats tant per membres de Serveis Socials Bàsics com per la resta de professionals. Anem a veure els passos que han de seguir uns i altres:

En prémer l'opció **"Afegir cas"** del menú esquerre de la pàgina, apareixerà una primera pàgina on s'han d'escriure els indicadors.

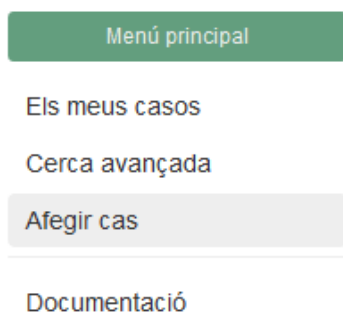


Figura A1. Mostra el menú principal de l'aplicació situat a la barra lateral esquerra amb l'opció d'Afegir cas premuda

No tots els indicadors poden ser escrits per tothom. En cada una de les capses es mostra qui és l'adequat per escriure'ls. Per tant, **no és necessari** escriure tots els indicadors. Aquesta informació, però, és rellevant, ja que és la primera que el membre de SSB observarà quan decideixi si accepta o no el cas.

La següent pàgina demana les dades de l'usuari/a. Els camps marcats amb un asterisc són obligatoris. El NIF ha de ser el real i ha de contenir la lletra seguint el format especificat. Si

l'usuari/a és menor d'edat o discapacitat s'hauran d'afegir, a més, les dades del representant actual.

Formulari per tal d'afegir les dades del nou usuari/a del cas. El formulari està organitzat en columnes i fileres. Les etiquetes dels camps són: Nom *, Cognoms *, NIF *, Data de Naixement *, Gènere *, Telèfon de l'usuari, Adreça, Email de l'usuari, i Altres. Els camps de text tenen exemples de dades: 'Escriu el nom', 'Escriu els cognoms', 'Ex: 45678235H', 'Ex: 23/08/1999', 'Ex: Carrer Tetuan 13, 5è 2a, 08013 Barcelona'. El camp de gènere té dos botons de selecció: 'Masculí' i 'Femení'. El camp 'Altres' té dos botons de selecció: 'Menor d'edat' i 'Discapacitat'.

Figura A2. Mostra el formulari per tal d'afegir les dades del nou usuari/a del cas.

A continuació, la següent pàgina demanarà que es seleccionin situacions. És important marcar-ne només les primordials. Aquelles en les que s'estigui segur o es tinguin indicis que poden ser presents en la problemàtica desenvolupada per l'usuari/a. Són aquestes situacions les que determinaran a quin membre de Serveis Socials Bàsics s'ha derivat el cas. De totes formes, tampoc és un fet irreversible. Més endavant poden ser modificades pels responsables adients.

Finalment ens trobaríem un panell amb el resum de les dades introduïdes. Si tot es considera correcte, caldrà prémer el botó **Crear cas**.

Si es volen modificar dades d'algun apartat, es pot fer prement els botons taronges presents a la part inferior de la pàgina. Cada un d'ell et traslladarà al lloc indicat. Un cop allà, heu de procedir a modificar ho que considereu oportú i prémer **Actualitzar**.

Barra de desplaçament entre els diferents apartats que conformen la creació d'un cas. La barra està organitzada en quatre botons taronges: 'Pas 1 - Indicadors', 'Pas 2 - Usuari', 'Pas 3 - Situacions' i 'Pas 4 - Confirmar'. El botó 'Pas 3 - Situacions' està destacat amb un color més fosc.

Si vols actualitzar alguna dada, prem el botó del pas en qüestió

Figura A3. Mostra la barra de desplaçament entre els diferents apartats que conformen la creació d'un cas.

Fòrum de treball del cas

El Fòrum de treball del cas és una eina de comunicació que utilitza Serveis Socials per interactuar i obtenir informació de la resta d'organismes que consideri necessaris.

Els membres de SSB poden interactuar amb tots els organismes que considerin necessaris. Per altra banda, els professionals d'altres organismes associats únicament poden interactuar amb membres de SSB. Això fa que la gestió del cas recaigui en mans d'un únic rol, acotant així la privacitat de la informació de l'usuari/a.

El primer pas per enviar un comentari és prémer el botó **Escriure comentari**, a continuació s'obrirà un formulari. En el primer camp s'haurà d'escollir el destinatari, que s'haurà de seleccionar de la llista desplegable.

A continuació s'haurà de procedir a escriure el comentari i finalment s'enviarà prement el botó **Enviar**.

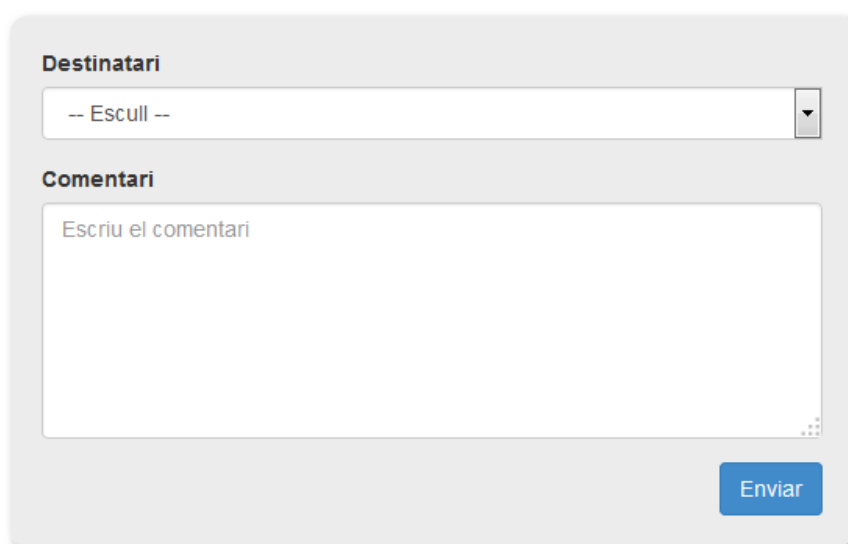


Figura A4. Mostra el formulari d'enviament de comentaris dins del fòrum d'un cas

Cal comentar que els fòrums són dependents dels casos. És una conversa centrada exclusivament en temes relacionats del propi cas. Si es vol realitzar un altre tipus de contacte extern als casos, l'opció correcta és utilitzar la missatgeria externa. A continuació la veiem.

Correu

La missatgeria externa de l'aplicació web funciona de manera independent dels casos. Ens ofereix la possibilitat d'establir un contacte amb un altre professional, sense que el motiu del missatge hagi d'estar forçosament relacionat amb la resolució d'un cas.

Veiem com fer possible aquesta comunicació.

Contactes

Abans de saber com enviar un missatge, ens hem de centrar en la llista de contactes. Aquesta llista ens determinarà **a qui** podem i a qui no podem enviar-ne.

Per tant, un cop hem premut l'enllaç que ens trasllada fins al correu, observarem una barra lateral a l'esquerra amb les diferents opcions que es poden realitzar.

Com que ara estem gestionant la llista de contactes, premem aquesta opció i observarem que la llista és buida si no en tenim cap.

Per afegir-ne un, haurem d'anar a **Afegir nou contacte**.

[+ Afegir nou contacte](#)

A continuació se'ns obrirà un formulari on haurem d'escriure l'identificador del professional que volem afegir. Com el seu nom indica, l'identificador **identifica** de manera unívoca. Per tant, si no es sap l'identificador de la persona que es vol afegir, no es podrà realitzar cap acció.

Així doncs, suposem que s'ha introduït un identificador vàlid i que el sistema ha trobat el professional en qüestió. Ens apareixerà un petit resum del contacte que volem afegir: el seu nom, cognoms i organisme al que pertany.



Figura A5. Mostra el contacte trobat un cop feta la cerca per identificador en afegir contactes.

En cas que efectivament es tracti de la persona a qui volíem cercar, procedirem a afegir-la a la nostra llista de contactes prement sobre l'enllaç de color blau. Un missatge d'èxit ha de confirmar que tot ha anat correctament.

Finalment, podrem comprovar, ara si, que a la llista hi apareix com a contacte a qui acabem d'afegir.

Enviar un correu

Si la llista de contactes és buida, si us plau, revisi la documentació anterior a aquest punt.

Quan intentem enviar un nou missatge, ens apareixerà el següent formulari:

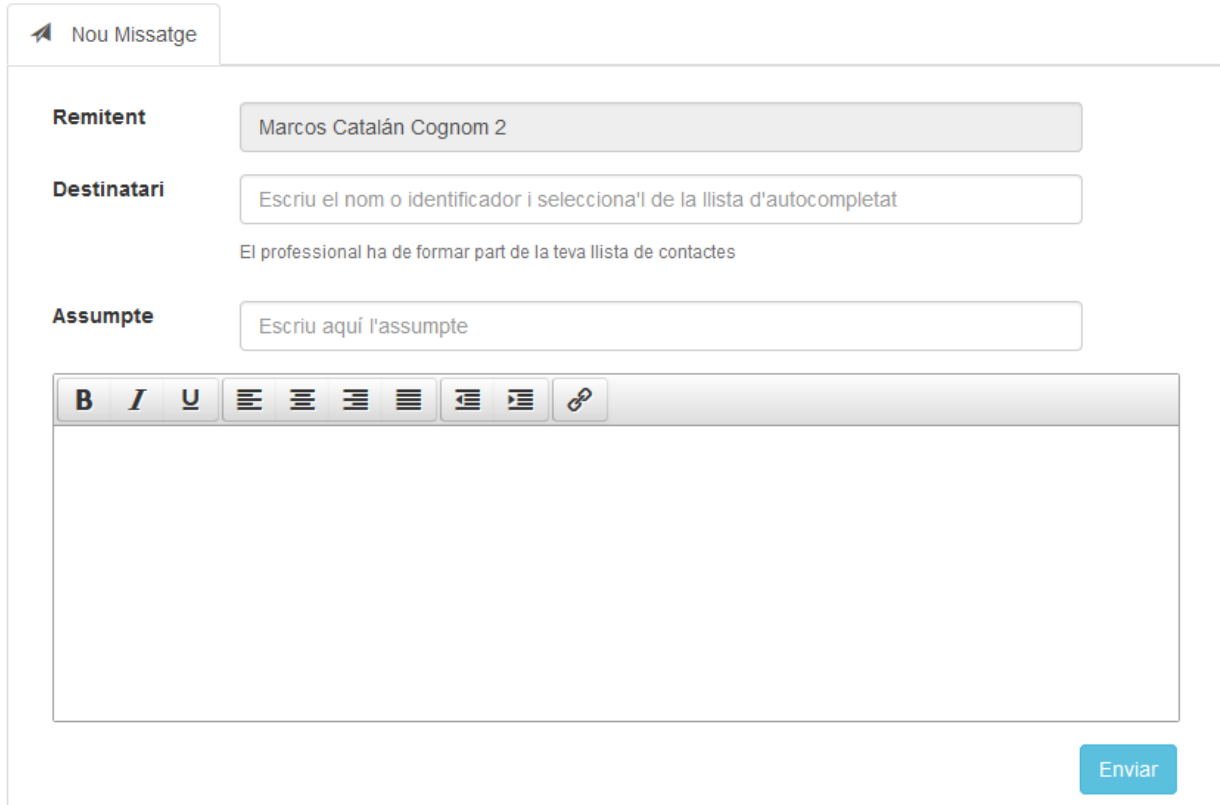


Figura A6. Mostra el contacte trobat un cop feta la cerca per identificador en afegir contactes.

En aquest formulari haurem d'omplir de forma obligatòria 3 camps: el destinatarí, l'assumpte i el cos del missatge.

El destinatarí és la part que podríem considerar que pot donar lloc a més equivocacions.

Per tal d'afegir el destinatarí, haurem de seleccionar-lo d'una llista desplegable que es crea mitjançant l'autocompletat. Aquest autocompletat es forma fent una cerca sobre els teus contactes.

Així doncs, un cop escrita la lletra del nom, cognom o identificador del destinatarí, s'obrirà una llista de possibles candidats, que s'anirà actualitzant cada cop que es premi una altra lletra. En seleccionar una de les persones, el destinatarí ja haurà estat escollit.

El camp de l'assumpte i del cos del missatges hauran de ser els següents en ser escrits.

En el cas del cos del missatge, pot ser formatat utilitzant les eines que es proporcionen. La funció de cada botó és la següent (s'anomenen amb l'ordre que apareixen en la figura anterior): Negreta, cursiva, subratllat, alineació del text cap a l'esquerra, alineació del text al

centre, alineació del text cap a la dreta, justificació del text, indentació cap a l'esquerra, indentació cap a la dreta i afegir un enllaç.

En prémer el botó d'**Enviar** es procedirà a realitzar l'enviament. Si tot s'ha produït correctament ha d'aparèixer un missatge de confirmació.