

Subtitle Converter for Video Editors

Víctor Hernández Torner

Treball Final del Grau de Multimèdia

Enginyeria Web

Consultor: Ignasi Lorente Puchades

Professor: Carlos Casado Martínez

12 de juny de 2017



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

Fitxa del treball final

Títol del treball:	<i>Subtitle Converter for Video Editors</i>
Nom de l'autor:	<i>Víctor Hernández Torner</i>
Nom del consultor/a:	<i>Ignasi Lorente Puchades</i>
Nom del PRA:	<i>Carlos Casado Martínez</i>
Data de lliurament (mm/aaaa):	<i>06/2017</i>
Titulació o programa:	<i>Grau de Multimèdia</i>
Àrea del Treball Final:	<i>Enginyeria Web</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Aplicació Web, Symfony, vídeo, subtítols</i>

Resum del Treball

Subtitle Converter for Video Editors és una aplicació web que té com a finalitat principal millorar la compatibilitat entre els programaris d'edició de vídeo professional i els fitxers de subtítols més comuns.

La idea d'aquest treball sorgeix de la meua experiència com a muntador/editor de vídeo professional en projectes audiovisuals i dels problemes que m'he trobat alhora d'importar i exportar arxius de subtítols comuns en els programaris d'edició de vídeo.

En aquest treball, s'ha aconseguit crear la versió beta de l'aplicació. Principalment, s'implementa la compatibilitat entre els diferents fitxers XML, que utilitza Final Cut Pro, programari d'edició de la companyia Apple, i el format de subtítols Subrip(.srt). L'aplicació ha estat dissenyada perquè en futures versions s'introdueixin nous formats d'exportació.

També, s'hi poden trobar tota la resta de funcionalitats que formaran part de la versió final, com són la gestió dels usuaris; l'edició de subtítols: modificar el text, definir l'estil i traduir-los; l'exportació a diferents formats de text; el destacat de subtítols en els arxius de text; la creació, edició i emmagatzematge dels paràmetres d'estil en fitxers XML i el canvi de la velocitat dels fotogrames.

Actualment, Subtitle Converter for Video Editors és un prototip d'aplicació que esperem que surti ben aviat al mercat i que es converteixi en una eina que faciliti la feina a tots els editors de vídeo del món.

Abstract

Subtitle Converter for Video Editors is a web application that its main purpose is to improve the compatibility between professional video editing software and the most common subtitle files.

The idea arises from my experience as an editor of professional audiovisual post-production projects and the problems that I found when I need to import or export common subtitle files into software edition.

In this work we have been able to realize the beta version of the application. Mainly, we have implemented the compatibility between the different XML files, that use Final Cut Pro, Apple's editing video software, and the subtitle format SubRip (.srt). The application has been designed so that in future versions we introduced new export formats.

Also, you can find other features that will be part of the final version, such as managing users; editing subtitles: change the text, styling and translate them; export to several text formats; highlighting subtitles in text files; creating, editing and storing style parameters in XML files and changing the frame rate.

In conclusion, we hope to publish soon Subtitle Converter for Video Editors in the market and it becomes a tool that helps to all video editors in the world.

Índex

Resum del Treball	ii
Abstract.....	iii
Índex	iv
1. Introducció	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball	2
1.3 Enfocament i mètode seguit.....	3
1.4 Planificació del Treball.....	4
1.4.1 Les activitats	4
1.4.2 La documentació.....	5
1.4.3 Les tasques.....	5
1.4.4 Pla de treball (Diagrama de Gantt)	6
1.4.5 Resum de les fites més rellevants	7
1.4.6 La retroalimentació.....	7
1.4.7 Anàlisi de riscos	8
1.5 Breu sumari de productes obtinguts.....	9
1.6 Breu descripció dels altres capítols de la memòria	10
2. Anàlisi i Disseny	12
2.1 Anàlisi de requisits	12
2.2 Diagrames UML.....	15
2.2.1 Casos d'ús	15
2.2.2 Diagrames d'activitats	23
2.2.3 El diagrama de classes	26
2.2.4 Interfície gràfica.....	27
2.2.5 Estructura de la base de dades	27
2.3 Arquitectura	29
2.3.1 Client/Servidor.....	29
2.3.2 Symfony	30
2.3.3 Base de dades	34
2.3.4 API/Llibreries de tercers	35
2.4 Estructura dels formats de subtítols	36
2.4.1 El format SubRip(.srt).....	37
2.4.2 Els formats de Final Cut Pro	39
2.5 Disseny gràfic i usabilitat	40
2.5.1 Perfil d'usuari	40
2.5.2 Història d'ús	41
2.5.3 Usabilitat (UX).....	42
2.5.3 Wireframes	45
2.5.4 Wall concept.....	51

3. Desenvolupament	52
3.1 El controlador.....	52
3.1.1 El Routing.....	52
3.1.2 Controladors, rutes i accions.....	54
3.1.3 Els formularis	56
3.2 La vista	59
3.2.1 Twig i les plantilles	60
3.2.2 L'ús d'Ajax.....	63
3.2.3 Programaris de tercers a la capa vista.....	65
3.3 El Model	66
3.3.1 Doctrine: la base de dades i les entitats	67
3.3.2 Estructura de classes del convertidor	70
3.3.3 El codi del convertidor	72
3.3.4 Mòduls i programari de tercers	77
3.3.5 Canvi de velocitat dels fotogrames	82
3.4 Seguretat	83
3.5 Tests	84
3.6 Versions de l'aplicació	86
3.7 Bugs.....	88
3.8 Instruccions d'instal·lació	88
4. Conclusions	93
5. Glossari	98
6. Bibliografia.....	103
7. Annexos	106
Annex 1 - Final Cut Pro Interchange Format (XML).....	106
Annex 2 - Final Cut Pro X Format (FCPXML).....	116
Annex 3 - Altres extractes de codi	127
Annex 4 - Programari i eines utilitzades.....	131

Llista de figures

Figura 1. Diagrama d'ús general.	15
Figura 2. Diagrama d'ús de l'edició de subtítols.	16
Figura 3. Diagrama d'ús de l'exportació d'arxius.	19
Figura 4. Diagrama d'activitats: exportar arxius de subtítols.	23
Figura 5. Diagrama d'activitats: exportar arxius de text.	24
Figura 6. Diagrama d'activitats: casos d'ús d'usuari i sistema.	25
Figura 7. Diagrama de classes.	26
Figura 8. L'interfície gràfica principal en una sola pàgina.	27
Figura 9. Taules de la base de dades.	28
Figura 10. Mapa de versions del programari Symfony.	31
Figura 11. Resum gràfic de les classes Request-Response a Symfony.	32
Figura 12. Wireframe: pàgina home.	45
Figura 13. Wireframe: sessió d'usuari.	45
Figura 14. Wireframe: edició de subtítols.	46
Figura 15. Wireframe: edició de subtítols amb el traductor.	47
Figura 16. Wireframe: pàgina d'exportació de subtítols.	48
Figura 17. Wireframe: finestres modals.	49
Figura 18. Wireframe: pàgina d'exportació de text.	49
Figura 19. Wireframe: pàgina de recuperació de la clau.	50
Figura 20. Informació sobre les vistes a la consola de Symfony.	62
Figura 21. Funcionament de Lightbox, plugin per JQuery.	65
Figura 22. Funcionament d'Spectrum, plugin per JQuery.	66

1. Introducció

1.1 Context i justificació del Treball

Subtitle converter for Video Editors(SCVE) és una aplicació web que té com a finalitat principal millorar la compatibilitat entre els programaris d'edició de vídeo professional i els arxius de subtítols més comuns. La idea d'aquest projecte sorgeix de la meua experiència com a muntador/editor professional de projectes audiovisuals.

Des de fa uns anys, molts editors de vídeo es trobem amb dificultats quan necessiten importar un arxiu de text amb subtítols a un programari d'edició de vídeo. Per diferents causes, la majoria de programaris professionals no són compatibles amb els arxius de subtítols més comuns.

Això, és un problema per molts motius però, sobretot, perquè moltes vegades les productores o empreses de serveis de post-producció demanen una còpia del projecte que s'està editant amb els subtítols incrustats en pantalla.

A causa de la problemàtica, molts muntadors es troben que si volen incrustar els subtítols en el vídeo que estan editant, l'única solució és tornar-los a escriure. Cosa que suposa una pèrdua d'esforç i temps considerable.

En l'actualitat, existeixen algunes aplicacions gratuïtes que permeten fer la conversió de formats. No obstant, presenten problemes diferents com la incompatibilitat amb les noves versions dels programaris, la impossibilitat de configurar els paràmetres dels arxius exportats o la insatisfacció dels resultats. Per aquestes raons, es pensa que la creació d'una aplicació web com SCVE és una oportunitat.

Com que en el mercat existeixen molts programaris i molts formats de subtítols diferents, per aquest treball, ens centrem en el programari d'edició Final Cut Pro de la companyia Apple, un dels més utilitzats en l'àmbit professional i semi-

professional, i amb el format de subtítols, Subrip(.srt), que és el més comú del mercat.

Final Cut només permet la importació de continguts textuais a través d'arxius d'etiquetatge .xml o .fcxml, segons la versió del programari en la que es treballa. Per tant, l'aplicació haurà de permetre la conversió en qualsevol dels sentits entre els tres tipus de format: .srt, .xml i .fcxml.

A més d'aquesta funcionalitat, el sistema ha d'oferir a l'usuari la possibilitat de definir tots aquells paràmetres que cada format permeti, editar i traduir el text de l'arxiu d'origen i exportar un llistat dels subtítols en un format text o full de càlcul.

La idea és que l'aplicació sigui escalable i oberta a tercers perquè en un futur es puguin afegir o actualitzar els nous formats que van apareixent.

1.2 Objectius del Treball

- Objectius principals del treball
 - Creació d'una aplicació web que permeti millorar la compatibilitat entre els programaris d'edició de vídeo professionals i els arxius de subtítols més comuns.
 - Elaborar i entregar la memòria, el codi font, una presentació en format lliure i una presentació en format vídeo del treball.
- Objectius Secundaris
 - Posar en pràctica els coneixements adquirits durant el Grau de Multimèdia. Sobretot, aquells que tenen relació amb els itineraris

d'Enginyeria Web i de Desenvolupament d'Aplicacions Multimèdia.

- Aprendre a planificar, estructurar i realitzar un projecte multimèdia des de zero seleccionant els procediments més adequats en cada moment del procés.
- Aprendre, experimentar i treballar amb noves tecnologies que no hem vist al llarg del Grau, però que estan relacionades amb els estudis i ens poden ser útils en el món laboral. Com per exemple, algun entorn de treball en PHP que implementi el patró MVC.
- Crear una interfície gràfica atractiva i usable que permeti una experiència gratificant a l'usuari.

1.3 Enfocament i mètode seguit

L'enginyeria de programari és una disciplina on hi participen molts factors i que en general presenta un grau d'incertesa elevat, per tant, qualsevol petit error o canvi en una de les fases del projecte pot afectar a la resta.

La metodologia de treball es basa en la classificació que fa Wysocki dels tipus de projecte. L'aplicació té un objectiu clar, tal i com s'explica en els punts anteriors (1.1 i 1.2), i una solució parcialment definida, atès que es treballen alguns aspectes de la nostra arquitectura amb una tecnologia que encara no coneixem. Ara bé, tot i ser desconeguda, es pot intuir per on aniran els trets.

També, s'ha de tenir en compte, que encara que estem davant d'un projecte curt i de poc risc, la nostra experiència en l'enginyeria de programari és pràcticament nul·la. Per tant, es necessita utilitzar un mètode que sigui senzill d'aplicar però que alhora permeti un petit marge per a la rectificació de possibles errors.

Per totes aquestes raons, s'opta per una metodologia clàssica o en cascada que es divideix en 4 fases: requisits, anàlisi i disseny, implementació i proves. La divisió en fases permet aprendre quins són els conceptes i les fites que es treballen a cadascuna d'elles, alhora que ajuda a adaptar el treball a el calendari d'entregues proposat per l'assignatura.

No obstant, per a millorar la tolerància als canvis i l'adaptabilitat, s'estableixen cicles de revisió a l'inici de cada fase, coincidint amb les correccions que el consultor fa de les activitats, que permeten revisar la feina que s'ha fet fins aquell moment i incorporar les correccions pertinents.

A més, no es descarta la possibilitat d'aplicar una metodologia iterativa i incremental en les fases d'arquitectura i implementació per tal d'acoblar les solucions tecnològiques.

1.4 Planificació del Treball

1.4.1 Les activitats

El treball es divideix en cinc activitats que tenen un objectiu concret i una data d'entrega assignada prèviament pels responsables de l'assignatura. Les quatre primeres ajuden a desenvolupar i acabar el treball i, en la cinquena, es defensa el treball davant dels tutors, professors i companys d'assignatura.

Activitat	Objectiu	Data d'entrega
PAC 1	Definició del treball	9/03/17
PAC 2	Anàlisi i disseny del treball	5/04/17
PAC 3	Implementació del treball	14/05/17
Lliurament Final	Entrega del treball acabat	12/06/17
Defensa virtual	Defensar el projecte	20/06/17

1.4.2 La documentació

Una de les tasques transversals més importants del treball i que s'inclou dins del diagrama de Gantt és la documentació. Aquesta tasca consisteix en registrar els resultats que s'han obtingut durant les diferents fases del treball. El resultat d'aquesta tasca és la memòria, document que s'entrega a l'acabar el projecte.

1.4.3 Les tasques

- PAC1 – Definició del treball
 - Definir l'abast del projecte
 - Elaborar del pla de treball
- PAC2 – Anàlisi i disseny de l'aplicació
 - Anàlisi dels requisits
 - Estudi de l'estructura interna i dels paràmetres que admeten els diferents formats de subtítols.
 - Estudi de les diferents codificacions de caràcters i la seva problemàtica.
 - Definir els requisits funcionals i no funcionals.
 - Diagrames UML
 - Casos d'ús
 - Diagrames de classe i altres
 - Estructura de la base de dades
 - Concepció gràfica de l'aplicació i usabilitat
 - Perfil dels usuaris i històries d'ús
 - Arbre de continguts i wireframes de baix nivell
 - Wireframes d'alt nivell: prototip gràfic
 - Arquitectura de l'aplicació
 - Definir la tecnologia i estructura de la capa client i servidor

- Definir la tecnologia i estructura de la base de dades
 - Ús d'Api's i llibreries de tercers
- PAC3 – Implementació de l'aplicació
 - Programació de la base de dades
 - Programació de l'esquelet de l'aplicació
 - Implementar Api de Google Translate
 - Test de la versió beta
 - Correccions i/millores
 - Crear el vídeo del funcionament de l'aplicació
- PAC Final – Entrega del treball
 - Programació dinamisme web
 - Test de la versió 1.0
 - Allotjar l'aplicació al servidor
 - Retocar la memòria
 - Creació de les presentacions
- Defensa pública del projecte
 - Revisar documentació i presentacions
 - Debat virtual

1.4.4 Pla de treball (Diagrama de Gantt)

El pla de treball es troba a la capeta Fitxers Adjunts de la Memòria amb el nom de PAC_FINAL_CALENDARI_Hernandez_Victor.png.

1.4.5 Resum de les fites més rellevants

Fita	Data d'entrega
PAC1	
Document Pla de treball acabat	07/03/17
Lliurament de la PAC1	09/07/17
PAC2	
Requisits documentats	14/03/17
Casos d'ús i diagrames UML documentats	20/03/17
Prototip gràfic de l'aplicació	27/03/17
Arquitectura documentada	04/04/17
Lliurament de la PAC2	05/04/17
PAC3	
Versió beta de l'aplicació acabada	03/05/17
Lliurament de la PAC3	14/05/17
LLIURAMENT FINAL	
Versió 1.0 de l'aplicació acabada	22/05/17
Lliurament Final (memòria i/o presentacions)	12/06/17
DEFENSA VIRTUAL	
Penjar presentacions al fòrum	15/06/17

1.4.6 La retroalimentació

La retroalimentació és una tasca que es realitza després de rebre la valoració que el tutor fa de cada activitat presentada. Aquesta consisteix en la revisió de tota la feina feta fins aquell moment a partir dels comentaris rebuts pel tutor.

A nivell pràctic això suposa que s'han d'anar actualitzant els documents principals del projecte com són el pla de treball, els requisits, el diagrames UML, el prototip gràfic, l'arquitectura o el codi font.

Per altra banda, durant la creació del treball també és molt important comentar amb el tutor les fites que l'alumne assoleix. D'aquesta manera els possibles errors es detecten abans que es facin més grans.

1.4.7 Anàlisis de riscos

Es troben dos riscos que tenen un nivell de severitat alt i que poden provocar una retard en les entregues.

- **RISC – 01**

- **Descripció:** haver fet una planificació del temps inadequada en les tasques de la fase d'anàlisi i disseny.
- **Probabilitat:** *ALTA*. A causa de la manca d'experiència en el disseny de programari és molt probable que hi hagi alguna subestimació o sobreestimació del temps necessari per realitzar les tasques
- **Impacte:** *ALT*. Retardar tota la planificació i afectar a les entregues parcials i final del treball.
- **Acció de mitigació:**
 - Contrastar amb el consultor la planificació proposada per la fase d'anàlisi i disseny.
 - Moure la data d'entrega de la PAC2 o intentar encabir tasques de disseny en la PAC3.
 - Treballar els caps de setmana per reajustar la falta de temps.

- **RISC – 02**

- **Descripció:** tenir problemes en la fase de disseny o en la d'implementació a causa de l'ús d'una tecnologia (entorn de treball en PHP) que ens és desconeguda.
- **Probabilitat.** *MITJA*. Tot i tenir certs coneixements, la manca d'experiència en aquesta tecnologia i el poc temps de què es disposa eleven la probabilitat.
- **Impacte:** *ALT*. Canviar d'arquitectura a mig projecte. L'impacte seria més alt si succeeix en la fase d'implementació que en la de disseny.
- **Acció de mitigació:**
 - Fer les proves d'implementació de l'entorn de treball que siguin necessàries en la fase de disseny.
 - Tenir una solució alternativa pensada amb la què ens sentim segurs.

1.5 Breu sumari de productes obtinguts

L'entrega de la documentació s'estructurarà en diferents carpetes dins de la carpeta principal de nom, PAC_FINAL_HERNANDEZ_VICTOR.

- La carpeta **Memòria** inclou:
 - La memòria del treball a l'arxiu ***PAC_FINAL_mem_Hernandez_Victor.pdf***
 - La carpeta ***Fitxers adjunts***, on s'inclouen fitxers complementaris a la memòria.

- La carpeta ***Fitxers de treball***, on s'inclouen fitxers originals que s'ha utilitzat per la realització del treball.
- La carpeta ***Captures de pantalla***, on s'inclouen captures de pantalla que mostren el funcionament de l'aplicació.
- La carpeta **Codi** inclou:
 - La carpeta ***SubtitleConverter_beta*** on s'inclouen tots els fitxers que pertanyen al codi font de l'aplicació.
 - La base de dades al fitxer ***subtitleConverter.sql***
 - La carpeta ***Fitxers de prova***, on es troben els fitxers de subtítols que ens permetran provar el funcionament de l'aplicació.
 - Les instruccions d'instal·lació a ***instruccions_install.txt***.
- La carpeta **Presentació** inclou:
 - La presentació escrita-visual del projecte a l'arxiu ***PAC_FINAL_presentació_Hernandez_Torner_Victor.pdf***
 - La presentació en vídeo del projecte a l'arxiu ***PAC_FINAL_video_Hernandez_Torner_Victor.mp4***
- L'autoinforme d'avaluació de les competències transversals a l'arxiu ***Informe_autoavaluació.pdf***.

1.6 Breu descripció dels altres capítols de la memòria

- **Capítol 2. Anàlisi i disseny:** s'explica tot el referent a la fase d'anàlisi i disseny del projecte. Aprofundim en l'anàlisi de requisits, el model de

classes, els diagrames UML, la base de dades, els prototips gràfics, la usabilitat i l'arquitectura.

- **Capítol 3. Desenvolupament:** es destaquen els aspectes més rellevants que s'han produït durant la implementació del codi.
- **Capítol 4. Conclusions:** s'explica quins aprenentatges s'ha adquirit durant el procés i de quina manera s'han transformat o no les idees inicials.
- **Capítol 5. Glossari:** definició dels termes i acrònims més rellevants utilitzats a la memòria.
- **Capítol 6. Bibliografia:** llista de les referències bibliogràfiques consultades per la realització de la memòria.
- **Capítol 7. Annexos:** Llistat d'apartats que són massa extensos per incloure dins la memòria i tenen un caràcter individual .

2. Anàlisi i Disseny

2.1 Anàlisi de requisits

En la definició dels requisits de l'aplicació han col·laborat quatre *stakeholders*. Tots ells vinculats a l'edició de vídeo professional però amb un perfil diferent. Hem pensat que un *stakeholders* que no tingui cap vinculació professional amb l'audiovisual no ens pot ajudar a definir els nostres requisits.

- **Emamuelle Tiziani:** Càmera i muntador professional. Habitualment treballava amb Final Cut, però des de fa poc s'ha passat a Adobe Première.
- **Àram Garriga:** Productor, muntador i etalonador. L'Àram té un perfil multitasca. És expert en crear màsters finals per emissions broadcast o cinema(DCP). Per tant, està molt acostumat a barallar-se amb tot tipus d'arxius multimèdia.
- **Víctor Hernández:** Productor i muntador. Té una llarga experiència com a muntador d'imatge i és expert en els sistemes d'edició Avid Media Composer i Final Cut Pro7. No li agrada la nova versió d'Apple, Final Cut Pro X.
- **Germán Roda:** Muntador de vídeo. Expert en la última versió de Final Cut Pro X.

Tots els requisits s'han acabat de definir a partir de les converses amb aquests *stakeholders*.

- Permetre a l'usuari pujar arxius de subtítols amb els formats següents:
 - Format Subrip: .srt

- Format .xml compatible amb les versions de programari 4, 5, 6 i 7 de Final Cut Pro.
 - Format .fcxml compatible amb les versions de programari de Final Cut Pro X.
- Un cop pujat un arxiu de subtítols, el sistema haurà de mostrar una taula amb el llistat de totes les línies de subtítols que hi ha a l'arxiu. Aquesta taula tindrà 4 columnes on es mostrarà la informació següent de cada subtítol:
 - Número del subtítol dins de l'arxiu(començant per 1)
 - Codi de temps d'entrada
 - Codi de temps de sortida
 - Text del subtítol(aquest text no pot tenir més de dues línies).
- L'usuari podrà editar el contingut textual cada línia de subtítol. Podrà fer les accions següents:
 - Podrà reescriure el contingut.
 - Podrà definir l'estil de la tipografia del subtítol com a regular o cursiva.
 - Podrà demanar al sistema que tradueixi de manera automàtica els subtítols a diferents idiomes.
- El sistema permetrà que els usuaris puguin registrar-se a l'aplicació i crear un compte. Tant els usuaris registrats com els no registrats podran utilitzar l'aplicació. La diferència serà que els usuaris registrats podran guardar i carregar els paràmetres dels arxius d'exportació que ells mateixos han creat.
- L'usuari podrà triar en quin format vol exportar els subtítols(els mateixos que els formats d'importació): .srt, .xml(versió 1) i .fcxml(v. 1.1 o 1.3).

- En el cas de triar una exportació amb .xml o .fcxml, l'usuari podrà triar entre pujar un arxiu .xml o .fcxml de referència creat per ell mateix a partir d'una seqüència de Final Cut o que el sistema en generi un de nou.
 - En cas que triï que el sistema en generi un de nou, haurà de definir els paràmetres de vídeo de la seqüència(resolució i fotogrames per segon).
- Un cop triat el format d'exportació, el sistema mostrarà a l'usuari els paràmetres que aquest pot definir. Els paràmetres seran diferents segons el format escollit. El sistema mostrarà uns paràmetres preestablerts que podran ser redefinits per l'usuari. Si l'usuari està registrat podrà desar i carregar els seus propis paràmetres.
- L'usuari també ha de poder exportar els subtítols en un format de text o full de càlcul com el .pdf, .csv, .xls o .odt(escollirem els més adients). En aquest casos, podrà ressaltar les línies de subtítols que desitgi. El tipus de ressaltat dependrà de les possibilitats que permeti cada format.
- L'usuari ha de poder descarregar els nous arxius al seu ordinador

2.2 Diagrames UML

2.2.1 Casos d'ús

En la nostra aplicació, l'actor principal sempre és l'usuari. A continuació, es mostra un diagrama de casos d'ús que simplifica els tres casos d'ús principals: l'inici de sessió(login), l'edició de subtítols i l'exportació tant a un arxiu de subtítols com a arxiu de text.

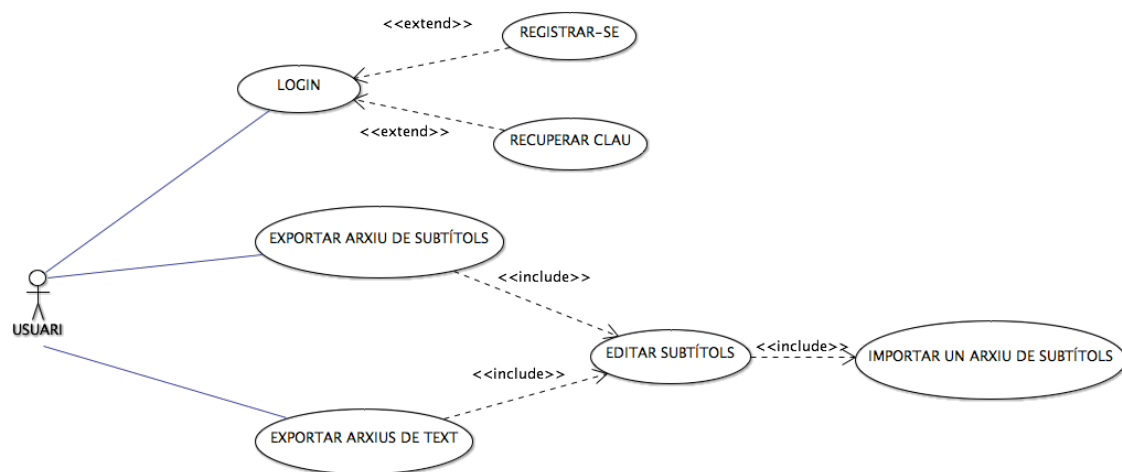


Figura 1. Diagrama d'ús general.

Per no estendre'ns massa, deixarem de banda els casos d'ús de nivell tasca més habituals com l'inici de sessió, registrar-se o recuperar la clau i ens centrarem en el cas importar un arxiu de subtítols. Aquest cas d'ús, inclou dos altres casos: per una banda, penjar un arxiu de subtítols i, per l'altra, seleccionar la codificació textual. En el segon cas, la idea és que el sistema sigui capaç de trobar la codificació de l'arxiu, sinó és possible, i tenint en compte que estem creant una aplicació professional, demanarem la codificació dels caràcters a l'usuari.

CAS D'ÚS: Importar un arxiu de subtítols

Actor principal: Usuari.

Nivell: Tasca.

Àmbit: Sistema.

Escenari principal d'èxit:

1. El sistema mostra la pantalla d'importació
2. L'usuari penja un arxiu de subtítols
3. L'usuari selecciona la codificació de l'arxiu
4. El sistema comprova que l'extensió de l'arxiu, llegeix l'arxiu i recupera el seu contingut.

Extensions:

- 4b. El sistema troba un error en la comprovació
- 4b.1 Es mostra un missatge d'error específic i tornem al punt 1

Els casos editar subtítols, exportar arxiu de subtítols i exportar arxiu de text són casos molt generals que inclouen i estenen altres casos d'ús més concrets. Per fer-ho més comprensible, separarem el casos relacionats amb l'edició dels d'exportacions.

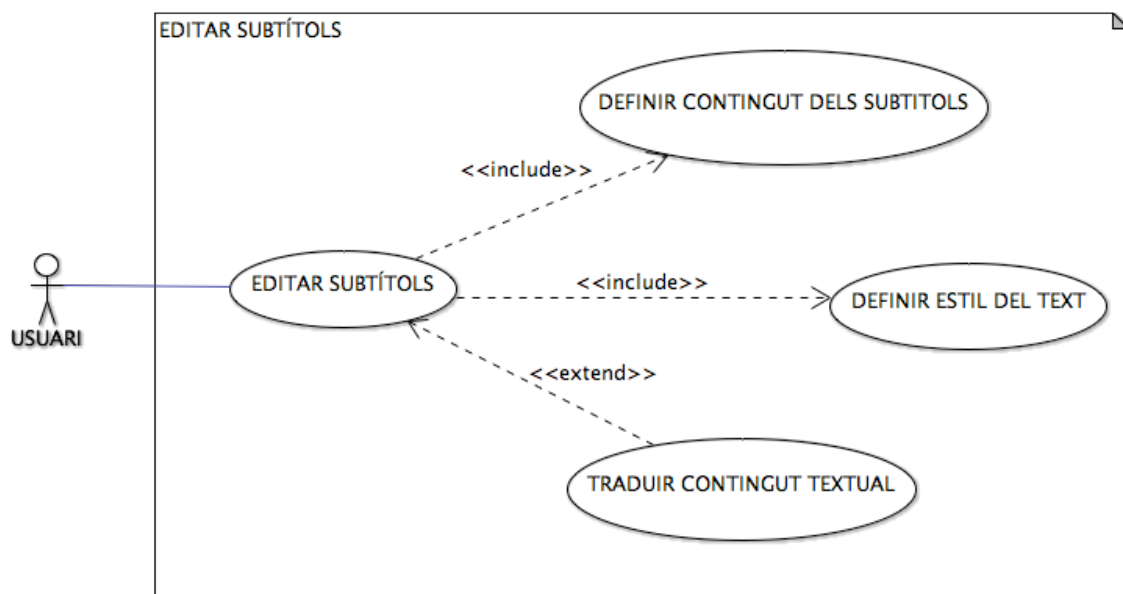


Figura 2. Diagrama d'ús de l'edició de subtítols.

Un cop importat un arxiu, el sistema mostra un llistat amb la informació dels subtítols a la finestra d'edició. En aquesta punt, és on l'usuari pot començar a editar la informació dels seus subtítols. El cas d'us editar subtítols bàsicament

serveix per incloure i estendre altres casos d'ús com els que es veuen en el diagrama.

CAS D'ÚS: Definir el contingut textual dels subtítols

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Escenari principal d'èxit:

1. El sistema mostra el llistat de subtítols a la pantalla d'edició.
2. L'usuari clica sobre la línia de subtítol que vol modificar i canvia el seu contingut.
3. El sistema mostra la línia modificada.

Extensions:

- 2b. L'usuari vol rectificar l'edició dels subtítols
- 2b.1. Tornem al punt 1

CAS D'ÚS: Definir l'estil del text

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Escenari principal d'èxit:

1. El sistema mostra el llistat de subtítols a la pantalla d'edició.
2. L'usuari marca quines línies vol destacar en cursiva.
3. El sistema mostra la línia destacada i el text en l'estil marcat.

Extensions:

- 2b. L'usuari vol rectificar la línia destacada
- 2b.1. L'usuari desmarca l'opció de cursiva

CAS D'ÚS: Editar subtítols

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Precondició: L'usuari ha importat un arxiu de subtítols amb èxit

Escenari principal d'èxit:

1. El sistema mostra el llistat de subtítols a la pantalla d'edició.
2. L'usuari pot modificar el contingut textual dels subtítols.
3. L'usuari pot definir l'estil d'una línia de subtítols.

Extensions:

- *) Si l'usuari no fa el punt 3, 4 o 5.
- *)1 El llistat no es modificarà i quedarà com es mostra en el punt 1

CAS D'ÚS: Traduir els subtítols a un altre idioma

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Casos d'ús relacionats: Estén el cas d'ús "Editar subtítols" en el pas 1.

Precondició: L'usuari ha importat un arxiu de subtítols amb èxit

Escenari principal d'èxit:

1. L'usuari activa l'opció traduir els subtítols del selector de traducció
2. El sistema mostra el selector amb l'opció traducció activada
3. El sistema mostra una columna nova de contingut textual buida
4. El sistema mostra els camps de l'opció de la traducció
5. L'usuari selecciona l'idioma d'origen i el de destí
6. L'usuari clica el botó de traduir
7. El sistema tradueix els subtítols
8. El sistema mostra els subtítols traduïts dins dels camps a la nova columna.

Extensions:

- 8b. A l'usuari no li agrada la traducció i prefereix l'original
- 9b.1. L'usuari desactiva l'opció traducció del selector
 - 9b.2. El sistema amaga els subtítols traduïts
- 8b. L'usuari no li agrada la traducció d'alguns subtítols i vol rectificar.

9c.1 L'usuari modifica el contingut textual dels subtítols.

2,3,4,5,6,7b. L'usuari finalment no vol traduir els subtítols

2,3,4,5,6,7b.1 Desactiva el selector.

2,3,4,5,6,7b.2 El sistema amaga els subtítols traduïts

Com es veu en el diagrama següent, els casos d'ús d'exportació són força més complexes que els d'edició. Per intentar entendre millor el funcionament de l'exportació, primer, especificarem de manera precisa els dos casos generals i a partir d'allà definirem els altres casos d'ús.

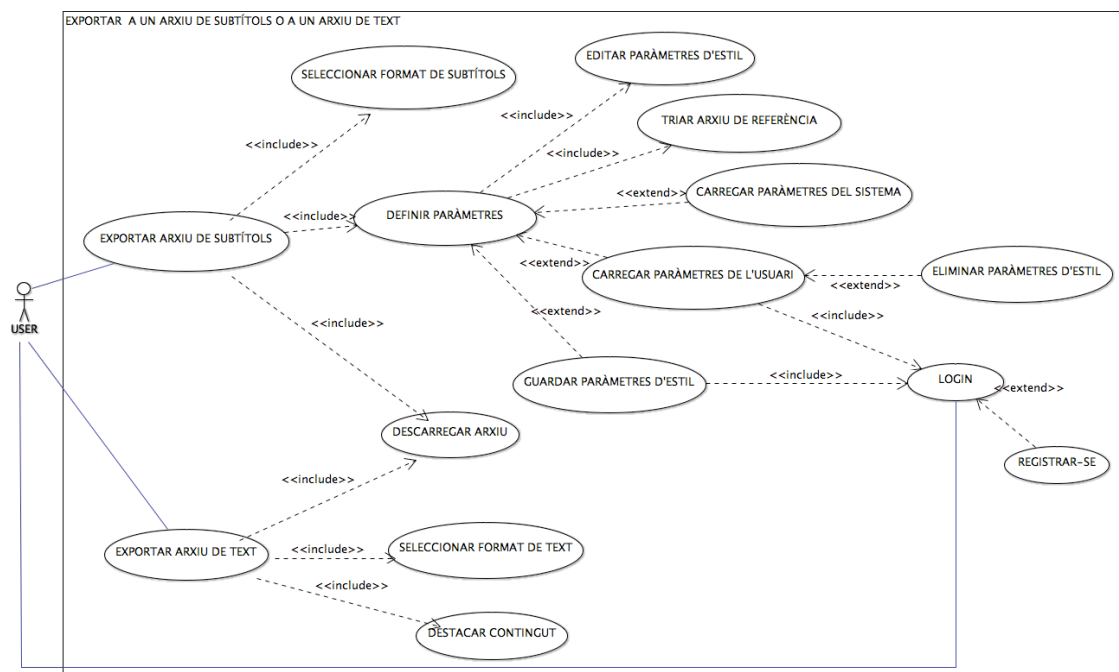


Figura 3. Diagrama d'ús de l'exportació d'arxius.

CAS D'ÚS: Exportar arxiu de text destacant alguns subtítols

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Escenari principal d'èxit:

1. L'usuari importa un arxiu
2. El sistema mostra la pantalla d'edició
3. L'usuari, si vol, edita els subtítols

4. L'usuari selecciona exportar arxiu de text
5. El sistema mostra les opcions d'exportació
6. L'usuari selecciona a quin format de text vol exportar
7. L'usuari tria que vol destacar els subtítols
8. El sistema mostra un llistat amb tots els subtítols.
9. L'usuari selecciona les línies que vol destacar
10. L'usuari selecciona descarregar l'arxiu
11. El sistema recull tota la informació i crea l'arxiu.
12. El sistema demana a l'usuari on vol descarregar l'arxiu.

Extensions:

- 6b. L'usuari selecciona que no vol destacar cap línia
 - 6b.1. Passem directament al punt 10.
- 7-10) L'usuari vol canviar el format d'exportació.
 - 7-10).1. L'usuari torna al punt 6
- 4-10) L'usuari vol editar els subtítols
 - 4-10).1. L'usuari torna al punt 6
- *) L'usuari vol canviar l'arxiu importat
 - *.1. L'usuari torna al punt 6

CAS D'ÚS: Exportar arxiu de subtítols amb un arxiu de referència

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Escenari principal d'èxit:

1. L'usuari importa un arxiu
2. El sistema mostra la pantalla d'edició
3. L'usuari, si vol, edita els subtítols
4. L'usuari selecciona exportar arxiu de subtítols
5. El sistema mostra els formats d'exportació
6. L'usuari selecciona a quin tipus de format vol exportar
7. El sistema mostra les opcions d'arxiu de referència
8. L'usuari tria un arxiu de referència

9. El sistema comprova l'arxiu de referència.
10. El sistema mostra el llistat dels paràmetres d'estil
11. L'usuari defineix els paràmetres d'estil
12. L'usuari selecciona descarregar l'arxiu
13. El sistema recull tota la informació i crea un nou arxiu.
14. El sistema demana a l'usuari on vol descarregar l'arxiu.

Extensions:

- 4/6/8/11b L'usuari vol editar el llistat de subtítols
 - 4/6/8/11b.1. Tornem al punt 2
- 6c Es selecciona un format que no permet definir cap paràmetre
 - 6c.1 Passem directament al punt 11
- 8/11c L'usuari vol canviar el format d'exportació.
 - 8/11c.1. Tornem al punt 4
- 8d L'usuari selecciona crear un arxiu de referència nou
 - 8d.1 L'usuari defineix els paràmetres de vídeo
- 9b L'arxiu no és correcte
 - 9b.1. El sistema mostra un missatge específic d'error
 - 9b.2. Tornem al punt 1.
- *) L'usuari vol canviar l'arxiu importat
 - *.1. Tornem al punt 1

CAS D'ÚS: Definir els paràmetres de vídeo

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Casos d'ús relacionats: Estén el cas d'ús "Exportar arxiu de subtítols" en el pas 8.

Escenari principal d'èxit:

1. El sistema mostra els paràmetres de vídeo
2. L'usuari defineix els paràmetres de vídeo

Per acabar, definirem el cas d'ús principal que s'estenen de la definició de paràmetres i que inclou un usuari enregistrat com és el de guardar els paràmetres d'estil.

CAS D'ÚS: Guardar els paràmetres d'estil

Actor principal: Usuari.

Nivell: Usuari.

Àmbit: Sistema.

Casos d'ús relacionats: Estén el cas d'ús "Exportar arxiu de subtítols" en el pas 11.

Escenari principal d'èxit:

1. L'usuari activa la llista de paràmetres i prem el botó de guardar paràmetre
2. Si l'usuari no ha iniciat la sessió, s'identifica.
3. El sistema comprova si els paràmetres introduïts són correctes.
4. El sistema demana a l'usuari un nom pels paràmetres
5. L'usuari introdueix el nom.
6. El sistema comprova que el nom no existeixi
7. El sistema enregistra la informació dels paràmetres al sistema.
8. El sistema actualitza el llistat de paràmetres i mostra els nou paràmetres a la llista.

Extensions:

- 3b Els paràmetres introduïts no són correctes
 - 3b.1 El Sistema mostra un missatge d'error
 - 3b.2 Tornem al punt 1
- 6b.1 El nom dels paràmetres ja existeix
 - 6b.1 El sistema mostra un missatge a l'usuari
 - 6b.2 Tornem al punt 4

2.2.2 Diagrames d'activitats

Hem creat dos diagrames d'activitats que resumeixen el funcionament dels casos d'ús: exportar arxiu de subtítols i exportar arxiu de text.

EXPORTAR ARXIU DE SUBTÍTOLS

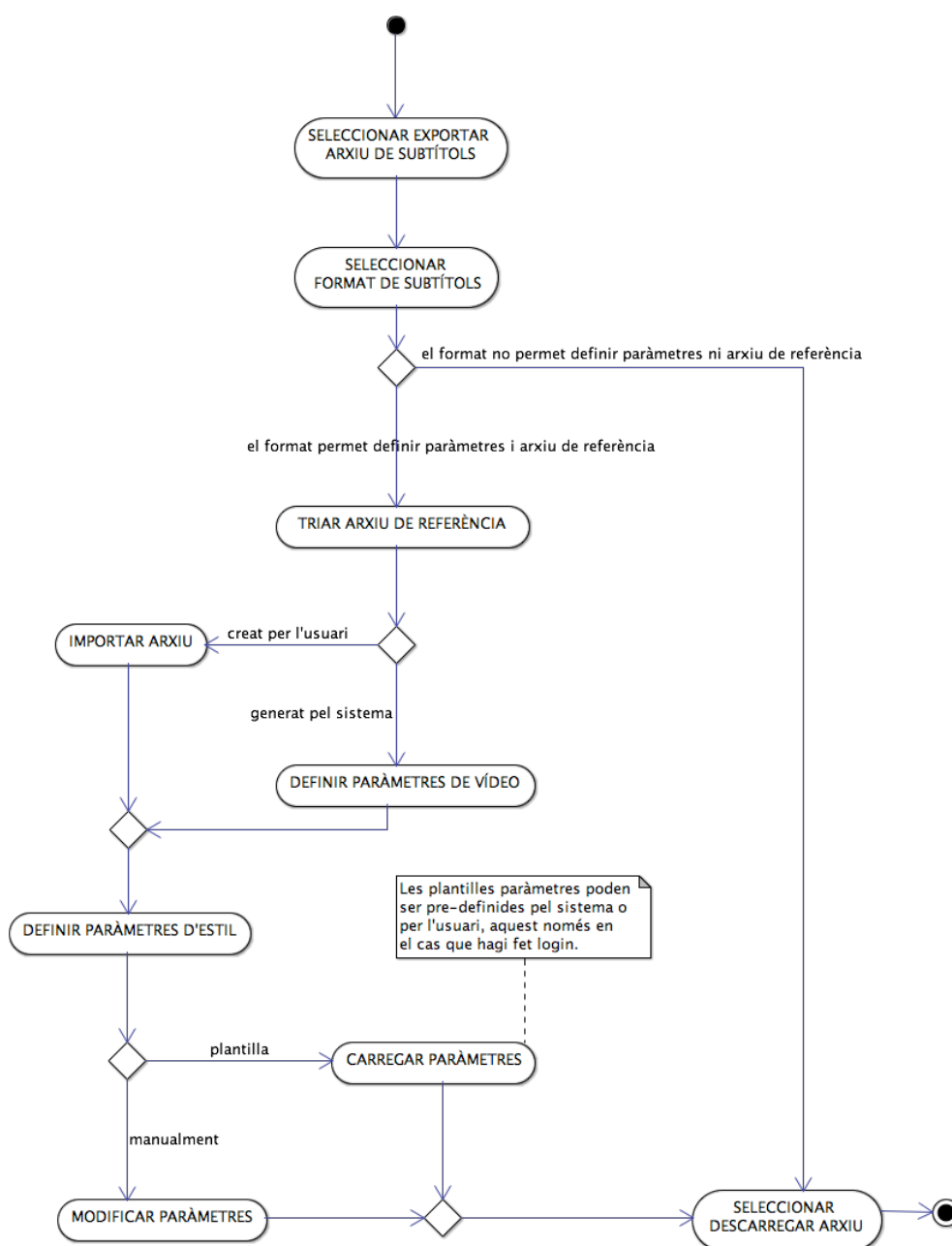


Figura 4. Diagrama d'activitats: exportar arxius de subtítols.

EXPORTAR ARXIU DE TEXT

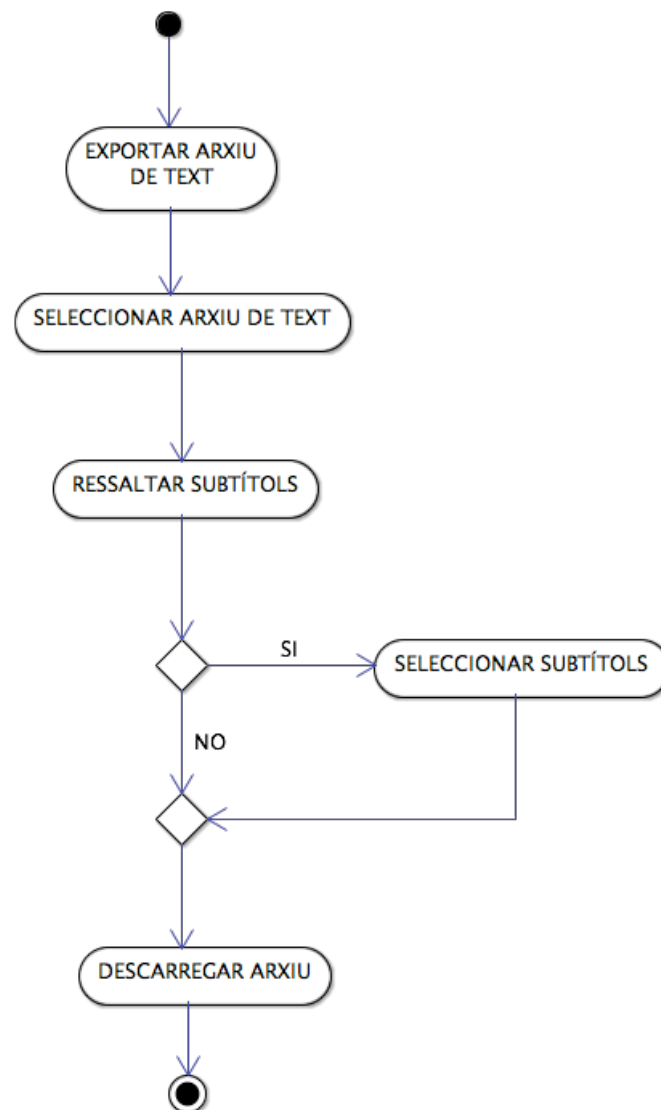


Figura 5. Diagrama d'activitats: exportar arxius de text.

Per altra banda, com es pot deduir de l'especificació dels casos d'ús, el domini del nostre sistema no servirà només per connectar amb la base de dades, sinó que haurà de solucionar el requisit principal de l'aplicació: la lectura, codificació i creació d'arxius.

Per aquesta raó, hem creat un diagrama d'activitats on hem mesclat els casos d'usos principals del actor usuari i les activitats principals que haurà de resoldre el nostre domini.

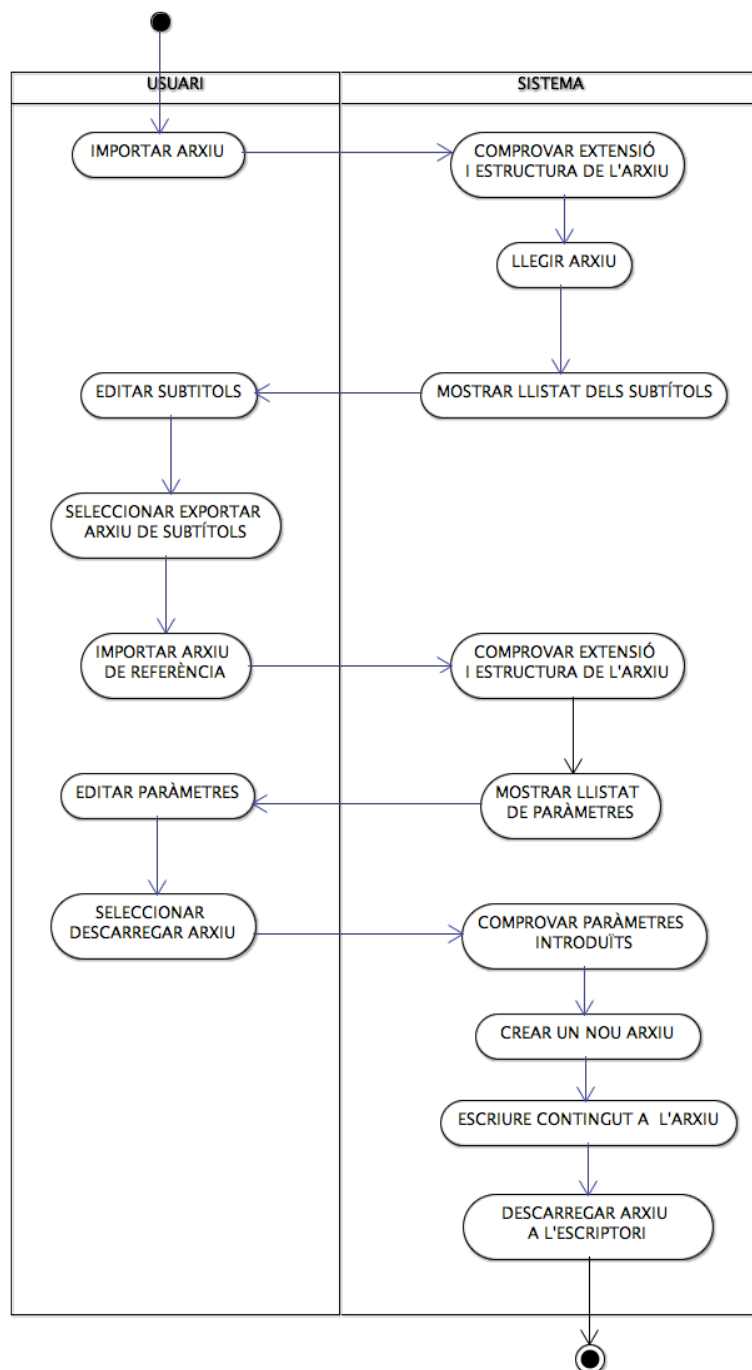


Figura 6. Diagrama d'activitats: casos d'ús d'usuari i sistema.

2.2.3 El diagrama de classes

El nostre diagrama estàtic es basa en dues classes abstractes que seran les que definiran el comportament del nostre convertidor: *SubtitleFormat* i *ParamPreset*. Aquestes faran d'interfície per complir el principi obert-tancat del programari, d'aquesta manera, es podran afegir nous tipus de format i nous paràmetres sense haver de modificar el codi. És important tenir en compte que pot existir algun *SubtitleFormat* que no tingui *ParamPreset*.

Tots els formats hauran d'implementar tres funcions principals: llegir els arxius i retornar un array amb la informació dels subtítols, crear i escriure un arxiu nou i comprovar que tant l'estructura de l'arxiu original com la de referència és correcta.

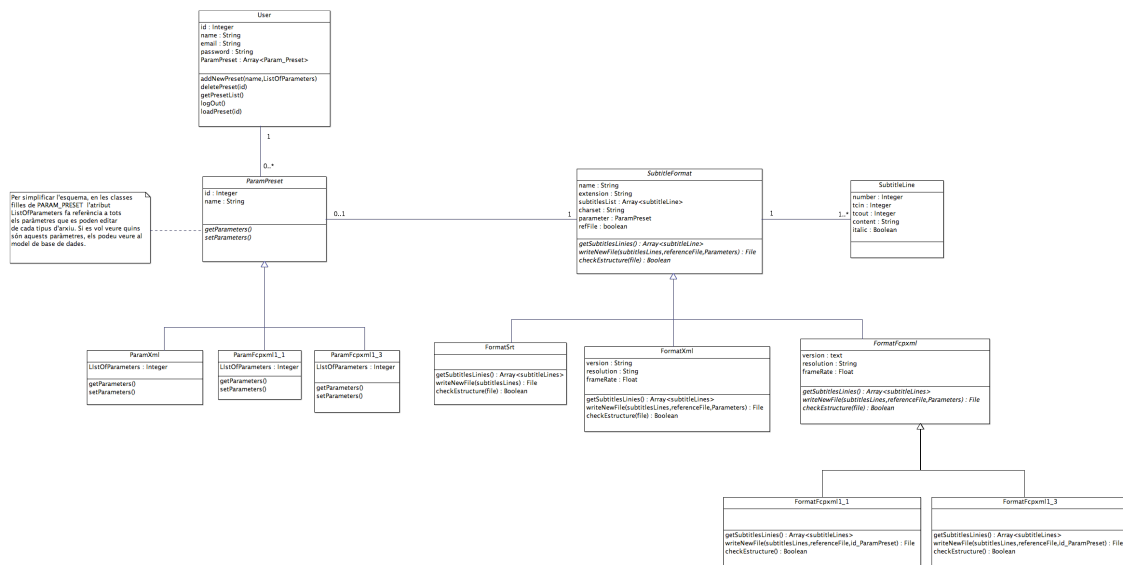


Figura 7. Diagrama de classes.

Per veure el diagrama amb més detall, podeu consultar l'arxiu

Memòria/Fitxers_adjunts/PAC_FINAL_ClassDiagram_Hernandez_Victor.png

2.4.4 Interfície gràfica

Una particularitat que volem destacar de la interfície gràfica i que ens permeten les tecnologies actuals és que el bloc principal de l'aplicació tingui una sola pàgina. És a dir, que les opcions d'edició, exportació a arxius de subtítols i exportació a arxius de text estiguin a un mateix nivell. Això facilitarà l'exportació a diferents formats i ajustar els paràmetres sense perdre la feina d'edició.

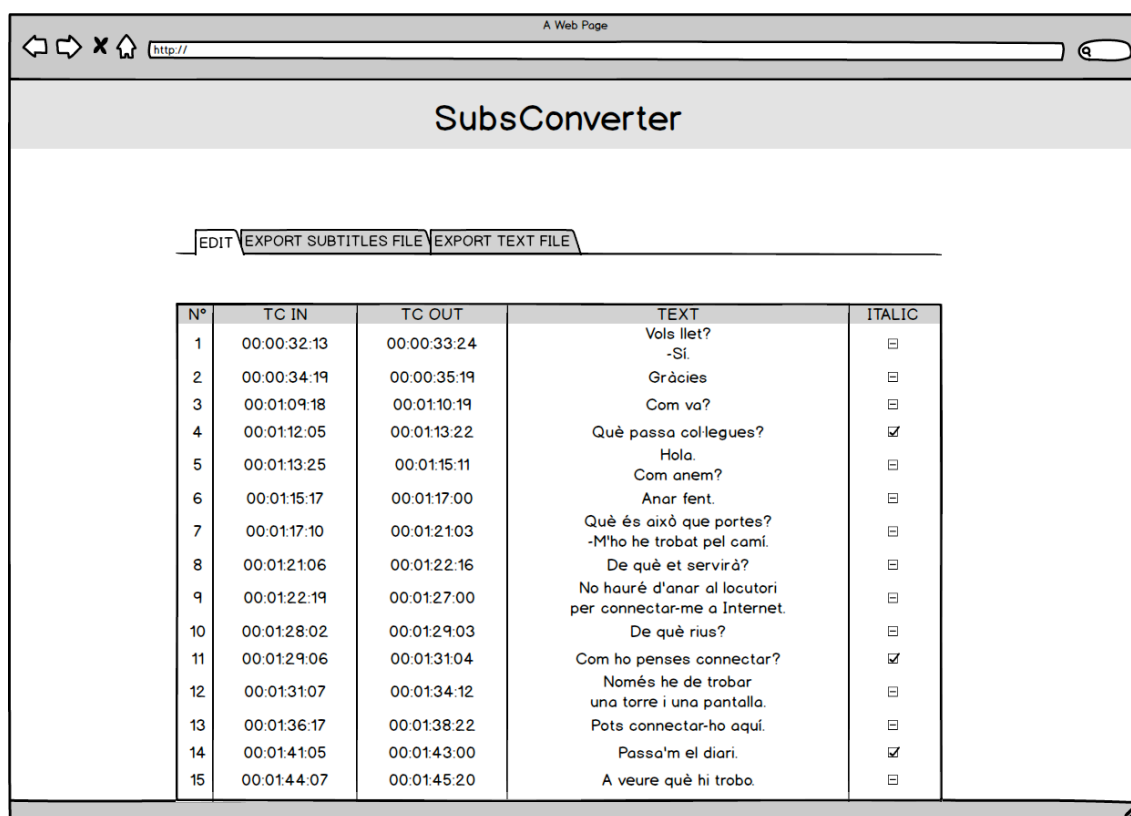


Figura 8. La interfície gràfica principal en una sola pàgina.

2.2.5 Estructura de la base de dades

La principal funció de l'estructura de la base de dades és relacionar cada usuari amb els llistats de paràmetres que ha emmagatzemat per cada tipus d'arxiu. Com que cada tipus d'arxiu té un llistat de paràmetres diferents, crearem una taula per cada tipus de paràmetre i aquestes tindran una relació de cardinalitat N:1 amb els usuaris, atès que cada usuari pot emmagatzemar configuracions

diferents de cada format de subtítols. Es destaca que els atributs 'name' de les taules de paràmetres són únics.

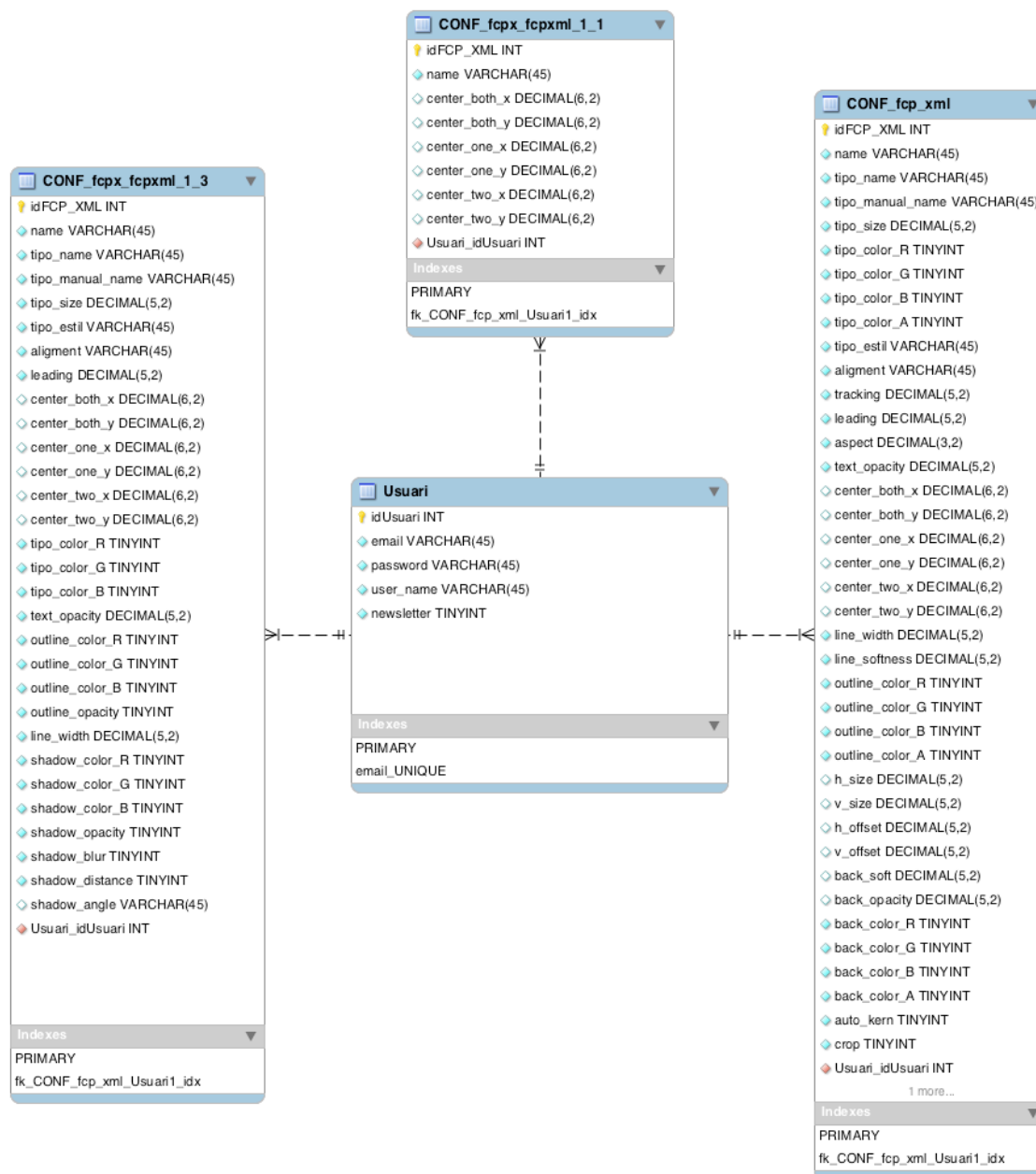


Figura 9. Taules de la base de dades.

2.3 Arquitectura

2.3.1 Client/Servidor

Per la relació client/servidor haurem de trobar una tecnologia que ens permeti implementar el patró d'arquitectura Model-Vista-Controlador. Per aquesta raó i amb la voluntat de descobrir una tecnologia molt estesa en el món professional, ens hem decantat per utilitzar Symfony, un entorn de treball(framework) de codi obert desenvolupat en PHP.

Symfony està format per un conjunt de més de 30 components i pensat per la realització d'aplicacions web. Symfony compta amb una gran comunitat de desenvolupadors, més de 300.000 a 120 països, això ens facilitarà suport, seguretat i documentació.

A més, com tot bon entorn, Symfony ens permet centrar-nos en la creació del lloc i no en la carrega que suposa implementar les eines comuns que comparteixen la majoria de llocs web.

L'arquitectura MVC és l'encarregada de separar la lògica de negoci (el model) i la presentació(la vista) per la qual cosa s'aconsegueix un manteniment més senzill de les aplicacions.

Per fer de pont entre el model i la vista, hi ha el controlador que captura les peticions de l'usuari, les envia al model i les retorna a la vista. El model s'encarrega de l'abstracció de la lògica del negoci i de la relació amb les dades, fent que la vista i les accions siguin independents.

Per la vista utilitzarem les tecnologies següents:

- Per l'etiquetatge usarem HTML 5 i plantilles Twig
- Pels estils usarem amb tot seguretat CSS3 i, probablement, les llibreries de Bootstrap

- Per la part dinàmica usarem Javascript i dos dels seus entorns: JQuery i Angular. També, Ajax, que en la nostre aplicació tindrà un gran protagonisme i ens permetrà realitzar connexions al servidor de manera asincrònica.

En canvi, pel controlador i el model treballarem, principalment amb PHP. No obstant, com que Symfony també permet l'ús de Yaml i XML, potser els acabem utilitzant.

Per altra banda, el client interactuarà bàsicament amb l'aplicació a través de formularis. El sistema li demanarà dades per després processar-les. Per tant, una de les funcions principals de la part client serà l'ús de formulari i la validació de les dades introduïdes abans que aquestes s'enviïn al servidor.

2.3.2 Symfony

Abans d'explicar quines són les principals característiques arquitectòniques de Symfony, triarem amb quina versió del programari disponible volem treballar. Per això, usarem el mapa de versions següent, on es pot veure el desenvolupament, l'estabilització i el manteniment de les versions antigues, actuals i futures.

Després de saber que entre les versions 2.8 i 3.0 no hi ha gaire diferència, que la versió 3.2 encara està en proves i que la 2.8 és una versió de llarga duració que tindrà un manteniment de seguretat fins a finals de 2019, ens hem decidit per l'ús de la 2.8.

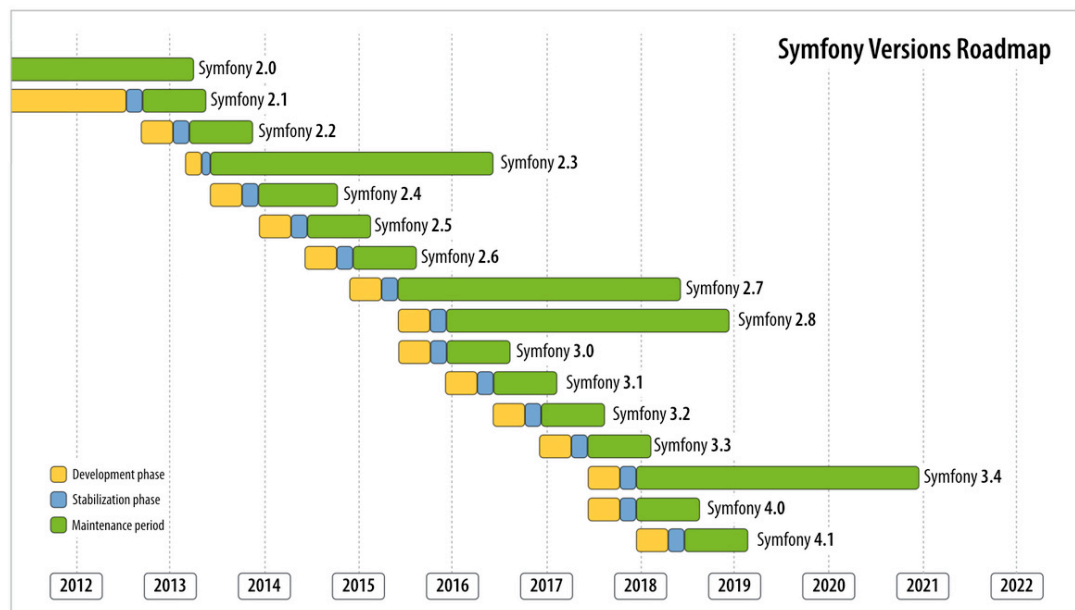


Figura 10. Mapa de versions del programari Symfony.

De l'arquitectura de Symfony destaquem els aspectes que ens han semblat més rellevants i els intentarem relacionar amb la nostra aplicació:

a) El sistema Request/Response

Per simplificar el funcionament de Symfony, podem dir que aquesta tecnologia es basa principalment en el protocol HTTP, que defineix la forma de comunicació entre un client i el servidor.

A Symfony existeixen dues classes, Request i Response, que són les encarregades de interactuar de forma senzilla entre la petició del client i la resposta del servidor. Per tant, tot està pensat per simplificar i modular aquesta relació.

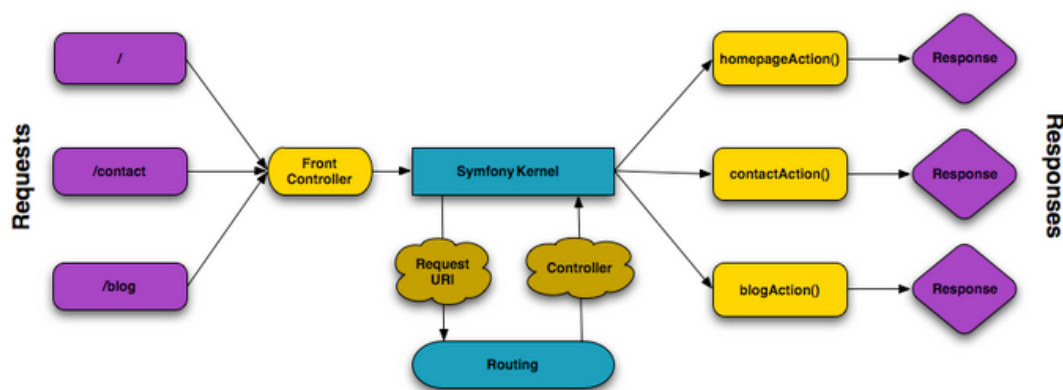


Figura 11. Resum gràfic de les classes Request-Response a Symfony.

Si subdividim la relació entre client(request) i servidor(Response), ens trobem una sèrie de processos intermedis que ens ajuden a modular la relació.

REQUEST>ROUTING>CONTROLLER/ACTION>MODEL>TEMPLATE>RESPONSE

- Request: l'objecte Request conté la informació de la petició.
- Routing: associa la URL amb una acció del controlador.
- Controller/Action: llegeix la informació de la classe Request, contacta amb el model i retorna la classe Response.
- Template: plantilles que usarem per la vista.
- Model: on s'implementa la lògica de l'aplicació i es crea la relació amb la bases de dades.

b) Els components

Symfony conté un conjunt de biblioteques PHP lliures i reutilitzables anomenades components. Aquestes són la base per la construcció de l'aplicació. N'hi ha aproximadament un total de 30 aproximadament i qualsevol d'aquests components és utilitzable fora de Symfony.

Un dels components de Symfony que segur que utilitzarem en la nostra aplicació serà FORM perquè facilita la creació, el processat i la reutilització

de formularis HTML. De fet, tota la majoria part de la nostra aplicació es basa en la gestió de formularis.

c) Els bundles

Un bundle és un concepte similar al dels connectors o plugins en altres aplicacions. La diferència és que a Symfony tot és un bundle, incloent tant la funcionalitat bàsica de la plataforma com el codi escrit per a la teva aplicació.

Els bundles són la part més important de Symfony, ja que permeten utilitzar funcionalitats construïdes per tercers o emmagatzemar les pròpies per distribuir-les i reutilitzar-les en altres projectes. A més, faciliten molt l'activació o desactivació de determinades característiques dins d'una aplicació.

Per exemple, un dels bundles que utilitzarem en la nostra aplicació serà Doctrine, que ens facilitarà la connexió amb el sistema de gestió de la base dades.

La instal·lació de bundles la podem fer a través de composer:

```
$ composer require doctrine/doctrine-bundle
```

Per activar qualsevol bundle dins de la nostra aplicació, l'haurèm d'afegir a la matriu `$bundles` de la classe `app/AppKernel.php`.

d) La instal·lació de Symfony

Per instal·lar Symfony en el nostre sistema optem amb les opcions següents: (analitzem només el cas de Linux i Mac)

- Utilitzant l'instal·lador oficial de Symfony:

```
$ sudo curl -Ls https://symfony.com/installer -o /usr/local/bin/symfony
```

```
$ chmod a+x /usr/local/bin/symfony  
$ symfony new PROJECT_NAME(blog) PROJECT_VERSION(2.8)
```

- Utilitzant composer:

```
$ composer create-project symfony/framework-standard-edition  
blog "2.8"
```

Un cop instal·lat Symfony, en el root principal tindrem la estructura de directoris següent:

- app: on es guarden els arxius de configuració de l'aplicació.
- bin: on hi ha les comandes que poden ser executades des del terminal.
- src: on guardem els bundles específics de l'aplicació o el codi personalitzat.
- vendor: on hi ha els components de l'aplicació, els bundles reutilitzables i les llibreries de tercers.
- web: on hi posarem el controlador frontal i els recursos públics del nostre projecte com poden ser les imatges, fulls d'estils o els scripts de Javascript.

2.3.3 Base de dades

Per la base de dades, utilitzarem Doctrine, un ORM (*Object Role Modeling*) que actua com a capa d'abstracció per sobre del Sistema de Gestió de Base de Dades i, per tant, simplifica l'accés i l'ús de les dades, i ens permet canviar el SGBD sense haver de modificar el codi de l'aplicació. La versió oficial estàndard de Symfony ja conté el bundle de Doctrine instal·lat.

Per configurar Doctrine haurem de definir els paràmetres següents:

```
# Doctrine Configuration (app/config/config.yml)  
doctrine:  
    dbal:
```

```
driver: pdo_mysql
host: '%database_host%'
port: '%database_port%'
dbname: '%database_name%'
user: '%database_user%'
password: '%database_password%'
charset: UTF8

# if using pdo_sqlite as your database driver:
# 1. add the path in parameters.yml
# ex. database_path: '%kernel.root_dir%/data/data.db3'
# 2. Uncomment database_path in parameters.yml.dist
# 3. Uncomment next line: path: '%database_path%'
```

2.3.4 API/Lliberies de tercers

La nostra aplicació utilitzarà diferents API's i lliberies de tercers. A banda de les ja esmentades en els punts anteriors. A continuació, expliquem quines són les més rellevants:

- a) **Api de GoogleTranslate:** utilitzarem aquesta API per la traducció dels subtítols a altres llengües. A més, ho farem a través de la classe de codi obert “php-google-translate-free”, publicada a Github per Adrián Barrio Andrés. Enllaç: <https://github.com/statickidz/php-google-translate-free/>
- b) **WhiteOctoberTCPDFbundle:** aquest bundle integra la llibreria de PHP de codi obert que ens permet crear PDF's. A més, permet crear documents Unicode UTF-8. Enllaç a la llibreria original: <https://tcpdf.org/> Exemple Unicode: https://tcpdf.org/examples/example_008/. Enllaç al bundle: <https://github.com/whiteoctober/WhiteOctoberTCPDFBundle>.
- c) **ExcelBundle:** aquest bundle integra la llibreria PHPExcel a Symfony, i permet crear, llegir i escriure fitxers de text en format excel. Enllaç: <https://github.com/liuggio/ExcelBundle>
- d) **Spectrum:** aquest plugin de JQuery ens permet implementar selectors de colors en els elements HTML d'una manera molt visual i atractiva. A més, ens permet introduir el color de manera manual en format RGB,

HSB o HTML. El plugin ens fa sentir com si estiguéssim utilitzant l'Adobe Photoshop. Enllaç: <http://plugins.jquery.com/colpick/>

- e) **FOSUserbundle**: llibreria molt completa que proporciona un marc per a la gestió d'usuaris i que té com a objectiu facilitar la implementació de les tasques més habituals. És una dels bundles més utilitzats pels usuaris de Symfony. Enllaç:

<http://symfony.com/doc/current/bundles/FOSUserBundle/index.html>

2.4 Estructura dels formats de subtítols

Una de les funcionalitats cabdals de l'aplicació és que ha de permetre a l'usuari importar i exportar arxius de subtítols en els formats Subrip(.srt), Final Cut Pro Interchange Format(.xml) i Final Cut Pro Format(.fcpxml). A més, l'arquitectura ha de permetre afegir més formats en un futur. Per tant, haurem de preveure mètodes de lectura i d'escriptura pels tres formats.

- a) **La lectura**: haurem d'obtenir les mateixes dades de tots tres formats per mostrar-les a l'usuari.
- Contingut textual.
 - Codi de temps d'entrada
 - Codi de temps de sortida
 - L'estil del text, si és pla o cursiva. La resta d'estils de text com la negreta, subratllat o el color concret d'un subtítol, encara que molts formats els implementen, no ens interessen perquè estan fora de l'àmbit professional al qual enfoquem l'aplicació.

La manera de realitzar la lectura serà molt diferent si es tracta del format .srt, .xml o fcpxml. Els formats .srt només emmagatzemen la informació sobre les subtítols, en canvi els de Final Cut Pro(FCP) són arxius que serveixen per emmagatzemar tota la informació que hi ha en una

seqüència de vídeo. Per tant, hi podem trobar la informació sobre els clips de vídeo, àudio, marques, efectes, títols de crèdit, etc.

- b) **L'escriptura:** l'usuari pot editar part del contingut de l'arxiu que ha penjat i, depenent del format, definir paràmetres sobre l'estil del text. Per tant, l'escriptura haurà de recollir tota aquesta informació i guardar-la en un nou arxiu.

En el cas dels arxius de FCP, com que treballem amb arxius .xml d'un projecte de vídeo, la d'informació que conté l'arxiu i que no fa referència als subtítols conté l'arxiu pot ser important per l'usuari. Per aquesta raó, l'aplicació donarà l'opció a l'usuari d'exportar un arxiu .xml nou o de penjar un de referència que tingui la informació del seu projecte. Per tant, en el cas de FCP es donen dos supòsits d'escriptura.

- c) **El paràmetres:** A banda de tots els elements comuns que han de tenir els subtítols i que ja hem definit en l'apartat d'escriptura, els arxius .xml permeten desar molts paràmetres que afecten a l'estil del text. Per tant, permetrem a l'usuari definir aquest paràmetres, que seran diferents segons el tipus i versió d'xml.

Per tot això, a continuació analitzem en detall els tres formats.

2.4.1 El format SubRip(.srt)

a) L'ESTRUCTURA

Els subtítols .srt són arxius de text sense format que contenen la informació bàsica dels subtítols. Exemple:

```
1
00:00:00,394 --> 00:00:03,031
<i>Adelante!</i>

2
00:00:03,510 --> 00:00:05,154
```

- Ponte esto.
- No gracias. Estoy bien sin él.

3

00:00:05,274 --> 00:00:07,021

Como prefieras.

Cada subtítol està compost per:

- Número del subtítol
- Codi de temps d'entrada + "-->" + Codi de temps de sortida. El Format del codi de temps és (hores : minuts : segons , mili-segons).
- Una línia en blanc que marca el final d'un subtítol i l'inici del següent.

També, es poden afegir estils al text amb l'ús d'etiquetes:

- *Cursiva* <i></i>
- **Negreta**
- Subratllat<u></u>
- **Color**

b) LECTURA I ESCRIPTURA

Per la lectura, no tindrem en compte les etiquetes , <u> i , en canvi sí, l'etiqueta <i>, que defineix les cursives. L'escriptura, en principi no suposa cap problema.

c) PARÀMETRES QUE POT DEFINIR L'USUARI

L'usuari no podrà especificar cap paràmetres més enllà del contingut i si vol que el text tingui un estil regular o itàlic.

2.4.2 Els formats de Final Cut Pro

Els arxius de Final Cut Pro són arxius XML (Extensible Markup Language) que serveixen per emmagatzemar de manera textual tota la informació d'un projecte d'edició. La idea d'aquest format és facilitar la transmissió d'informació entre Final Cut Pro i altres eines que hi ha al mercat i que no són compatibles amb el programari.

D'aquesta manera, en un arxiu XML de FCP, podem exportar tota la informació del nostre projecte d'edició de vídeo, com les pistes de vídeo, àudio, efectes, títols, marques, etc... En conseqüència, si volem importar els subtítols al nostre projecte de FCP, ho haurem de fer a través de l'XML.

Per tant, hem de considerar la possibilitat de què l'usuari que vol exportar els subtítols en un format de FCP, vulgui fer-ho sobre un arxiu propi per tal de sincronitzar els subtítols amb la seva feina d'edició.

També, hem de tenir en compte que els subtítols en FCP no són línies de text, sinó que són clips de vídeo generats a través d'un editor textos. Aquests es creen amb l'editor de títols del programari d'edició, se'ls hi dona la forma desitjada i es col·loquen dins en una de les pistes de vídeo dins de la seqüència que estiguem treballant. Normalment, s'utilitzen efectes de text predefinits que ja venen inclosos amb el programari.

Actualment, en el món professional s'utilitzen dues versions de Final Cut Pro, la versió 7(FCP7), antiga però encara força utilitzada i que treballa amb arxius Final Cut Interchange Format, amb extensió .xml, i la nova versió, Final Cut Pro X(FCPX), que comença a desplaçar a l'antiga i utilitza arxius Final Cut Pro X Format, amb extensió .fcpxml. Això ens obliga a analitzar els dos formats per separat.

Pel nostre anàlisi haurem de tenir en compte:

- Versions diferents dels arxius XML i compatibilitat amb les versions del programari
- Quins editor s'utilitzen habitualment per generar els subtítols al programari i com es diferencien a l'XML.
- Analitzar l'estructura general de l'XML
- Analitzar l'estructura dels subtítols i com es defineixen els paràmetres
- Com efectuar la lectura dels subtítols
- Com escriure els subtítols dins d'un arxiu
- Quins paràmetres pot definir l'usuari

Això ho farem, per una banda, exportant i analitzant un arxiu de mostra i, per l'altra, analitzant la documentació de suport d'Apple.

A causa de la seva extensió, l'anàlisi complert dels formats Final Cut Pro Interchange Format i Final Cut Pro X Format els podeu trobar al l'apartat [d'Annexos](#).

2.5 Disseny gràfic i usabilitat

2.5.1 Perfil d'usuari

Els usuaris principals de la nostra aplicació són persones que es dediquen a l'edició de vídeo professional o a l'edició de subtítols. Per tant, el públic objectiu és reduït i concret. No obstant, hem de pensar que cada vegada més hi ha gent que utilitza els editors de vídeo com a complement multimèdia o per la creació de vídeos personals o familiars i que també poden ser usuaris potencials.

Per tant, intentarem que l'aplicació tingui un caire professional, però que alhora sigui transparent i accessible a un usuari comú. A més, si ens fixem amb les alternatives que hi ha al mercat, la millora de la arquitectura de la informació és un dels aspectes on hi ha molt marge de millora. Aquesta idea, va ser repetida per diferents *stakeholders* durant les entrevistes.

2.5.2 Història d'ús

En aquest apartat, s'explica una història d'us real on es veuen els beneficis que pot comportar l'ús de la nostra aplicació.

Nom: Manel Martínez

Edat: 35 anys

Professió: Muntador de vídeo autònom

Descripció de l'escenari: El Manel està muntant un llargmetratge i acaba de finalitzar la seva primera versió. El productor del film el truca, perquè hi ha uns inversors americans que estan interessants en veure'l, i li demana que faci un còpia amb subtítols en anglès incrustats en pantalla.

L'editor de subtítols envia al Manel un arxiu .srt amb els subtítols anglesos sincronitzats. El Manel té el muntatge d'imatge i so a l'editor de vídeo, en el seu cas el Final Cut Pro. La manera més senzilla d'incrustar els subtítols, és importar l'arxiu .srt al programa d'edició i després exportar una movie amb els subtítols incrustats. El problema és que a Final Cut Pro no es poden importar subtítols amb format .srt.

El Manel busca per la xarxa alguna eina que li solucioni el problema però no la troba. Llavors truca a l'editor de subtítols i li demana si li pot enviar l'arxiu en altres formats. Aquest li envia els subtítols en varis fitxers XML, però cap d'ells és compatible amb Final Cut Pro.

El Manel truca al productor i li demana si li pot entregar una còpia amb els subtítols al marge. El productor, entre incrèdul i sorprès, li diu que no, que ell necessita una còpia amb els subtítols incrustats perquè no sap amb quin tipus de reproductor i quin tècnic es trobaran a la sala d'exhibició, i que és imprescindible que els subtítols no fallin.

El Manel desesperat truca a un amic i li demana ajuda. L'amic li diu que hi ha

un aplicació que costa 300 euros que li soluciona el problema. Finalment, el Manel, gràcies a un comentari d'un usuari en un fòrum, troba una aplicació X a Internet que li permet exportar, però només ho fa a la versió antiga de Final Cut.

Davant de la desesperació, el Manel instal·la l'antic programari que ja no utilitzava, crea un nou projecte, importa la pel·lícula i sincronitza el subtítols que ha exportat amb l'aplicació X.

Un cop carregat l'arxiu de subtítols, el Manel es dona compte que alguns caràcters es visualitzen de manera errònia i que no s'han respectat els subtítols que contenen cursives. El Manel decideix rectificar els tots els errors manualment, cosa que li suposa quasi 45 minuts de feina. Tampoc li agrada el tipus i la mida de tipografia utilitzada. En aquest cas, el Manel decideix no tocar-ho perquè hauria de rectificar subtítol a subtítol i la pel·lícula en té 892. Finalment, exporta la pel·lícula i entrega la còpia per exhibició.

Tres dies després el productor el truca i li diu que la projecció havia anat molt bé però que la mida dels subtítols era massa gran.

2.5.3 Usabilitat (UX)

En aquest apartat es defineixen els aspectes sobre la usabilitat que es tenen en compte alhora de crear l'aplicació:

1. **Versió d'escriptori:** de moment, la nostra aplicació només té sentit utilitzar-la des de l'escriptori. Per tant, la nostra web ha d'estar pensada només pel navegador web de l'escriptori.
2. **L'etiquetatge:** a la nostra aplicació hi haurà un munt de conceptes tècnics que s'han de descriure i que l'usuari ha de comprendre. Per tant, l'etiquetatge ha de ser clar i entenedor. Si es troben dificultats,

s'afegiran explicacions o es posaran exemples visuals per tal d'assegurar la comprensió del missatge.

3. **Prevenció d'errors:** la nostra aplicació interactuarà amb l'usuari principalment a través de formularis. Previndrem els possibles errors que l'usuari pugui cometre alhora d'introduir valors en el formulari. L'ús de selectors o de barres de desplaçament adequades a cada cas ens poden ajudar en aquesta funció. Per exemple, millor utilitzar un selector de color que dir-li a l'usuari que introdueixi manualment el color en format HTML.
4. **Adaptació al tipus d'usuari:** l'usuari de la nostra aplicació pot tenir diferents necessitats. Per una banda, ens podem trobar a un usuari que necessita temps i vol editar els subtítols i els seus paràmetres i, per l'altra, l'usuari que només vol fer una conversió ràpida de format. Per tant, necessitem que la nostra aplicació tingui la capacitat d'adaptar-se a les dues situacions. Per aquesta, s'intenta reduir al mínim el nombre de nivells de la nostra aplicació.
5. **Web en un o dos nivells:** l'aplicació compta bàsicament amb una pàgina home, on es demana a l'usuari que importi un arxiu, i una pàgina principal, on l'edició i l'exportació estaran al mateix nivell. Per tant, la usabilitat respondrà als diferents tipus d'usuari.
6. **Dinamisme del contingut:** exprimir al màxim les possibilitats de les tecnologies existents per tal de crear una web on el contingut es mostri d'una manera dinàmica i atractiva.
7. **Resolució de pantalla:** tindrem en compte que la resolució més utilitzada pels usuaris de tot el món segons [els informes oficials de 2017](#) és la 1366x768, per tant agafarem aquesta com a resolució base. A més, hem de tenir en compte que la majoria dels nostres usuaris tindran

equips pensats per treballar en vídeo i, per tant, comptaran amb pantalles amb resolució FULL HD(1920X1080) o superior.

8. **Disseny responsiu:** el nostre disseny haurà d'intentar adaptar-se al màxim de resolucions possibles dins de les característiques del projecte. En el nostre cas, per la pantalla d'edició de subtítols necessitarem un ample mínim, sobretot si l'usuari vol traduir contingut. No obstant, com ja hem plantejat al punt 7, no creiem que tinguem problemes en aquest punt.
9. **Tipografia:** s'usen tipografies webfont sanserif per facilitar la llegibilitat dels continguts i perquè ens semblen estèticament més atractives.
10. **Feedback:** si hi ha temps, en aquesta versió s'intentarà implementar algun tipus d'espai perquè l'usuari pugui compartir el seu feedback amb els responsable de l'aplicació i amb la resta d'usuaris. Sinó arribem a temps, buscarem una solució més senzilla.
11. **Documentació:** l'aplicació posarà a l'abast de l'usuari algun tipus de manual o de vídeo explicatiu on es mostrin les funcionalitats principals de l'aplicació.
12. **Internacionalització:** l'aplicació només estarà disponible en llengua anglesa, atès que la majoria de la terminologia que s'usarà és molt específica i de caràcter purament tècnic. Molts dels mots tècnics que s'utilitzen són universals i no tenen ni ús ni fàcil traducció en la resta d'idiomes.

2.5.3 Wireframes

Pàgina home, on l'usuari carrega els arxius de subtítols.

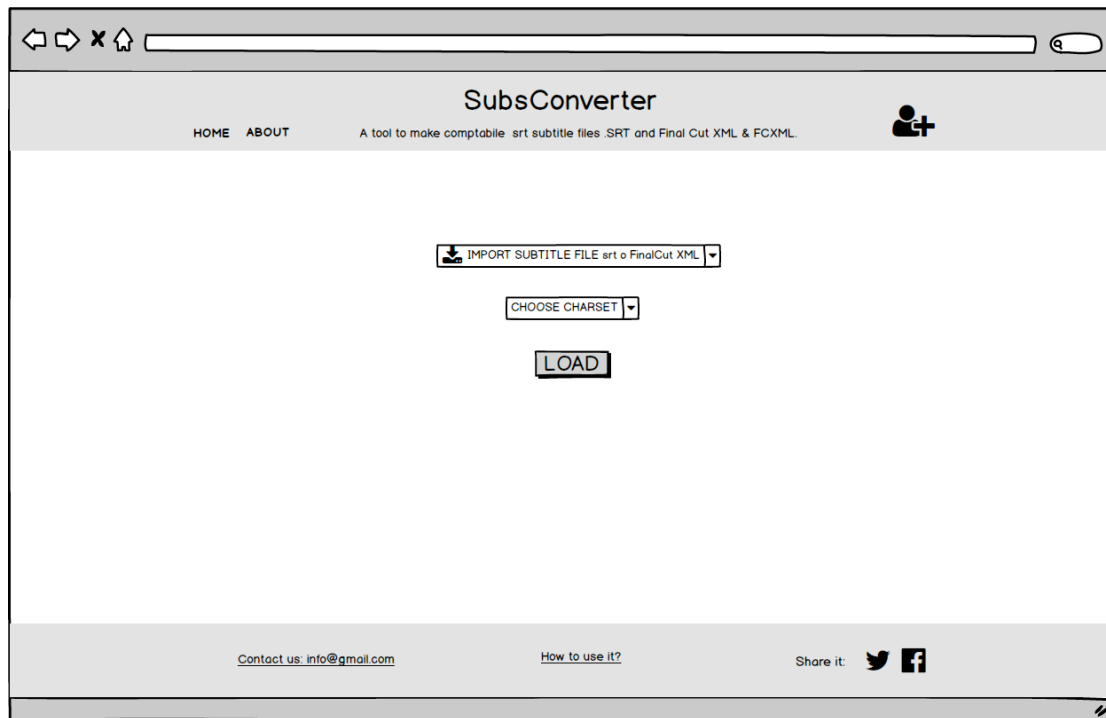


Figura 12. Wireframe: pàgina home.

Un cop l'usuari hagi iniciat la sessió, es mostra un menú amb les opcions de l'usuari.

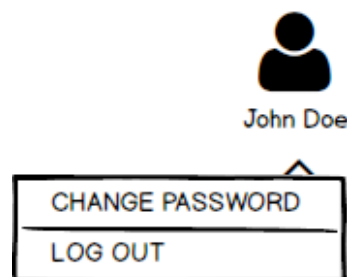


Figura 13. Wireframe: sessió d'usuari.

Pàgina principal amb la pestanya d'edició activada.

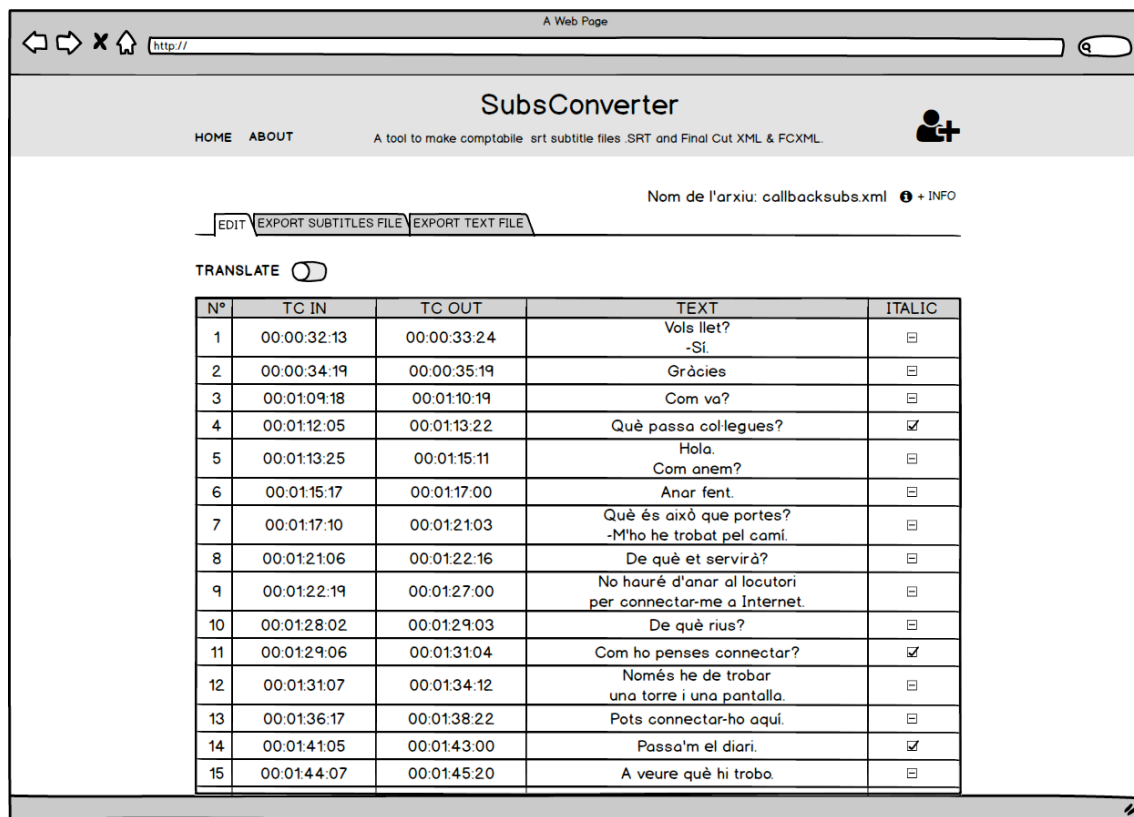


Figura 14. Wireframe: edició de subtítols.

Pàgina principal amb la pestanya d'edició activada i el text traduït

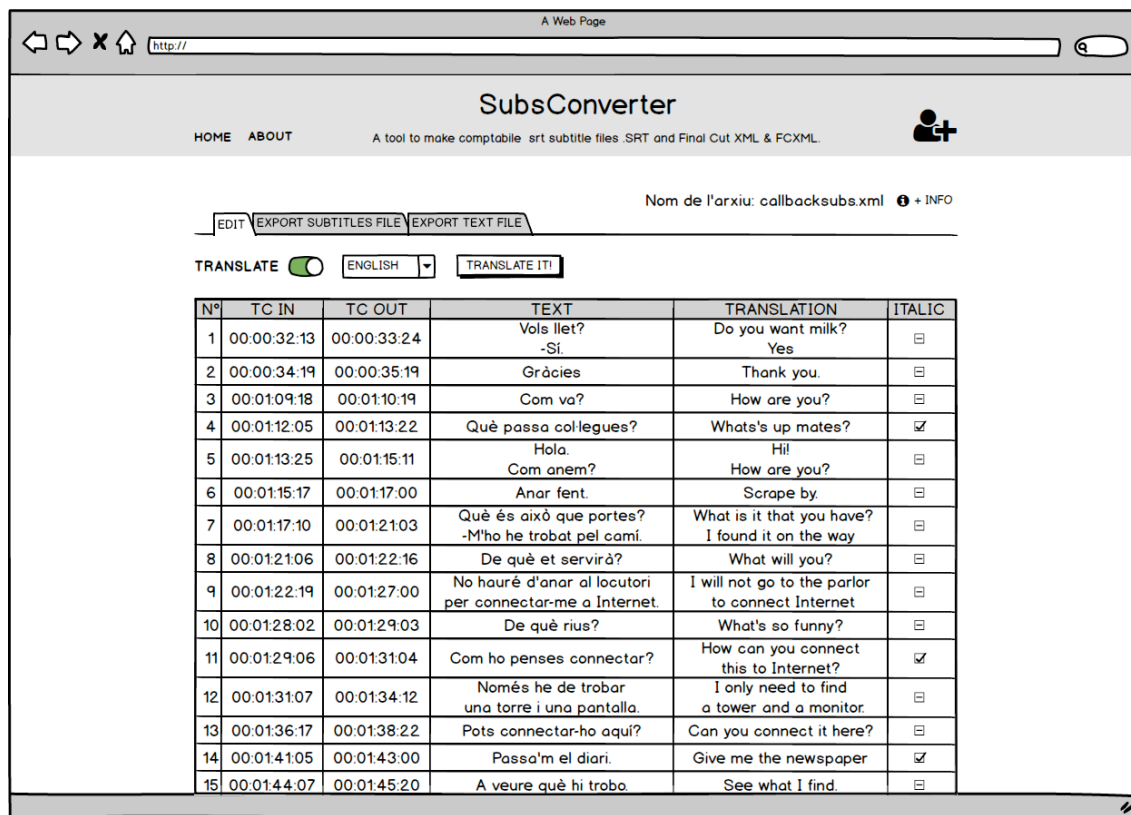
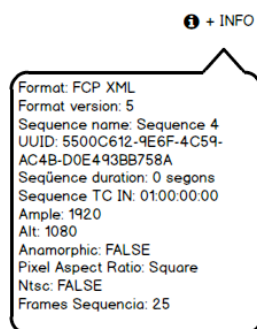


Figura 15. Wireframe: edició de subtítols amb el traductor.

A la part superior dreta, per sota la capçalera, hi ha el nom de l'arxiu. Al costat hi ha una icona de més informació. Si l'usuari clica aquesta icona, el sistema mostrarà més informació sobre l'arxiu. Aquesta icona només apareixerà quan carreguem arxius que continguin informació rellevant com els xml o fcpxml.



La visualització de la informació es farà a través d'una finestra pop, com es veu a la imatge, o d'una animació amb efecte de lliscament. Aquesta segona opció implicaria desplaçar cap avall la taula amb el llistat dels subtítols. Aquesta decisió es prendrà a l'etapa d'implementació.

Pàgina principal amb la pestanya d'exportació a arxiu de subtítols activada

The wireframe shows a web browser window displaying the SubConverter application. The page title is "SubConverter" and the subtitle is "A tool to make compatible srt subtitle files. SRT and Final Cut XML & FCXML." The user is logged in as "John Doe".

The main content area has three tabs: "EDIT", "EXPORT SUBTITLES FILE" (active), and "EXPORT TEXT FILE". The active tab shows the following settings:

- FORMAT:** FCP7 XML
- XML VIDEO SETTINGS:**
 - Reference File: CHOOSE FILE
 - Resolution: 1920x1080
 - Framerate: 23.97
 - USE AUTOMATIC FILE: ☒
- STYLE SETTINGS:**
 - LOAD STYLE SETTINGS: [button]
 - TEXT SETTINGS:
 - Typography: Arial
 - Typography size: 18
 - Style(affects all subs): As edit
 - Text Color: [color picker]
 - CENTER:
 - Choose coordinates: The same for one and two line subs
 - X: [input] Y: [input]

At the bottom of the main content area, there is a red note: "(S'AFFEIXEN TANTS TIPUS DE PREFERÈNCIES COM EL FORMAT HO PERMETI)". Below this is an "EXPORT FILE" button.

On the right side, there is a sidebar with a "LOAD STYLE SETTINGS" dropdown menu. The menu is open, showing a cascading list of style settings:

- SYSTEM
 - SD 480
 - SD 576
 - HD 720
 - HD 1080
 - 2K
 - 4K
 - 5K
- USER
 - My Favourite1
 - My Favourite2 (checked)
 - My Favourite3
- SAVE
- DELETE

Figura 16. Wireframe: pàgina d'exportació de subtítols.

A la dreta de la imatge, veiem com és el menú en cascada que apareix quan l'usuari intenta seleccionar una configuració preestablerta.

En el cas de l'edició de paràmetres, el sistema interactuarà amb l'usuari a través de finestra modal. A continuació, veiem dos exemples: un, el sistema demana a l'usuari el nom d'una nova configuració i, dos, com el sistema avisa a l'usuari de que el nom introduït ja existeix.

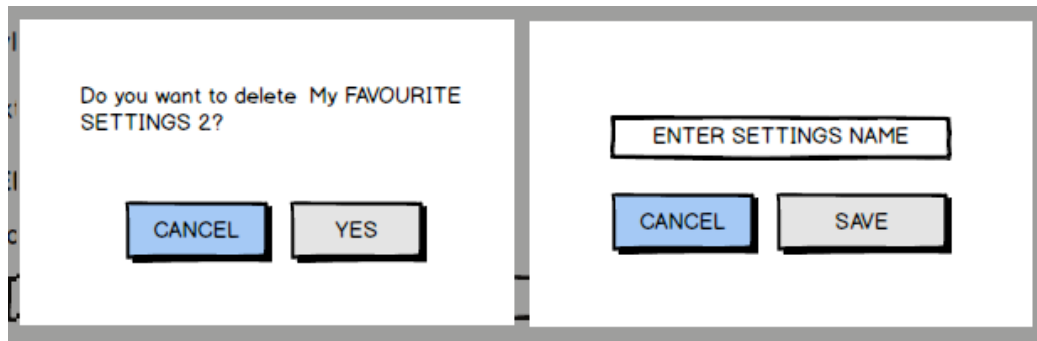


Figura 17. Wireframe: finestres modals.

Pàgina principal amb la pestanya d'exportació a arxiu de text activada

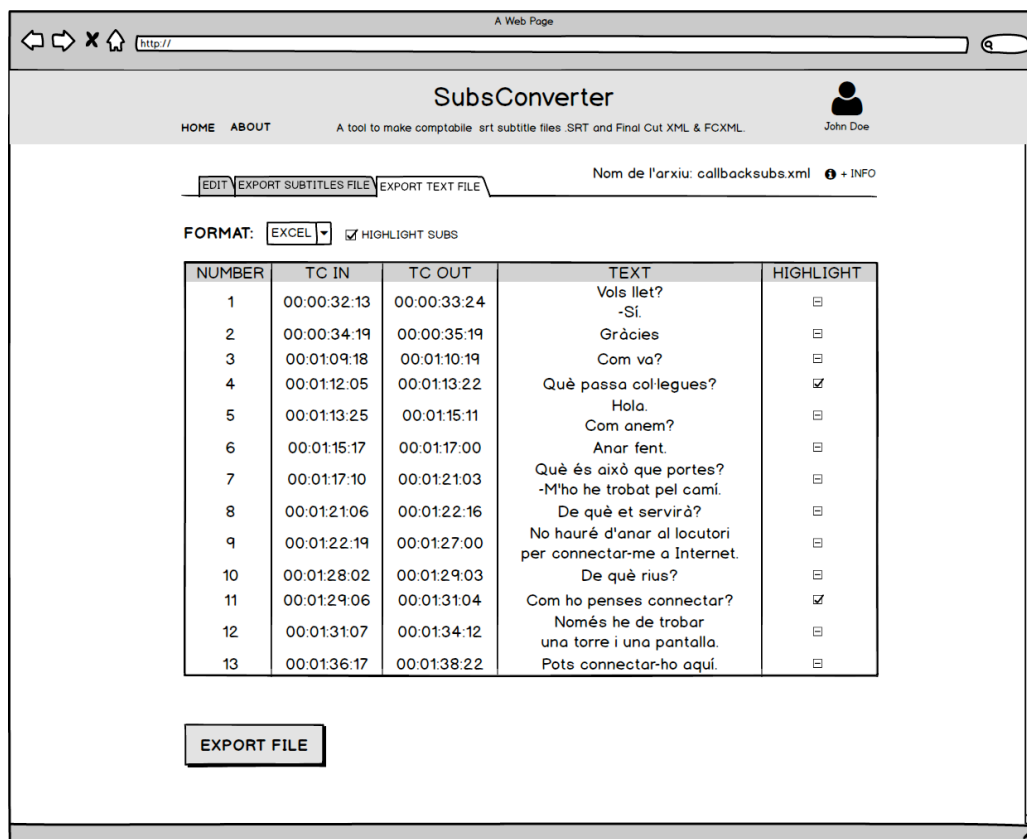


Figura 18. Wireframe: pàgina d'exportació de text.

Pàgina de recuperació de la clau

The wireframe shows a web browser window titled "A Web Page" with a search bar. The page header for "SubsConverter" includes links for "HOME" and "ABOUT", a description "A tool to make comptabile srt subtitle files. SRT and Final Cut XML & FCXML.", and a user icon. The main content area features the heading "FORGOT YOUR PASSWORD?" followed by an "ENTER AN EMAIL" input field and "CANCEL" and "RESET PASSWORD" buttons. The footer contains contact information "Contact us: info@gmail.com", a link "How to use it?", and social media icons for Twitter and Facebook.

A Web Page

SubsConverter

HOME ABOUT A tool to make comptabile srt subtitle files. SRT and Final Cut XML & FCXML.

FORGOT YOUR PASSWORD?

ENTER AN EMAIL

CANCEL RESET PASSWORD

Contact us: info@gmail.com How to use it? Share it: Twitter Facebook

Figura 19. Wireframe: pàgina de recuperació de la clau.

2.5.4 Wall concept

Per descriure l'aspecte gràfic de l'aplicació, es crea un marc de referència visual (Wall Concept) per ajudar a definir la gamma cromàtica i els referents gràfics en que es vol treballar. En el nostre cas es treballa amb un gamma monocromàtica on hi ha un color predominat, en aquest cas un color càlid proper al vermell. També, es valoraran altres colors com alguna tonalitat de groc i vermell que ens ajudin a ressaltar algun element gràfic.



3. Desenvolupament

En aquest apartat intentarem simplificar els punts claus del desenvolupament de la nostra aplicació. Symfony és una tecnologia que utilitza el patró MVC (Model, Vista i Controlador), per tant, intentarem explicar com hem realitzat la implementació del nostre projecte a través de la divisió d'aquests tres conceptes.

3.1 El controlador

Com ja hem comentat en apartats anteriors, el controlador serveix de frontissa entre la vista i el model. Recull les peticions de l'usuari, les envia al model i espera una resposta que retorna a la vista.

A Symfony, els controladors s'emmagatzemen dins de la carpeta Controller que hi ha dins de cada Bundle. En el nostre cas, com que només tenim un bundle, els podem trobar tots a `src/AppBundle/Controller`.

3.1.1 El Routing

A Symfony, el Routing és el sistema que ens permet associar les URL i els controladors. Cada ruta apunta a una acció que s'executa dins d'un controlador. Aquest pot emmagatzemar tantes accions com considerem convenient.

Per definir les rutes a Symfony hi ha diferents metodologies. Aquestes són les dues que hem considerat més interessants

- Definir-les totes dins de l'arxiu `app/config/routing.yml`. L'avantatge és que estan totes juntes en un sol espai i és més fàcil localitzar-les, sobretot si s'està treballant en un projecte molt gran.

- Definir-les dins del controlador a sobre de cada acció amb la tecnologia Annotation. Els elements bàsics d'una ruta són el nom, que ens permetrà cridar-la des de la vista i la seva URL. A partir d'aquí, es poden afegir altres anotacions com els tipus de petició, paràmetres, valors per defecte dels paràmetres, etc.

```
class DefaultController extends Controller
{
    /**
     * @Route("/", name="portada")
     */
    public function portadaAction()
    {
    }
}
```

Es va triar la segona opció perquè va ser la primera que vam conèixer i implementar. A més, el fet que aquesta opció comparteixi espai amb el controlador, ens ajuda a veure els errors de nomenclatura que podem cometre amb més facilitat. A més, atès que la mida del nostre projecte no és massa gran, les anotacions no suposen cap inconvenient. No obstant, si tornéssim a realitzar el projecte aconsellàriem l'ús de la primera opció.

Una idea molt interessant que implementa el sistema de Routing de Symfony és que es poden enviar paràmetres dins de les URL's i, aquests es poden utilitzar directament dins de l'aplicació sense haver de recuperar la informació. Els paràmetres que s'envien dins de la ruta i es defineixen entre { } en el Routing.

Per exemple, en la nostra aplicació quan l'usuari selecciona quin format de subtítols vol exportar, el sistema envia la URL `/show/subs/{subsFormat}`, on `subsFormat` és el format d'exportació. El controlador recull la petició i carrega directament la vista que va associada a aquest format utilitzant el paràmetre inclòs en la URL de la vista.

```
/**
 * @Route("/show/subsForm/{subsFormat}",
 *     name="showSubsForm",
 *     options={"expose"=true},
 *     condition="request.isXmlHttpRequest()")
 */
public function showSubsFormAction($subsFormat)
{
    $template = $this->render( 'format/'.$subsFormat.'.Form.html.twig',
        array('paramId' => 4))->getContent();

    return new JsonResponse($template);
}
```

Aquí ens adonem de la potència real de Symfony: aquestes quatre línies ens permeten carregar qualsevol la vista amb els formularis dels nostres arxius de subtítols.

3.1.2 Controladors, rutes i accions

Un cop acabada la implementació, ens hem adonat que una bona pràctica abans de començar un altre projecte seria fer un llistat de quines són les accions que han de fer els controladors i quin nom hi posem a les rutes que hi van associades. D'aquesta manera seria més fàcil definir de forma acurada el nom de les accions i les rutes i evitariem alguns errors de repetició.

El nostre projecte consta de 5 controladors. Cadascun d'ells associats a una part rellevant de l'aplicació:

- **IndexController:** bàsicament s'encarrega gestionar la creació i validació del formulari de pujada d'arxiu de la pàgina home. Un cop validat el formulari carrega la pàgina principal.
- **EditorController:** s'encarrega de tot allò referent a l'editor de subtítols. En aquest cas, no té massa feina i només implementa l'acció que permet la traducció dels subtítols.
- **ExportSubsController:** controla tot allò referent a la visualització

i exportació dels arxius de subtítols.

- **ExportTextController:** controla tot allò referent a la visualització i exportació dels arxius de text.
- **ParameterController:** controla la visualització, creació i edició dels paràmetres d'estil.

Controlador	Acció	URL	Finalitat
Index	upload	/	Crea la vista del formulari de la pàgina home. Recull les dades, les envia la model i crida a la vista de l'editor.
	about	/about	Gestiona la càrrega de la pàgina about.
Editor	ajaxTranslate (ajax)	/translate	Gestiona les peticions de traducció.
ExportSubs	showSubsForm (ajax)	/show/subs/{subsFormat}	Controla les vistes dels formularis d'exportació d'arxius de subtítols.
	exportSubs	/export/subs/{subsFormat}	Recull la informació dels subtítols i les envia al model per la creació del nou arxiu de subtítols.
ExportText	exportText	/export/text/{textFormat}	Recull informació dels subtítols i l'envia al model per la creació del nou arxiu de text.
Parameter	ParameterList	/parameterList	Controla el llistat de paràmetres d'estil que hi ha al select.
	LoadParameterListContent (ajax)	/parameterList/{format}/{paramId}	Controla les vistes un cop seleccionat un paràmetre d'estil.
	existParam (ajax)	/exist/param	Comprova si existeix el nom d'un paràmetre d'estil.
	addParameter (ajax)	/parameterList/add/{format}/{userId}/{paramId}	Afegir nous paràmetres.
	deleteParameter (ajax)	/parameterList/delete/{format}/{userId}/{paramId}	Esborrar paràmetres.
	updateParameter (ajax)	/parameterList/update/{format}/{userId}/{paramId}	Actualitzar paràmetres.

Com es pot veure destaquem totes aquelles accions que es fan mitjançant peticions Ajax. Com veurem més endavant, aquesta tecnologia ha estat fonamental en la implementació del nostre projecte.

3.1.3 Els formularis

Els formularis són uns dels puntals del nostre projecte. En aquest aspecte Symfony conté la llibreria Form que facilita la seva creació. No obstant, un dels problemes que hem hagut de solucionar durant el període d'implementació és que els formularis de la nostra aplicació són de diferents naturaleses i Symfony no s'adapta igual de bé a totes elles. Això ens ha obligat a trobar diferents mètodes per implementar-los.

- **Formulari associat a una entitat:** aquest és el tipus de formulari ideal quan es treballa amb Symfony perquè permet crear-lo de manera molt senzilla des del controlador i associar-lo a una Entitat(classe que està representada a la base de dades). D'aquesta manera, es simplifica molt la validació, recuperació i emmagatzematge de les dades. En aquests tipus de formularis, la informació de validació es defineix dins de l'atribut de cada entitat gràcies a la tecnologia Annotation. En la nostra aplicació aquest tipus de formularis són els que hem utilitzat pel registre d'usuaris. Per crear-los, s'han aprofitat les plantilles que venien amb el bundle FOSUserBundle.
- **Formulari no associat a cap entitat:** Symfony també permet crear formularis que no estiguin associats a cap entitat. El procés de creació d'aquests és molt similar a l'anterior, tot i que, les normes de validació s'ha de definir dins del controlador. En aquest cas, posarem com exemple la implementació del nostre formulari de pujada d'arxius i, de passada, veurem com el controlador gestiona la rebuda d'informació.

```

public function uploadAction(Request $request){

    // CREEM EL FORMULARI DINS DE LA VARIABLE $FORM.
    // CADA MÈTODE ADD() AFEGEIX UN CAMP ON DEFINIM EL TIPUS, EL
    // CONTINGUT, EL TEXT DEL SEU LABEL I LES SEVES NORMES DE VALIDACIÓ
    // (CONSTRAINTS).
    // EN AQUEST CAS HEM CREAT EL FORMULARI DINS DEL CONTROLADOR PERÒ
    // L'HAGUÉSSIM POGUT CREAR EN UNA CLASSE A BANDA I IMPORTAR-LO.

    $form = $this->createFormBuilder()
        ->add('subtitleFile', FileType::class, array(
            'constraints' => new NotBlank,
            'label' => 'Choose a Subtitle File (.srt, .xml, .fcpxml)'))
        ->add('charset', ChoiceType::class, array(
            'choices' => array(
                'SELECT CHARSET' => 0,
                'ISO-8859-1 (Western Europe)' => 'ISO-8859-1',
                'ISO-8859-2 (Central Europe)' => 'ISO-8859-2'),
            //... -- OMETEM PART DEL CODI PER REITERATIU --

            'choices_as_values' => true,
            'constraints' => new NotEqualTo('0'))))
        ->add('frameRate', ChoiceType::class, array(
            'choices' => array(
                'SELECT FRAME RATE' => 0,
                '23.98' => '23.98',
                '24' => '24',
                '25' => '25',
                '29.97' => '29.97',
                '30' => '30',
            ),
            'data' => '24',
            'choices_as_values' => true,
            'constraints' => new NotEqualTo('0'),
        ))
        ->add('send', SubmitType::class)
        ->getForm();

    // EL MÈTODE HANDLEREQUEST () ENS PERMET RECOLLIR, VALIDAR
    // LES DADES I FER SUBMIT
    $form->handleRequest($request);

    // SI EL FORMULARI ES VÀLID I HA FET SUBMIT
    if ($form->isSubmitted() && $form->isValid()) {

        // GUARDEM LA INFORMACIÓ DEL FORMULARI EN VARIABLES.
        // RECUPEREM L'ARXIU DE SUBTÍTOLS QUE HA PUJAT L'USUARI I
        // EXTREIEM EL SEU NOM ORIGINAL I LA SEVA EXTENSIÓ

        $file = $form['subtitleFile']->getData();
        $name = $file->getClientOriginalName();
        $extension = $file->getClientOriginalExtension();

        // COMPROVEM L'EXTENSIÓ PER SABER QUIN ÉS EL FORMAT I
        // LLAVORS CREEM UN NOU OBJECTE QUE ENS PERMETRÀ LLEGIR
        // LA INFORMACIÓ QUE CONTÉ L'ARXIU. PER ACABAR RENDERITZEM
        // LA VISTA CORRESPONENT AMB LA INFORMACIÓ OBTINGUDA
    }
}

```

```

switch ($extension) {
    case 'srt':

        $charset = $form['charset']->getData();
        $frameRate = $form['frameRate']->getData();
        // CREEM L'OBJECTE
        $srt = new FormatSRT($file, $name, $extension);

        // CARREGUEM L'ARXIU
        $srt->loadFile();
        $srt->setCharset($charset);
        $srt->setFrameRate($frameRate);

        // OBTENIM EL LLISTAT DELS SUBTÍTOLS
        $subtitols = $srt->getSubtitlesList();
        // RENDERITZEM LA VISTA
        return $this->render('subsInfo/srtInfo.html.twig',
array('subtitles' => $subtitols, 'fileName' => $srt->getName(),
'charset' => $srt->getCharset(), 'extension'=>$srt->getExtension(),
'frameRate'=>$srt->getFrameRate(), 'title' => 'Homepage'
));
        // EN EL CAS DELS ARXIS XML FAREM EL MATEIX PROCÉS
        // PERÒ HAUREM DE RECUPERAR ELS PARÀMETRES DE VÍDEO
    case 'xml':

        $xml = new FormatFCP7($file, $name, $extension);
        //... -- OMETEM PART DEL CODI PER REITERATIU --
        // RECUPERAR ELS PARÀMETRES DE VÍDEO
        $videoParam = $xml->getVideoParams();
        return $this->render('subsInfo/fcpInfo.html.twig',
array('subtitles' => $subtitols, 'fileName' => $xml->getName(),
'charset' => $xml->getCharset(), 'extension'=>$xml->getExtension(),
'version'=>$xml->getVersion(), 'frameRate'=>$xml-
>getFrameRate(), 'videoParam'=>$videoParam,
));

    case 'fcpxml':

        //... -- OMETEM PART DEL CODI PER REITERATIU --
        }
    } else {
        // SI EL FORMULARI FALLA, TORNEM A MOSTRAR EL FORMULARI
        return $this->render('upLoadFile.html.twig', array(
            'form' => $form->createView()
        ));
    }
    // SI NO S'HA FET SUBMIT, MOSTREM EL FORMULARI.
    return $this->render('upLoadFile.html.twig', array(
        'form' => $form->createView(),
    ));
}

```

- **Formulari compost per entitats i per no entitats:** aquest és el cas del nostre formulari principal, que és el de la finestra d'edició. Aquest és un formulari complex i que va mutant segons els formats d'exportació que tria l'usuari. El formulari ha de recollir tota la informació sobre l'edició dels subtítols i tots els paràmetres del format d'exportació per crear un nou arxiu de subtítols.

Després d'intentar implementar alguna solució utilitzant la llibreria Form de Symfony sense massa èxit, es va optar per implementar-lo sense aquesta llibreria.

Per fer-ho, es creen els camps de formulari a la vista, es validen les dades al costat del client, s'envia la informació al controlador HTTP(POST), es recuperen les dades, s'analitza quin el format d'exportació i, finalment, s'envia la informació al model perquè aquest generi un nou arxiu.

3.2 La vista

A Symfony les vistes es guarden per defecte a la carpeta `app/resources/views`. En la nostra aplicació hem guardat les vistes principals en el root de la carpeta i les secundaries dins d'altres carpetes.

- Principals:
 - **base:** mostra la informació comuna a totes les pàgines.
 - **editor:** mostra el contingut de l'editor.
 - **uploadFile:** mostra el contingut de la pàgina principal.
 - **about:** mostra la informació de la pàgina about.
- Secundaries:

- **exportPdf**: vistes que es creen per l'exportació dels pdf's.
- **format**: aquestes vistes inclouen els elements del formulari dels formats de subtítols.
- **includes**: vistes que inclouen les etiquetes <option> dels selectores d'idiomes, resolució de vídeo, tipografies, etc..
- **parameter**: vistes que creen el llistat de paràmetres d'estil que l'usuari té disponibles.
- **subsInfo**: vistes amb la informació de cada arxiu de subtítols.

3.2.1 Twig i les plantilles

L'ús de la tecnologia Twig i de les plantilles permet subdividir la vista en blocs de codi HTML que són redundants. D'aquesta manera el codi es reutilitza i es facilita qualsevol modificació. A aquelles vistes que es reutilitzen les anomenem plantilles.

Twig disposa de diferents mètodes per relacionar les plantilles entre si.

- **Extends**: cridem a la vista pare des de la vista filla. Si ens fixem en l'exemple de l'apartat anterior, podem analitzar com es carreguen les vistes en la pantalla d'edició.

Un cop el controlador ha recuperat tota la informació de l'arxiu, ja pot renderitzar la vista per crear la pàgina d'edició. Aquesta ens mostrarà la informació de l'arxiu, el llistat del subtítols i les opcions d'exportació.

```
return $this->render('subsInfo/fcpInfo.html.twig',
    array('subtitles' => $subtitols,
          'fileName' => $xml->getName(),
          'charset' => $xml->getCharset(),
          'extension'=>$xml->getExtension(),
          'version'=>$xml->getVersion(),
          'frameRate'=>$xml->getFrameRate(),
          'videoParam'=>$videoParam
    ));
```


Com podem veure, el primer que fa el controlador és renderitzar la vista que mostra la informació de l'arxiu de subtítols `subsInfo/fcpInfo.html.twig`, atès que per cada format es mostra una informació diferent. Aquesta, alhora, crida la plantilla `editor.html.twig`, que mostrarà tot el contingut restant de l'editor i aquesta, fa una última crida a `base.html.twig`, que és la plantilla que conté la capçalera del nostre projecte i que és la base de totes les pàgines. Per tant, direm que la pàgina editor està formada per tres vistes diferents. Si volem fer un `extends`, haurem de col·locar la comanda següent a l'inici de cada vista:

```
{% extends 'base.html.twig' %}
```

- **Include:** inclou una vista dins d'una altra. En la vista de l'editor, `editor.html.twig`, utilitzem diferents `includes`. Un d'ells és `optionLang.html.twig`, que carrega les més de 100 etiquetes `<option>` que formen part dels selectors(`<select>`) que permeten a l'usuari escollir els l'idioma per la traducció de subtítols. Aquest `include` el podem utilitzar cada vegada que el necessitem i redueix la llargada del codi de la vista que l'inclou, cosa que facilita la programació.

```
{% include 'includes/optionLang.html.twig' %}
```

- **Render/controller:** crida a un controlador i ens mostra la seva vista renderitzada. En la nostra aplicació, l'hem utilitzat per mostrar el llistat de paràmetres d'estil que l'usuari té disponibles. El llistat d'aquests paràmetres depèn de si l'usuari ha iniciat la sessió. Si l'ha iniciat, accedirem a la base de dades per mostrar els llistat per defecte del sistema i els que l'usuari té emmagatzemats, sinó només mostrarem els del sistema. La vista envia el id de l'usuari(si s'escau), el format i l'identificador del paràmetre escollit.

```

{% if is_granted('ROLE_USER') %}
    {{ render(controller('AppBundle:Parameter:ParameterList', {
        'userId': app.user.id, 'format': 'fcp7', 'selected':
        param.id })) }}
{% else %}
    {{ render(controller('AppBundle:Parameter:ParameterList',
        { 'format': 'fcp7', 'selected': param.id })) }}
{% endif %}

```

Si volem saber quines vistes s'han carregat al crear una pàgina, podem accedir a l'apartat Twig dins de la consola de Symfony. Malauradament, la consola no ens dona la informació de les vistes carregades a través de la tecnologia Ajax, però és de gran ajuda per comprovar quines són les vistes que conformen la nostra pàgina.

Rendered Templates

Template Name	Render Count
subsInfo/srtInfo.html.twig	1
editor.html.twig	1
base.html.twig	1
includes/optionLang.html.twig	2
@WebProfiler/Profiler/toolbar_js.html.twig	1
@WebProfiler/Profiler/base_js.html.twig	1

Rendering Call Graph

```

main 839.91ms/100%
└─ subsInfo/srtInfo.html.twig 838.42ms/100%
  └─ editor.html.twig 838.28ms/100%
    └─ base.html.twig 838.14ms/100%
      └─ editor.html.twig::block(title)
      └─ editor.html.twig::block(stylesheets)
        └─ base.html.twig::block(stylesheets)
      └─ editor.html.twig::block(javascripts) 1.71ms/0%
        └─ base.html.twig::block(javascripts)
      └─ editor.html.twig::block(section) 835.45ms/99%
        └─ subsInfo/srtInfo.html.twig::block(fileInfo)
          └─ includes/optionLang.html.twig
          └─ includes/optionLang.html.twig
└─ @WebProfiler/Profiler/toolbar_js.html.twig 1.49ms/0%
  └─ @WebProfiler/Profiler/base_js.html.twig

```

Figura 20. Informació sobre les vistes a la consola de Symfony.

3.2.2 L'ús d'Ajax

La nostra aplicació només té dues pàgines principals: la pàgina home, on pugem els arxius de subtítols, i la pàgina editor, on els podem editar i exportar a diferents formats.

Des del principi, una de les funcionalitats que teníem clares era que una sola pàgina englobés aquestes dues accions, perquè això agilitza molt la feina als usuaris als quals va destinada l'aplicació.

La tecnologia Ajax gràcies a la seva capacitat de modificar les pàgines sense haver d'actualitzar-les ens ha permès que l'edició i l'exportació convisquin en un mateix pla. Podem canviar els paràmetres d'exportació sense perdre la feina d'edició i a l'inrevés. Per analitzar els avantatges i el funcionament d'Ajax, agafarem d'exemple la càrrega de contingut dels paràmetres d'estil.

Quan l'usuari selecciona un paràmetre d'estil, utilitzem JQuery per fer una petició Ajax al controlador. En aquesta petició creem una ruta amb la informació del format i el identificador del paràmetre demanat.

```
/*CHECK STLYE SETTINGS CHANGES - AJAX*/
$(document).on('change', '#styleSettingsSelect', function () {
    var paramId = this.value;
    var format = $("select[name=subsFileType]").val();
    // EL BUNDLE JsROUTING ENS PERMET FER CREAR UNA RUTA PER
    // SYMFONY DES DE JAVASCRIPT
    var URL = Routing.generate('parameterListContentAjax', {
        format: format, paramId: paramId });
    $.ajax({
        url: URL,
        dataType: 'json',
        success: function(json) {
            // ESBORREM EL CONTINGUT DEL CONTENIDOR I CARREGUEM EL
            // NOU
            $('#styleSettingsContainer').empty().html(json);
            // INICIALIZEM LES FUNCIONES JQEURY QUE CONTROLLEN
            // ELS SELECTORS DE COLORS I EL SELECTOR DE COORDENADES
            coordinates();
            initialiseSpectrum();
        },
        error : function(jqXHR, status, error) {
```

```

        alert('¡Api Error! Plese, try later.');
```

Per l'altra banda, el controlador rep la sol·licitud, comprova quin format sol·licitat per l'usuari, busca la informació a la base de dades i retorna la vista demanada.

```

// IMPORTANT UTILITZAR LA CLASSE JSONRESPONSE
use Symfony\Component\HttpFoundation\JsonResponse;

// IMPORTANT DEFINIR LA CONDICIÓN "request.isXmlHttpRequest()"
/**
 *@Route("/parameterList/Content",
 *name="parameterListContentAjax",
 *options={"expose"=true},
 *condition="request.isXmlHttpRequest()")
 */
public function LoadParameterListContentAjaxAction(Request $request,
$format, $paramId)
{
    switch ($format) {
        case 'fcp7':

            $em = $this->getDoctrine()->getManager();
            $param = $em->find('AppBundle:FCP7Param', $paramId);

            // IMPORTANT UTILITZAR EL MÈTODE getContent() ALHORA
            // DE RENDERITZAR LA VISTA
            $template = $this->render(
                'format/'.$format.'StyleSettings.html.twig',
                array('param' => $param))->getContent();

            return new JsonResponse($template);

        case:''...
            # code...
            break;
    }
}

```

3.2.3 Programaris de tercers a la capa vista

En aquest apartat volem destacar altres programaris de tercers que s'han utilitzat per millorar la vista.

- **Lightbox:** plugin de JQuery que hem utilitzat per mostrar imatges en una finestra modal. La seva integració és molt senzilla, només cal enllaçar els arxius Javascript i CSS amb la nostra pàgina i afegir l'atribut `data-lightbox="nom"`, amb un nom que l'identifiqui, dins de les etiquetes que contenen les imatges que volem visualitzar. Si en volem visualitzar varies en una sola finestra modal, com és el nostre cas, el nom ha de ser el mateix per totes les imatges.

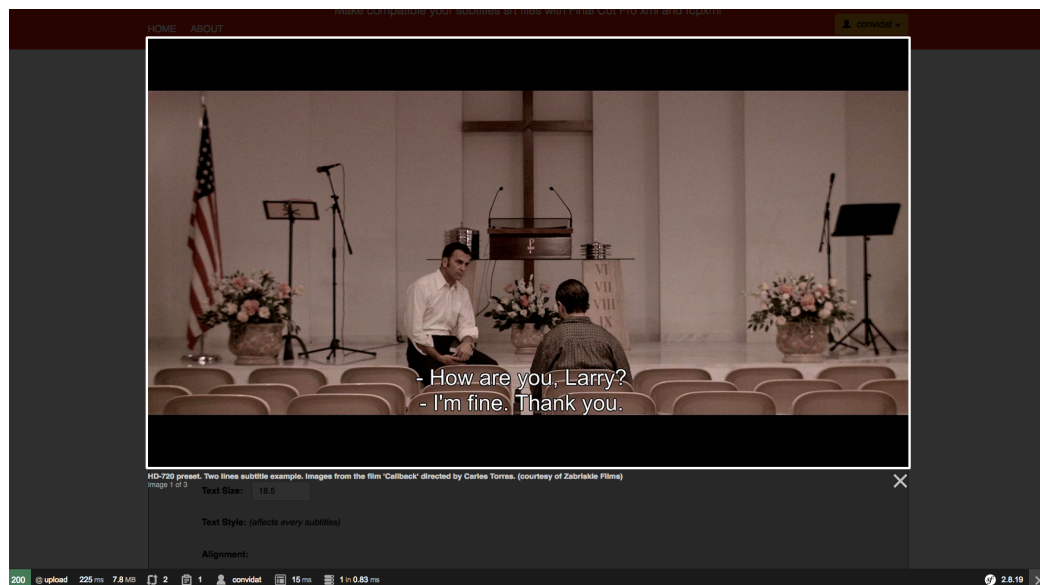


Figura 21. Funcionament de Lightbox, plugin per JQuery.

- **Spectrum:** és un plugin JQuery que permet crear selectors de colors a mida. L'ús d'aquest plugin és indispensable atès que el `<input type="color">` disponible per Html5 no és compatible amb tots els navegadors.

Spectrum és altament configurable i proporciona moltes opcions de visualització. Nosaltres hem creat una model força complet que permeti a l'usuari triar entre diferents mètodes de selecció: colors predefinits, introducció manual o selecció del color a la paleta.

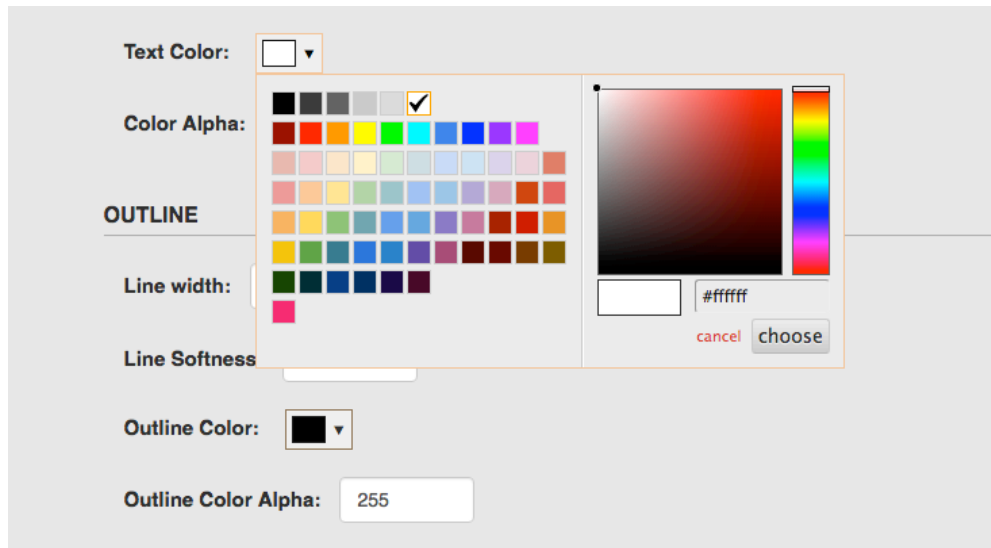


Figura 22. Funcionament d'Spectrum, plugin per JQuery.

3.3 El Model

La nostra capa model es divideix en tres grans blocs.

- **Les entitats:** són objectes PHP utilitzats per gestionar la informació de la base de dades.
- **El convertidor:** és un conjunt de classes de creació pròpia que ens permeten llegir la informació dels arxius de subtítols i crear-ne de nous segons el seu format. Aquest conjunt de classes les trobem a la carpeta `src/AppBundle/Model/SubsFormat`.

- **Mòduls i programaris de tercers:** bundles o classes PHP creades per tercers que hem utilitzat en el model de la nostra aplicació. Es troben a la carpeta `/vendor`.

3.3.1 Doctrine: la base de dades i les entitats

Les aplicacions Symfony no gestionen la seva informació accedint directament a la base de dades. Creen, modifiquen i esborren la informació mitjançant objectes PHP anomenats Entitats. Això és possible gràcies a unes llibreries externes anomenades ORM o Object-Relational Mapping. Symfony ha adaptat la solució Doctrine per aplicar el ORM al seu sistema.

En conseqüència, l'ús d'aquestes classes és ideal per aquelles entitats que tenen una representació a la base de dades. En el nostre cas, ho hem utilitzat per emmagatzemar la informació dels paràmetres i dels usuaris. En el cas dels usuaris, ho hem implementat a través de FOSUser Bundle com expliquem més endavant.

- **Creació de les entitats i de les taules**

Les entitats es defineixen mitjançant classes PHP dins de la carpeta `src/AppBundle/Entity` del sistema. Doctrine no obliga que aquestes classes heretin d'una altra classe especial, sinó que es poden crear com si fossin una classe pare.

Per vincular-les amb Doctrine cal importar la classe ORM que permet fer el mapatge i usar les anotacions pertinents en la definició de la taula i dels atributs. Mostrem la capçalera i els primer atributs de la nostra entitat `FCP7Param`.

```
use Doctrine\ORM\Mapping as ORM;

/**
 * FCP7Param
 *
 * @ORM\Table(name="fcp7param")
 *
 * @ORM\Entity(repositoryClass="AppBundle\Repository\FCP7ParamRepository")
 */
class FCP7Param
{
    /**
     * @var int
     *
     * @ORM\Column(name="id", type="integer")
     * @ORM\Id
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    private $id;

    /**
     * @var string
     *
     * @ORM\Column(name="name", type="string", length=255)
     */
    private $name;
```

Un detall interessant és que Doctrine permet afegir atributs i mètodes que no vulguem desar a la base de dades. Només cal treure l'anotació `@ORM` de l'atribut en qüestió.

Per definir una relació de cardinalitat 1:N com la que hi ha entre els paràmetres de l'aplicació i els usuaris s'utilitza l'anotació `ManyToOne` amb la ruta per accedir a l'entitat que relacionem.

```
/** @ORM\ManyToOne(targetEntity="AppBundle\Entity\User") */
protected $user;
```

Per altra banda, la creació de classes sempre ha estat un procés força pesat. Per aquesta raó Symfony inclou una sèrie de comandaments per gestionar la entitats, la creació i actualització de les taules, i la visualització de les sentències SQL que s'executaran


```
// Creació d'entitats
php app/console generate:doctrine:entities

// Creació de taules
php app/console doctrine:schema:create

// Actualització de taules
php app/console doctrine:schema:update -force

// Visualització de les setències SQLQue s'executaran
php app/console doctrine:schema:update --dump-sql
```

- **Accés a la base de dades**

Per accedir a la base de dades des del controlador haurem d'inicialitzar l'objecte Entity Manager.

```
$em = $this->getDoctrine()->getManager();
```

Un cop instanciat ja podem consultar i editar les taules de la base de dades

- **Consultar a la base de dades**

Symfony implementa diferents mètodes per fer consultes bàsiques que simplifiquen les sentències SQL. Amb el mètode `getRepository()` accedim a la taula que volem consultar i amb els mètodes `find()`, `findAll()`, `findBy()` i `findOneBy()` enviem els arguments que estem cercant.

Per consultes més complexes, podem utilitzar el mètode `createQuery()` que es permet fer consultes SQL tipus `SELECT`, `UPDATE` o `DELETE`. Aquestes consultes es poden crear directament al controlador o en un mètode dins del repositori de l'entitat, per després ser cridat des del controlador.

En el nostre cas, amb els mètodes bàsics n'hem tingut prou. A continuació, veiem la comanada que es permet trobar els paràmetres que un usuari té emmagatzemats a la base de dades.

```
$userOptions = $em->getRepository('AppBundle:FCP7Param')  
->findBy(array('user' => $userId));
```

Per guardar la informació a la base de dades s'utilitzen dos mètodes. El primer `persist()`, que prepara la informació i `flush()` que l'executa. Per veure un exemple de funcionament, veiem com creem un objecte `FCP7param` i el guardem a la base de dades en el controlador de paràmetres.

```
$param = new FCP7Param();  
$param->setName($name);  
$param->setTypo($params['typo']);  
$param->setTextSize($params['textSize']);  
$param->setTextStyle($params['textStyle']);  
$param->setAlignment($params['alignment']);  
$param->setTracking($params['tracking']);  
//... Obviem la resta de setters  
  
$em = $this->getDoctrine()->getManager();  
$em->persist($param);  
$flush = $em->flush();
```

Per inserir un objecte usarem `persist($objecte)`, per actualitzar-lo `persist()` sense cap paràmetre i per esborrar-lo `remove($objecte)`.

3.3.2 Estructura de classes del convertidor

En l'estructura de classes del nostre convertidor ha de permetre llegir, crear i editar els diferents formats de subtítols. Un dels principis del programari que ha de respectar és el principi d'obert-tancat, atès que volem que en un futur el nostre sistema creixi i es vagin afegint nous formats de subtítols.

Per aconseguir això, utilitzarem l'abstracció, l'herència i el poliformisme. Totes les classe que defineixen els formats de subtítols seran filles de la classe para Format.

Format és un classe abstracte, de la qual no podem crear-ne cap instància, però que marcarà les propietats i els mètodes comuns de tots els formats de subtítols.

```
abstract class Format {  
  
    protected $file;  
    protected $name;  
    protected $extension;  
    protected $charset;  
    protected $frameRate;  
  
    protected $subtitlesList;  
  
    public function __construct($file=null) {  
        $this->file = $file;  
        if ($file !== null) $this->loadFile($this->file);  
    }  
  
    abstract public function loadFile();  
    abstract public function getSubtitlesList();  
    abstract public function createNewFile();  
  
    // S'obvien tots els mètodes getter & setters  
}
```

Cada classe filla compartirà els atributs de la classe pare i n'afegirà de propis. Per exemple, en el cas dels arxius XML haurem d'afegir els atributs de versió de l'arxiu i llistat de paràmetres d'estil.

En quant els mètodes totes les classes filles hauran de sobreescrivre els tres mètodes abstractes que es defineixen a la classe pare i podran afegir-ne tots aquells que necessitin.

Tot i això, una de les dificultats que ens hem trobat amb PHP respecte a la programació orientada a objectes és que no admet la sobrecàrrega de mètodes. Això ens ha suposat un inconvenient sobretot en la creació de constructors i la sobreescritura dels mètodes abstractes heretats.

Per solucionar-ho, hem optat per l'ús de valors per defecte nuls en els paràmetres dels mètodes.

```
//EXEMPLE DEL CONSTRUCTOR
public function __construct($file=null) {
    $this->file = $file;
    if ($file !== null) $this->loadFile($this->file);
}

// COMPAREM MÈTODE ABSTRACTE DE LA CLASSE PARE I FILLA
// CLASSE PARE
abstract public function createNewFile();
// CLASSE FILLA
public function createNewFile($subtitlesList=null,
    $originalFrameRate=null, $targetFrameRate=null);
```

3.3.3 El codi del convertidor

En aquest apartat analitzarem el codi que hem utilitzat en els tres mètodes principals de cada format de subtítols.

- **loadFile()**: aquest mètode obre l'arxiu de subtítols i el prepara per ser llegit.
 - **SRT**
En el cas del arxius .srt utilitzem el mètode file() que ens retorna un array amb cadascuna de les línies que conté l'arxiu.

```
public function loadFile(){
    $this->fileLines = file($this->file);
}
```

- **XML i FCPXML**

En el cas dels arxius .XML utilitzarem dos mètodes: per una banda, `simplexml_load_file()`, que ens permet interpretar l'arxiu com un objecte. Així, podem accedir a la informació seguint la ruta exacta que hi ha entre els nodes. Aquest mètode l'hem usat per llegir els arxius XML de FCP7.

Per altra banda, també podem usar la biblioteca DOM de php. Aquesta biblioteca implementa dos mètodes que ens faciliten la lectura: `getAttribute()` i `getElementsByTagName()`. Per utilitzar-la haurem de crear un nou objecte `DomDocument` i carregar-hi l'arxiu XML. Aquest mètode l'hem usat per llegir els arxius FCPXML de FCPX i les paràmetres de vídeo dels arxius FCP7 i FCPX.

```
public function loadFile(){
    // Open File
    $root = simplexml_load_file($this->file);
    $this->xml = new \DomDocument();
    $this->xml->loadXML($root->asXML());
}
```

- **getSubtitlesList():** Llegeix la informació de l'arxiu i retorna un array amb tots els objectes Subtítols.

- **SRT**

En el cas dels .srt recorrem línia a línia l'arxiu tenint en compte que la primera línia de cada subtítol és un nombre. Que la següent, és el codi de temps i que la resta és text fins que trobem una línia en blanc. Per tant, per realitzar la lectura hem creat unes constants que defineixen cada tipus de línia i un switch que revisa l'estat de les constants i carrega la part del codi que ens interessa. Un cop capturada la informació de cada subtítol creem un nou objecte Subtítol i tornem a començar.

Un dels inconvenients que ens hem trobat a l'utilitzar aquest sistema és que hi ha molts arxius .srt que no tenen una línia en blanc al final i, per tant, l'script no capturava l'última línia. Per arreglar això, vam afegir la condició de que si ens trobem a l'última línia i aquesta no està buida, emmagatzemem el text dins de la variable corresponent.

Podeu consultar el codi complert a [l'Annex 3. Altres extractes de codi](#).

- **XML i FCPXML**

La lectura en el cas dels arxius XML i FCPXML és molt més simple només cal capturar en una variable tots els nodes que contenen la informació sobre els subtítols i utilitzar el constructor foreach per recorre-la. Per cada element node s'emmagatzema la informació i es crea un objecte subtítol.

En el cas dels arxius XML es comproven tots els nodes generatoritem que hi ha dins de cada node track.

```
$trackNumbers = $this->xmlSimple->sequence->media->
video->track;
foreach ($trackNumbers as $track) {
    foreach ($track->generatoritem as $generatoritem)
    {
        //.....
    }
}
```

Podeu consultar aquest codi complert a [l'Annex 3. Altres extractes de codi](#).

En el cas del FCPXML només cal comprovar tots els nodes title.

```
$titlesTag = $this->xml->getElementsByTagName("title");
foreach($titlesTag as $title)
{
    //.....
}
```

- **createNewFile**: aquest mètode ens permet crear, escriure i tancar un nou arxiu.

- **SRT**

Per crear un arxiu Srt necessitem els paràmetres següents: un array amb els objectes subtítols, la velocitat dels fotogrames de l'arxiu original i la de l'arxiu de sortida. Per realitzar el procés d'escriptura hem utilitzat tres mètodes nadius de PHP:

- **Fopen()**: ens permet crear i obrir un nou arxiu. Quan l'instanciem hem de definir el tipus d'accés. En aquest cas és "W" perquè hi volem escriure.
- **Fwrite()**: aquest mètode ens permet escriure dins l'arxiu. Per crear els salts de línia hem utilitzat la constant PHP_EOL.
- **Fclose()**: ens permet tancar l'arxiu.

Podeu consultar aquest codi complet a [l'Annex 3. Altres extractes de codi](#).

- **XML**

En el cas de l'XML, l'escriptura és un procés molt feixuc i repetitiu atès que cada subtítol consta d'unes 350 línies. La raó és que cada subtítol conté tots els paràmetres d'estil i aquests es defineixen mitjançant l'etiquetatge XML.

Per crear un arxiu XML, al marge dels subtítols i de la velocitat dels fotogrames(frame rate), també necessitarem els paràmetres de vídeo, els paràmetres d'estil i l'arxiu de referència(en el cas que l'usuari en vulgui utilitzar un de propi). El procés consta de quatre fases:

- **Carregar la ruta de l'arxiu XML de referència.**

En aquest cas hem de contemplar dues opcions: que l'usuari pengi un arxiu de referència o que demani al sistema generar-ne un de nou. En ambdós, s'usa el mètode `loadXML()`, com ja fèiem en el procés de lectura.

- **Crear l'etiqueta contenidora principal.**

Creem una nova etiqueta `<track>` que contindrà tots els subtítols. D'aquesta manera apareixeran tots dins de la mateixa pista en l'editor de vídeo.

Per crear la nova etiqueta s'utilitzen dos mètodes d'escriptura que ens proporciona el DOM: per una banda, `createElement(nom, contingut)`, amb el qual creem l'element desitjat i, per l'altra banda, `appendChild(node)`, el qual ens permet inserir-lo dins de l'etiqueta seleccionada.

```
// CREATING THE MAIN NODE TRACK
$track = $file->createElement("track");
$newTrack    =    $file->getElementsByTagName('video')-
>item(0)->appendChild($track);
```


- **Crear cada subtítol.**

Cada subtítol consta de unes 300 etiquetes aproximadament. Per tant, per crear cadascuna d'aquestes etiquetes utilitzarem els mateixos mètodes que amb l'etiqueta `<track>`. Un altre mètode que utilitzarem en aquesta fase serà `setAttribute(nom, contingut)` que ens permetrà crear atributs dins dels nodes. D'aquestes tres-centes etiquetes, veurem un parell d'exemples aleatoris: l'etiqueta `<generatoritem>`, que és el contenidor de cadascun dels subtítols, i l'etiqueta `<start>`, que conté el codi de temps d'inici de cada subtítol i crida a un mètode que el transforma en en format 'hh:mm:ss:ff' en fotogrames.

```
// GENERATOR ITEM
$generatoritem = $file->createElement("generatoritem");
$generatoritem->setAttribute("id", "Outline Text");
$newGeneratoritem = $track->appendChild($generatoritem);

// START
$start= $file->createElement("start", self::tcToFrames
($sub->getTcIn(), $frameRate));
$newStart = $generatoritem->appendChild($start);
```

3.3.4 Mòduls i programari de tercers

- **FOSUserBundle**

Aquest és el bundle de Symfony més important que hem implementat en la nostra aplicació. FOSUserBundle és una llibreria molt completa que proporciona un marc per a la gestió d'usuaris i que té com a objectiu facilitar la implementació de les tasques més habituals. En la nostra aplicació l'hem utilitzat per implementar:

- Registre d'usuari
- Inici de sessió
- Edició del perfil

- Canvi de clau
- Recuperació de la clau

L'estructura de FOSUserBundle(FOSUB) està composta pel model, les vistes i els controladors. La nostra feina ha consistit en entendre com funciona per adaptar-la a la nostra aplicació.

- **Configuració dels paràmetres de seguretat**

Aquesta és un tasca clau abans d'utilitzar el bundle. Aquests paràmetres es defineixen dins l'arxiu `app/config/security.yml`. A continuació veiem la informació clau que cal definir.

```
security:
  encoders:
    // TIPUS D'ENCRIPCIÓ DE LA CLAU
    FOS\UserBundle\Model\UserInterface: bcrypt
  // ROLES DEFINITS A L'APLICACIÓ
  role_hierarchy:
    ROLE_ADMIN: ROLE_USER
  // ROLES DEFINITS A L'APLICACIÓ
  providers:
    // QUIN ÉS L'IDENTIFICADOR DEL NOSTRE USUARI. EN EL
    // NOSTRE CAS PODEM INICIAR LA SESSIÓ AMB EL
    // USERNAME i AMB EL CORREU.
    fos_userbundle:
      id: fos_user.user_provider.username_email
  firewalls:
    main:
      pattern: ^/
      // DEFINICIÓ DELS PARÀMETRES DEL LOGIN
      form_login:
        provider: fos_userbundle
        login_path: fos_user_security_login
        check_path: fos_user_security_check
      // SEGURETAT CSRF
      csrf_token_generator: security.csrf.token_m
      anager
      // ACTIVACIÓ DE REMEMBER ME
      remember_me: true
      default_target_path: /
      logout:
        path: fos_user_security_logout
        target: /
```

```

// AJUSTOS REMEMBER ME
remember_me:
    secret: '%secret%'
    lifetime: 604800 # 1 week in seconds
    anonymous: true
// CONTROL DELS ACCESSOS
access_control:
    -{path: ^/login$, roles:IS_AUTHENTICATED_ANONYMOUSLY }
    -{path: ^/register, roles: IS_AUTHENTICATED_ANONYMOUSLY }
    -{path: ^/resetting, roles: IS_AUTHENTICATED_ANONYMOUSLY }

```

○ La creació de l'entitat User.

Per poder crear la taula usuari a la nostra base de dades, hem de crear una classe filla de la classe BaseUser que implementa per defecte FOSUB. Això, ens permet sobreescrivre i afegir alguns mètodes i atributs sense haver de tocar la classe mare. En el nostre cas, ho hem aprofitat per definir les restriccions de validació dels camps id, password i usuari amb els seus respectius missatges d'error. Com que els hem definit dins de l'entitat afectarà a tots els formularis que utilitzin aquesta classe.

```

/**
 * @ORM\Entity
 * @ORM\Table(name="fos_user")
 */
class User extends BaseUser
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue(strategy="AUTO")
     */
    protected $id;

    /**
     * @Assert\Length(
     *     min=6,
     *     max=255,
     *     minMessage="The username must have at least {{
limit }} characters.",
     *     maxMessage="The password must not have most
than {{ limit }} characters."
     * )
     */
}

```

```
protected $username;

/**
 * @Assert\Length(
 *     min=8,
 *     max=255,
 *     minMessage="The password must have at least {{
limit }} characters.",
 *     maxMessage="The password must not have more
than {{ limit }} characters."
 * )
 */
protected $plainPassword;

public function __construct()
{
    parent::__construct();
    // your own logic
}
}
```

- **Modificació de les vistes**

El bundle inclou unes vistes per defecte, però a nosaltres ens interessa s'adaptar-les al nostre projecte. Per aquesta raó, copiem les vistes del FOSUB a la nostra carpeta de vistes per editar-les i estilitzar-les a la nostra conveniència.

- **Google Translate**

Fa uns anys Google va tancar la seva Api de traducció gratuïta. Actualment, l'Api de Google Translate és de pagament a partir del milió de paraules traduïdes. A Github, hem trobat la classe GoogleTranslate, que ens permet traduir textos a través de l'antiga l'API de Google de manera gratuïta.

Aquesta classe no és massa ràpida però, de moment, ens permet implementar aquesta funcionalitat. L'ús de la classe és molt senzill, només cal que cridem el mètode estàtic translate() amb el text que

volem traduir, l'idioma del text original i l'idioma de traducció. Veiem un exemple del funcionament:

Recuperem la informació que hem de traduir del formulari i fem una petició Ajax al controlador, que cridarà a la classe GoogleTranslate i retornarà la traducció a la vista.

```
//VISTA
$( "#translate" ).click(function() {
    // PER CADA TEXT AREA, FEM UNA CRIDA
    $(".textOriginal").each(function(index) {
        var text = $(this).val();
        var targetLang = $('select[name=targetLang]').val();
        var sourceLang = $('select[name=sourceLang]').val();
        var URL = Routing.generate('ajaxtranslate');
        $.ajax({
            url: URL,
            context: this,
            data: {
                sl: sourceLang,
                tl: targetLang,
                q: text,
            },
            dataType: 'json',
            success: function(json) {
                $('#textTranslate'+(index+1)).val(json.trans);
            },
            error: function(jqXHR, status, error) {
                alert('¡Api Error! Plese, try later.');
```

```
        $text);  
    // GUARDEM EL RESULTAT EN UN ARRAY ASSOCIATIU  
    $arr = ["trans" => $result];  
    // RETORNEM EL JSON  
    return new JsonResponse($arr);  
}
```

- **FOSJsRouting**

Aquest petit bundle permet generar rutes compatibles amb Symfony des de Javascript. A més, permet utilitzar el nom de la ruta tal i com la tenim declarada al nostre controlador i passar-hi paràmetres. Nosaltres l'hem utilitzat per crear les URL's en les peticions Ajax.

```
var URLnova = Routing.generate('showSubsForm',  
    { 'subsFormat': format });
```

Aquest és un bundle senzill però imprescindible pel nostre projecte.

3.3.5 Canvi de velocitat dels fotogrames

Unes de les funcionalitats que volia destacar del convertidor és el canvi de velocitat dels fotogrames. Aquesta funcionalitat és comuna a tots els formats de subtítols i l'hem implementat dins de la classe pare Format.

Per canviar la velocitat dels fotogrames, primer, transformem el codi de temps (en format hh:mm:ss:ff) a nombre total de fotogrames tenint en compte la velocitat dels fotogrames(frame rate) d'entrada. Un cop obtinguts el nombre total de fotogrames, creem el nou codi de temps tenint en compte la velocitat dels fotogrames de sortida.

```
// Funció on enviem el codi de temps, el frame rate d'entrada  
// i el de sortida  
// Aquesta s'encarrega de cridar a les funcions que fan la conversió  
protected function changeFrameRate($tc, $originalFr, $targetFr) {  
    $frames = $this->tcToFrames($tc, $originalFr);  
    $newTc = $this->framesToTc($frames, $targetFr);
```

```
        return $newTc;
    }
    // Funció que transforma de fotogrames a codi de temps
    protected function framesToTc($tc, $frameRate) {

        $secondsTotal = floor($tc/$frameRate);
        $frames = $tc % $frameRate;

        $hours = floor($secondsTotal / 3600);
        $mins = floor($secondsTotal / 60 % 60);
        $secs = floor($secondsTotal % 60);

        $tcString = str_pad($hours, 2, "0",
            STR_PAD_LEFT).":".str_pad($mins, 2, "0",
            STR_PAD_LEFT).":".str_pad($secs, 2, "0",
            STR_PAD_LEFT).":".str_pad($frames, 2, "0", STR_PAD_LEFT);
        return $tcString;
    }

    // Funció que transforma de codi de temps a fotogrames
    protected function tcToFrames($tc, $frameRate) {

        $hores = substr($tc, 0, 2) * 3600 * $frameRate;
        $minuts = substr($tc, 3, 2) * 60 * $frameRate;
        $segons = substr($tc, 6, 2) * $frameRate;
        $frames = substr($tc, 9, 2);
        $framesTotals = $hores + $minuts + $segons + $frames;
        return $framesTotals;
    }
}
```

Les funcions de transformació les hem creat per separat perquè en algunes ocasions necessitem cridar-les individualment.

3.4 Seguretat

En l'apartat de seguretat s'han tingut en compte la seguretat de les dades, comunicacions i prevenció dels atacs més habituals.

Mesures sobre la seguretat de les dades:

- Validació de les dades que s'introdueixen al formulari, tant en el costat del client com en el servidor.

- Encriptació de la clau de l'usuari utilitzant la codificació bcrypt.
- L'administrador serà l'única persona que pugui accedir a la base de dades.

Mesures sobre les comunicacions:

- S'ha configurat el firewall de Symfony. Hi hem definit els diferents rols d'usuari i s'han limitat les rutes on aquests poden accedir.
- Pel que fa a les comunicacions en general, s'haurà de parlar amb l'empresa d'allotjament.

Mesura sobre la prevenció d'atacs:

Els dos tipus d'atacs més habituals que el nostre lloc web pot rebre són els d'injecció SQL i els de CSRF.

- Doctrine inclou un sistema de seguretat automàtic per evitar els atacs d'injecció SQL.
- Symfony inclou un sistema de seguretat per preveure els atacs CSRF, però en aquest cas l'hem de configurar. Per fer-ho, haurem d'accedir al fitxer de configuració de seguretat de Symfony. [app/config/security.yml](#). Un cop configurada només cal afegir el token en els formularis de registre i de recuperació.

Un bona pràctica de seguretat per evitar els atacs de robots és l'ús de captchas en els formularis de registre i de recuperació de clau. En aquesta versió no s'ha implementat, però es podria afegir en un futur.

3.5 Tests

Per verificar el bon funcionament de l'aplicació s'han creat els test següents:

Tests d'usuari

- Registrar un nou usuari, canviar la clau, tancar la sessió i tornar a accedir amb el nom d'usuari i la nova clau.
- Editar el nom i el correu en el formulari d'edició de perfil i intentar entrar a l'aplicació amb el nom i el correu antic.
- En els formularis de registre, edició de perfil, canvi de clau i recuperació de clau intentar introduir un nom d'usuari menor a 6 lletres i una clau menor de 8.
- En els formularis de registre i canvi de clau, on es demana repetir de la clau, intentar introduir dues claus diferents i comprovar si dona error.
- Entrar al formulari de recuperació de clau i enviar les dades. Quan es rebí el correu clicar a l'enllaç i restablir la clau. Comprovar que funciona. Com que encara no tenim l'aplicació al servidor, per poder veure que l'enviament del correu funciona, s'ha d'anar a l'arxiu [app/config/config_dev.yml](#) i variar el valor de [webprofiler:intercept_redirects](#) a [true](#). D'aquesta manera, podrem interceptar el correu que el sistema envia a l'usuari i accedir a l'enllaç que ens permet variar la clau.
- Accedir a l'aplicació amb un usuari registrat, crear nous paràmetres, editar-los, esborrar-los i comprovar que el llistat s'actualitza. Un cop fet, tancar la sessió i tornar a entrar per comprovar si el llistat de paràmetres és el mateix que abans de sortir.

Tests d'usabilitat

- Com que els usuaris que realitzin els nostres tests d'usabilitat han de tenir instal·lat el programari Final Cut Pro, hem demanat als stakeholders que provin la nostra aplicació. Se'ls demana que utilitzin el web i que sobretot pugin, exportin i provin tot tipus d'arxius de subtítol per posar a prova l'aplicació.

- Descarregar arxius .srt d'Internet de manera aleatòria i fer proves de lectura.
- Utilitzar arxius .srt, xml i .fcpxml que ja sabem que funcionin correctament i exportar-los a arxius de FCP i FCPX variant tots els paràmetres d'estil de manera evident per veure si funcionen correctament.
- Utilitzar arxius .srt amb pocs subtítols per comprovar que el canvi de velocitat dels fotogrames funciona correctament.
- Utilitzar arxius .srt amb pocs subtítols traduir-los a tots els idiomes utilitzant l'aplicació, i exportar-los als diferents formats.
- Comprovar que els arxius .csv siguin compatibles amb Excell i Open Office.

Tests de seguretat

- Introduir informació incorrecte en els formularis de la pàgina principal i comprovar que el sistema genera error.
- Fer proves simples de Sql injection. Un cop tinguem el servidor preparat, usarem un programari que verifiqui la fortalesa de la nostra aplicació davant d'atacs d'injecció.

3.6 Versions de l'aplicació

En aquest apartat, destaquem els continguts i les funcionalitats que s'han anat afegint dins de cada versió de l'aplicació.

Versió Alpha

- Esquelet de l'aplicació en Symfony

- Estructura de la vista(HTML, CSS i TWIG)
- Creació dels formularis d'exportació
- Lectura dels formats srt, xml i fcpxml
- Exportació a formats de subtítols srt i xml
- Exportació a formats de text .pdf i .csv
- Traducció dels subtítols

Versió Beta

- Millores en l'interfície de l'usuari (JavaScript i AJAX)
- Base de dades (Doctrine)
- Gestió dels usuaris
- Millores en els formularis
- Creació i edició dels paràmetres d'estil
- Incorporació de Bootstrap
- Creació de la pàgina about

Versió 1.0 *(preparada per ser publicada)*

- Exportació a format de subtítols fcpxml
- Exportació a format de text excel
- Millores en el rendiment de l'aplicació
- Millores en la visualització de la codificació de caràcters en l'exportació a formats xml i pdf.
- Millores en els estils
- Creació del logotip

- Creació de captches en els formularis de registre i de recuperació de clau.

3.7 Bugs

Després d'haver fet el test de proves només es detecten errors de certes codificacions de caràcters en l'exportació d'arxius pdf i xml.

- Problemes amb els caràcters asiàtics en l'exportació a pdf.
- Problemes amb la majoria de caràcters no occidentals en l'exportació a .XML. S'ha comprovat que el problema està en el convertidor i no en l'editor de vídeo.

Al marge d'aquests errors, es detecten problemes de rendiment quan es carrega un arxiu amb molts subtítols.

- S'alenteix el funcionament de la pàgina d'edició.
- Augmenta exponencialment el temps d'exportació d'un pdf.
- La traducció de subtítols triga en començar.

Tots aquests errors no són de fàcil solució i s'hauran de millorar en la versió.

3.8 Instruccions d'instal·lació

REQUISITS PHP

Abans d'instal·lar una aplicació Symfony haurem de comprovar si la nostra instal·lació de PHP compleix els requisits següents:

- Versió de PHP 5.3.9
- Extensió JSON habilitada
- Mòdul ctype habilitat
- Configuració del date-timezone en el php.ini

A causa de les característiques concretes de la nostra aplicació i amb la intenció d'evitar errors, es recomana canviar o controlar les directrius següents de l'arxiu php.ini.

- max_input_vars = 10000. Augmentar el nombre de variables que es poden enviar per POST.
- max_execution_time = 120 o més. Augmentar el temps d'execució atès que l'exportació d'algun pdf pot trigar.
- max_input_time = 120. Augmentar el temps de pujada d'arxius per si algun usuari disposa d'una connexió lenta.
- post_max_size = 64M o més. Augmentem la mida dels arxius del post perquè algun arxiu .xml pot pesar més del compte. Amb 64M ens assegurem que no hi haurà cap problema.
- File_uploads = On. Activem la pujada d'arxius.
- upload_max_size = 50M. Augmentem la mida dels arxius que pugem al servidor.

INSTAL·LANT L'APLICACIÓ

En primer lloc, hem de tenir Composer instal·lat de manera global al nostre sistema. Si utilitzem Linux o Mac OS X, haurem d'executar les comandes següents:

```
$ curl -Ss https://getcomposer.org/installer | php
```

```
$ sudo mv composer.phar /usr/local/bin/composer
```

Si utilitzem Windows, hem de descarregar l'instal·lador executable de Composer i seguir les passes indicades a l'enllaç següent: <https://getcomposer.org/doc/00-intro.md#installation-windows>

Un cop instal·lat Composer, hem d'obrir l'arxiu subs.zip i d'executar les ordres següents dins de la carpeta principal per a instal·lar les dependències del projecte (incloent-hi Symfony):

```
$ cd ./subs  
$ composer install
```

BASE DE DADES

Per configurar el nom, l'usuari i la clau base de dades haurem d'accedir a `app>config>parameters.yml`. Com que hem treballat amb Doctrine, també haurem de definir el sistema de gestió de base de dades (SGBD) que volem utilitzar. Per això, haurem d'anar a l'apartat `doctrine: dbal:` de l'arxiu `app>config>parameters.yml`.

En el nostre cas, com que hem utilitzat MySQL, hem definit el driver `pdo_mysql`. Però si vosaltres utilitzeu un altre SGBD, podeu consultar la nomenclatura dels diferents drivers a l'apartat [3.1.2 de la documentació de Doctrine](#).

Un cop definit el driver, haurem d'importar la base de dades (l'arxiu `subs.sql` que s'adjunta dins de la carpeta Codi) al nostre SGDB.

PROVANT L'APLICACIÓ

La forma més senzilla de provar l'aplicació és executar-la al lloc web intern de

PHP, que permet l'execució sense necessitat d'usar Apache o Nginx.

```
$ cd ./subs  
$ php app/console server: run
```

Si apareix aquest missatge: `Server running on http://localhost: 8000`, ja pots obrir el teu navegador i accedir a l'aplicació a l'adreça <http://localhost:8000/>.

USUARI

Per accedir a l'aplicació com a usuari, us podeu registrar o utilitzar un que hem creat de prova:

Nom de l'usuari: convidat.

Clau: Barceloneta.

FITXERS DE PROVA

Dins de la carpeta fitxers de prova, s'adjunten diferents arxius de prova.

- **film_ISO-8859-1.srt**: fitxer .srt codificat a ISO-8859-1 i a 23,98fps que conté 761 subtítols. Ideal per veure l'aplicació funcionant amb el tipus d'arxius que la majoria d'usuaris utilitzarà.
- **Uber_alles_windows1252.srt**: fitxer .srt codificat en Windows-1251 i a 24fps que conté 69 subtítols. Ideal per fer proves amb el traductor i d'exportació.
- **ukraines_windows1251.srt**: fitxer.srt codificat en Windows-1252 i a 23,98fps que conté quasi 122 subtítols. Ideal per comprovar el funcionament de la codificació de caràcters i fer una traducció a algun idioma que conegueu.
- **Uber_alles_REEL6_24fps.xml**: fitxer .xml de Final Cut Pro 7 que conté

quasi 91 subtítols. Ideal per comprovar la lectura dels arxius .xml.

- **arxiu_referencia.xml**: fitxer .xml sense cap subtítol que es pot utilitzar com arxiu de referència quan exportem .xml de Final Cut Pro 7.
- **prova.fcpxml**: fitxer de FCPX què conté 12 subtítols. Ideal per comprovar la lectura dels arxius .fcpxml.

4. Conclusions

Subtitle Converter for Video Editors és la primera aplicació web que realitzo després de cinc anys d'estudis a la UOC. Aquest treball m'ha servit, principalment, per adonar-me'n de la veritable dimensió que suposa la realització d'una aplicació web.

Quines lliçons hem après?

- A planificar un projecte multimèdia des de zero i poder veure el resultat d'aquesta planificació.
- A perdre la por a l'hora d'enfrontar-se a un nou projecte utilitzant una tecnologia desconeguda.
- A diferenciar realment quines són les tasques prioritàries i quines són les secundàries, sobretot en la fase de desenvolupament. S'estalvia moltíssim temps si les tasques d'implementació es fan en el moment que toca i no abans.
- A valorar la importància de cada etapa del procés. Millor trigar quatre dies en tancar una etapa que començar l'etapa següent abans d'hora. En realitat és un estalvi de temps encara que en el seu moment pensem que el calendari és el que mana.
- A entendre millor què és el disseny de l'arquitectura i adonar-me'n de la importància que pot arribar a tenir en l'èxit d'una aplicació.
- A diferenciar clarament les fases d'arquitectura i desenvolupament.
- A tenir en compte la documentació que s'ha creat durant l'etapa de disseny durant l'etapa d'implementació. En algun moment m'he bloquejat

per no saber fer alguna cosa i no recordava que ja ho havia solucionat en la fase de disseny.

- A entendre la importància de la especialització. En aquesta PAC he fet de dissenyador gràfic, he programat en 6 tecnologies diferents, he aplicat un entorn informàtic que no coneixia, he fet d'enginyer de programari, he creat la base de dades... Un cop acabat el treball em pregunto si és millor saber una mica de cada cosa o especialitzar-se en una. També, fer tantes tasques diferents m'ha ajudat a saber quines són les que més em motiven.
- A adonar-me de la quantitat de programari fet per tercers que existeix, i del suc que se'n pot treure si se'n fa un bon ús. Fent el treball he entès la importància real de la reutilització, terme que ha aparegut durant tot el Grau en Multimèdia.
- A saber quines són les virtuts de cada tecnologia que hem utilitzat i saber en quin moment del procés s'han d'implementar.
- A com documentar un projecte ajuda molt a millorar-lo.
- A conèixer el funcionament d'entorns com Symfony i de tecnologies com Twig i Bootstrap.
- A conèixer la tecnologia ORM, de la qual m'he enamorat.
- A ampliar coneixements en els llenguatges HTML, CSS, PHP, Javascript i JQuery.
- A entendre que una aplicació web és la suma de moltíssims elements i que com millor convergeixin entre ells, millor serà el resultat final.

Hem assolit els nostres objectius?

Com ja explico en la introducció d'aquest punt, si analitzo els objectius des d'un punt de vista personal, no puc estar més content de l'aprenentatge que ha suposat per mi la realització d'aquest treball.

En canvi, si ho faig des d'un punt de vista tècnic-professional, tot i que penso que s'han implementat totes les funcionalitats proposades, n'hi ha algunes de les quals no estic del tot satisfet de la manera que s'han resolt:

- La codificació de caràcters: volia aconseguir que l'aplicació llegís automàticament la codificació de caràcters de l'arxiu que pengés l'usuari i no ho he aconseguit. Finalment, serà l'usuari qui ho indiqui al sistema.
- Els estils i el disseny gràfic: no estic massa satisfet amb l'estètica de l'aplicació, sobre tot si ens fixem en els estils. Penso que l'ús que he fet de Bootstrap ha estat un error, ja que la nostra aplicació necessitava un disseny especial que li aportés caràcter propi sense allunyar-la de la seva finalitat. Bootstrap és ideal per entorn d'intranet perquè és fàcil d'implementar i visualment atractiu, però no per una aplicació que pretén mostrar-se al públic.

No crear un prototip HI-FI durant la fase de disseny va ser un error perquè segurament ens haguéssim adonat del problema. Cal crear un logotip per l'aplicació i alguns elements gràfics originals que ens permetin destacar-la.

- El dinamisme dels elements web: el rendiment de l'aplicació quan es treballa amb arxius de molts subtítols ha dificultat molt la inclusió d'animacions en els elements web. Posem un exemple, volíem que la columna de traducció de la pantalla d'edició s'obrís i és tanqués amb una animació quan l'usuari activés el traductor, però això no ha estat possible, perquè l'acció s'alentia molt quan el nombre de subtítols era superior a uns 700 aproximadament.

Anàlisi crítica del seguiment de la planificació i metodologia?

Un dels problemes principals que hem tingut durant el procés ha estat la planificació i, no tant el seguiment de la mateixa. És molt complicat planificar un projecte de manera adequada si no ho has fet abans.

En conseqüència, el nostre calendari partia d'una planificació incompleta, imprecisa i poc realista. Es va fer seguint les pautes que havíem après durant el Grau, però no vam aconseguir una planificació que s'ajustés a la realitat. Mancava definir alguns processos, els temps estaven desajustats i l'ordre d'algunes tasques era incorrecte.

Això va provocar que en comptes de seguir la planificació, anéssim fent en un ordre lògic fins a obtenir el resultat final. Evidentment, aquesta manera de funcionar ens ha provocat pèrdua de temps, però alhora, ens ha ajudat a entendre com fer una millor planificació en un futur.

En canvi, sobre la metodologia, penso que es va encertar en la manera de plantejar-la. Crec que seguir una metodologia en cascada és el que m'ha ajudat a acabar el projecte. Penso que aquesta metodologia és ideal per fer front a un primer projecte perquè t'ajuda a no perdre't.

Tot i això, em penedeixo de no haver començat a implementar l'aplicació en el mateix moment que vaig començar a analitzar l'entorn Symphony. Si ho hagués fet, els temps s'haguessin reduït bastant.

Un cop acabat el treball i després d'haver fet aquest primer projecte, penso que sóc molt més capaç de definir els diferents processos i tasques que formen part d'un projecte multimèdia, per tant, si hagués d'escollir una metodologia per un segon projecte triaria la iterativa.

Línies de treball futur a explorar o que han quedat pendents?

Personalment, penso que en aquests moments tenim una aplicació que implementa totes les seves funcionalitats, però que no acaba d'estar a punt per sortir al mercat i ser penjada en un servidor. Per tant, el nostre treball de futur, a banda de millorar algunes funcionalitats, s'ha de basar en fer de SCVE una aplicació adulta.

- Millorar els estils.
- Crear el logotip i elements gràfics per l'aplicació.
- Fer proves de rendiment al servidor, sobretot tenint en compte el funcionament de la pantalla d'edició quan es penja un arxiu amb moltes línies de subtítols i el temps d'exportació d'arxius pdf.
- Provar alternatives a la classe TCPDF per la generació d'arxius .pdf.
- Solucionar els problemes de codificació de caràcters en els arxius XML i PDF.
- Revisió i control dels errors de format dels arxius que puja l'usuari. També, s'ha de crear missatges d'error personalitzats per cada error.
- Intentar que el sistema sigui capaç de llegir per ell mateix la codificació de caràcters d'un arxiu.
- Revisar es arxius que puja l'usuari i implementar el control d'excepcions.
- Un cop obtinguts els resultats del rendiment, millorar el dinamisme dels elements gràfics de l'aplicació. Implementar finestres modals i loaders.
- Provar i valorar econòmicament la inclusió de l'Api de pagament de Google Translate.

- Crear les pàgines de contacte, suport i manuals.
- Introduir l'aplicació a les xarxes socials.
- Cercar un traductor professional que conegui el camp del vídeo perquè revisi i corregeixi els mots i els termes emprats a la nostra web.

5. Glossari

AJAX (Asynchronous Javascript And Xml). Tecnologia que permet realitzar peticions asincròniques al servidor. D'aquesta manera, permet modificar els continguts d'una pàgina web sense haver carregat-la.

ANNOTATION: és una forma d'afegir metadades al codi font d'una aplicació perquè després estiguin disponibles en temps d'execució.

API (Application Programming Interface): són un conjunt d'ordres, funcions i protocols informàtics que permeten als desenvolupadors crear programes específics per a certs sistemes operatius.

Bootstrap. És un entorn de treball basat en HTML i CSS, així com, extensions de JavaScript, que implementa plantilles que faciliten el disseny de llocs i d'aplicacions web.

Broadcast. És el conjunt de característiques que ha de complir un senyal per emetre programes de televisió o ràdio. Es coneix com a qualitat broadcast aquella que compleix els requisits d'emissió.

CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*): és un test de desafiament i resposta que es fa servir en computació per a determinar quan l'usuari és un humà o no.

CSRF(*Cross-Site Request Forgery*): és un mètode pel qual un usuari malintencionat intenta fer que els usuaris, sense saber-ho, enviïn dades que no volen enviar.

CSS (*Cascading Style Sheets*): Llenguatge de fulls d'estil que serveix per definir la forma dels elements que formen part d'un document HTML.

DCP (Digital Cinema Package). És el nom en que reben la col·lecció d'arxius digitals que s'usen per emmagatzemar i transmetre cinema digital, àudio, imatge i fluxos de dades.

FCPXML (Final Cut Pro eXtensible Markup Language). És un tipus d'arxiu XML creat especialment pel programari d'edició de vídeo Final Cut Pro.

Fotograma. És cadascuna de les imatges fixes que formen part d'una pel·lícula o imatge en moviment.

Framework (Entorn de treball). La seva traducció al català seria entorn o marc de treball i defineix la infraestructura de programari que, en la programació orientada a objectes, facilita la concepció de les aplicacions mitjançant la utilització de biblioteques de classes o generadors de programes.

HTML (Hyper Text Markup Language). Llenguatge de marcatge dissenyat per estructurar textos i que s'utilitza per la creació de documents en pàgines web. Està compost per una sèrie d'etiquetes que el navegador interpreta i dona forma a la pantalla.

HSB (Hue Saturation Brightness). Sistema de color que defineix el color segons la seva tonalitat(Hue), saturació(Saturation) i brillantor(Brightness).

Javascript. Llenguatge de programació interpretat que s'utilitza al costat del client i que serveix per millorar la interfície de l'usuari i el dinamisme en pàgines web.

Jquery. Entorn de treball basat en el llenguatge Javascript que facilita el seu ús.

MVC (Model Vista Controlador). És un patró de disseny del programari que s'utilitza per la implementació d'interfícies d'usuari.

MySQL. Sistema de gestió de bases de dades relacional usa el llenguatge SQL (Structured Query Language).

ORM (Object Relational Mapping) és una tècnica de programació per convertir dades de llenguatges de programació orientats a objectes en la seva representació en bases de dades relacionals.

PHP (Hipertext Preprocessor). Llenguatge de programació interpretat que s'utilitza per a generar pàgines web de forma dinàmica.

Plugin. Aplicació informàtica que afegeix una funcionalitat addicional o una nova característica a un programari ja existent.

Preset. Terme anglès que serveix per definir una cosa que ja està definida. En el món de l'audiovisual un preset són una sèrie d'ajustaments que ja venen preestablerts de sèrie.

RGB (Red Green Blue). Sistema de color que defineix un color segons la intensitat que té cada color primari dins del color representat: vermell(Red), verd(Green) i blau(Blue).

SYMFONY. Entorn de treball basat en el llenguatge PHP que facilita la creació de llocs web aplicant el patró de programari MVC.

TWIG. És un motor de plantilla pel llenguatge de programació PHP que facilita el seu ús i la seva nomenclatura.

SRT. Extensió que tenen els arxius de subtítols SubRip.

Timeline. És una manera de mostrar una sèrie d'objectes o esdeveniments en ordre cronològic. Aplicat a l'edició de vídeo, el timeline és l'espai on es col·loquen els arxius de vídeo que conformen el muntatge en ordre cronològic.

Wireframe. Esquema o prototip que ens mostra la disposició i la forma dels

elements que formaran part d'una pàgina web.

Editor de vídeo. És un programari que permet editar arxius de vídeo en un ordinador.

XML (eXtensible Markup Language). És un metallenguatge que s'utilitza per dissenyar llenguatges de marcat, i que permet definir etiquetes personalitzades i facilita l'organització de les dades.

6. Bibliografia

Llibres i apunts consultats:

- Eguiluz, Javier (2017). *Desarrollo Web Ágil con Symfony2*.
- D'Andrea, Edgar (2009). *PHP 6. Barcelona: Inforbooks*.
- Monjo Palau, Tona (2011). *Disseny d'interfícies multimèdia*. Barcelona: FUOC.
- Pradel Miquel, Jordi i Raya Martos, José A. (2011). *Anàlisi i disseny de patrons*. Barcelona: FUOC.
- Pradel Miquel i Jordi; Raya Martos, José A. (2011). *Enginyeria del programari*. Barcelona: FUOC.
- Berni Millet, Piero i Gil de la Iglesia, Dídac (2010). *Disseny de base de dades*. Barcelona: FUOC.
- Berni Millet, Piero i Gil de la Iglesia, Dídac (2011). *Desenvolupament web amb PHP i MySQL*. Barcelona: FUOC.
- Morville, Peter i Rosenfeld, Louis (2010). *Arquitectura de la informació per al Wide Web*. Barcelona: FUOC.
- Brínquez Jiménez, Jordi (2010). *Disseny i programació orientada a objectes*. Barcelona: FUOC.

Recursos en línia consultats:

- *Final Cut Pro X Format* [Manual en línia]. APPLE. [Data de consulta: juny del 2017]
<<https://developer.apple.com/library/content/documentation/FinalCutProX/Reference/FinalCutProXXMLFormat/Resources/Resources.html> - [//apple-ref/doc/uid/TP40011227-CH16-SW7](https://apple-ref/doc/uid/TP40011227-CH16-SW7)>
- *Final Cut Pro 7 XML Interchange Format* [Manual en línia]. APPLE. [Data de

consulta: juny del 2017]

<<https://developer.apple.com/library/content/documentation/FinalCutProX/Reference/FinalCutProXXMLFormat/Resources/Resources.html> - [//apple_ref/doc/uid/TP40011227-CH16-SW7](https://apple_ref/doc/uid/TP40011227-CH16-SW7)>

- *Subrip* [en línia]. Viquipèdia, l'Enciclopèdia Lliure. [Data de consulta: juny del 2016] <<https://es.wikipedia.org/wiki/SubRip>>
- *Symfony bundles* [Manual en línia]. SYMFONY. [Data de consulta: juny del 2017]<<https://symfony.com/doc/bundles/>>
- *Symfony reference* [Manual en línia]. SYMFONY. [Data de consulta: juny del 2017]<<https://symfony.com/doc/current/reference/index.html>>
- *Symfony components* [Manual en línia]. SYMFONY. [Data de consulta: juny del 2017]<<https://symfony.com/doc/current/components/index.html>>
- *Getting started with FOSUserBundle* [Manual en línia]. SYMFONY. [Data de consulta: juny del 2017] <<http://symfony.com/doc/current/bundles/FOSUserBundle/index.html>>
- Hernández Alejandro. *Introducción al framework Symfony* [Manual en línia]. SLIDESHARE. [Data de consulta: juny del 2017] <<https://www.slideshare.net/aleherse/introduccion-al-framework-symfony>>
- *Doctrine documentation* [Manual en línia]. DOCTRINE. [Data de consulta: juny del 2017]<<http://docs.doctrine-project.org/en/latest/>>
- *Symfony 2.3, el libro oficial* [en línia]. LIBROSWEB. [Data de consulta: juny del 2017] <http://librosweb.es/libro/symfony_2_3/>
- *Jquery.support* [Manual en línia]. JQUERY. [Data de consulta: juny del 2017]<<http://api.jquery.com/jquery.support/>>
- *Bootstrap getting started* [Manual en línia]. BOOSTRAP. [Data de consulta: juny del 2017]<<http://getbootstrap.com/getting-started/>>
- *Php manual* [Manual en línia]. PHP. [Data de consulta: juny del 2017]<<http://php.net/manual/es/index.php>>
- *Twig documentation* [Manual en línia]. TWIG. [Data de consulta: juny del 2017]<<https://twig.sensiolabs.org/doc/2.x/>>

- *Stack Overflow Contributions* [Manual en línia]. STACK OVERFLOW. [Data de consulta: juny del 2017]<<https://stackoverflow.com/>>
- *Usar la Api del traductor de Google* [en línia]. ADRIAN BARRIOS ANDRES. [Data de consulta: desembre del 2017]
<<https://statickidz.com/proyectos/traductor-google-api-php-gratis/>>
- *Bootstrap 3 Tutorial* [Manual en línia]. W3SCHOOLS. [Data de consulta: juny del 2017]<<https://www.w3schools.com/bootstrap/default.asp>>
- *HTML5 Tutorial* [Manual en línia]. W3SCHOOLS. [Data de consulta: juny del 2017]<<https://www.w3schools.com/html/default.asp>>

7. Annexos

Annex 1 - Final Cut Pro Interchange Format (XML)

▪ CARACTERÍSTQUES DEL FORMAT

- **Extensió:** arxius XML amb extensió .xml.
- **Nombre de versions:** 5 versions que estan relacionats amb les diferents versions del programari.
 - FCP4 suporta XML versió 1
 - FCP5 suporta XML versions 1 i 2
 - FCP5.1 suporta XML versions 1, 2 i 3
 - FCP6 suporta XML versions 1, 2, 3 i 4
 - FCP7 suporta XML versions 1, 2, 3, 4 i 5.

Les versions noves del programari són compatibles amb els arxius .xml antics i l'etiquetatge dels clips de text és igual en totes les versions. Per tant, implementaren les funcionalitats següents:

- L'aplicació podrà llegir les 5 versions de l'arxiu .xml.
- Si l'usuari penja un arxiu .xml de referència, l'aplicació exportarà l'xml en la mateixa versió que l'hagi penjat l'usuari.
- Si l'usuari vol crear un .xml nou des de l'aplicació, aquest serà la versió 1.

Per aquesta raó, en l'apartat següent analitzem l'estructura de la versió 1 de l'xml.

- **Editors de text:** Els subtítols o els títols en Final Cut Pro 4/5/6 i 7 es creen, principalment, amb dos editors de textos, Text i Outline Text, que venen de sèrie amb el programari. Es possible que s'utilitzin d'altres

però, en el cas de la lectura admetrem només aquests dos i, en el cas de l'escriptura, utilitzarem "Outline Text", atès que ens permet definir un major nombre de paràmetres.

▪ FINAL CUT PRO INTERCHANGE FORMAT versió 1.0

Estructura general del fitxer

Com es pot comprovar en l'exemple que hi ha més endavant, l'arrel de l'arxiu està formada per dues etiquetes: <xmemeI>, que identifica quina és la versió del XML, i, <sequence>, on es guarda tota la informació sobre la seqüència del projecte.

En els primers fills de l'etiqueta <sequence> es guarden els paràmetres generals de la seqüència: el nom a <name>, la duració a <duration>, la informació sobre els fotogrames a <rate>, la informació sobre el codi de temps <timecode> i, finalment, tota la informació respecte el vídeo i l'àudio a <media>.

Dins de <media>, es troba l'etiqueta <vídeo> que conté <format>, on hi troben les característiques del format de vídeo, i les etiquetes <track> que guarden la informació que hi ha editada a cada pista de vídeo. En un .xml hi ha tantes etiquetes <track> com pistes de vídeo tingui la seqüència. L'ordre de les etiquetes es correspon a el de les pistes de vídeo que hi ha al timeline del projecte.

En groc es marquen aquelles etiquetes que contenen informació que haurem de tenir en compte sobre la seqüència de vídeo d'un arxiu XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE xmemeI>
<xmemeI version="1"> ----> versió del XML
<sequence id="New sequence"> ----> ID = nom de la seqüència
  <name>New sequence</name> ----> nom de la seqüència
  <duration>116524</duration> ----> duració total de la seqüència en FOTOGRAFES
  <rate>
    <ntsc>FALSE</ntsc> ----> Sistema de vídeo
```

```

<timebase>24</timebase> ----> Framerate
</rate>
<timecode>
  <rate>
    <ntsc>FALSE</ntsc> ----> Sistema de vídeo
    <timebase>24</timebase> ----> Framerate
  </rate>
  <string>01:00:00:00</string> ----> Inici del timecode de la seqüència
  hores:minuts:segons:fotogrames
  <frame>86400</frame> ----> duració total de la seqüència en FOTOGRAMES
  <source>source</source>
  <displayformat>NDF</displayformat>
</timecode>
<in>-1</in>
<out>-1</out>
<media>
  <video>
    <format>
      <samplecharacteristics>
        <width>1920</width> ----> Resolució en píxels de l'ample
        <height>1080</height> ----> Resolució en píxels de l'alçada
        <anamorphic>FALSE</anamorphic> ----> Aspecte anamòrfic
        <pixelaspectratio>Square</pixelaspectratio> ----> Relació d'aspecte píxel
        <fielddominance>none</fielddominance> ----> Camp dominant
      <rate>
        <ntsc>FALSE</ntsc> ----> Sistema de vídeo
        <timebase>24</timebase> ----> Framerate
      </rate>
      <colordepth>24</colordepth>
    </samplecharacteristics>
  </format>
</video>
</media>

```

Estructura dels subtítols – Final Cut Interchange Pro Format v 1.0

L'etiqueta <generatoritem> és l'encarregada de definir un clip que ha estat creat a partir d'un efecte o preset de FCP7. El nom del preset usat apareix com atribut "id" dins de l'etiqueta.

En el nostre cas, l'etiqueta <generatoritem> és l'encarregada de definir un clip de text nou (en aquest cas, usant el preset "Outline Text") que contindrà els nostres subtítols. Aquesta etiqueta, <generatoritem>, es situa dins de <track>, que és la pista de vídeo on es troba el subtítol.

Dins de cada <generatoritem>, s'especifica el contingut, la forma i l'estil de cada subtítol. Com podem veure en el exemple següent, es poden definir una quantitat important de paràmetres que afecten a l'estil del text.

En groc es marquen aquelles etiquetes que defineixen els diferents paràmetres. Per veure el codi sencer anar a l'ANNEX1.

```
<generatoritem id="Outline Text">
  <name>Outline Text</name>
  <duration>3000</duration>
  <rate>
    <ntsc>FALSE</ntsc>
    <timebase>24</timebase>
  </rate>
  <in>1375</in>
  <out>1435</out>
  <start>1400</start> ----> Codi de temps d'entrada en FOTOGAMES
  <end>1460</end> ----> Codi de temps sortida en FOTOGAMES
  <enabled>TRUE</enabled>
  <anamorphic>FALSE</anamorphic>
  <alphatype>black</alphatype>
  <effect>
    <name>Outline Text</name>
    <effectid>Outline Text</effectid>
    <effectcategory>Text</effectcategory>
    <effecttype>generator</effecttype>
    <mediatype>video</mediatype>
    <parameter>
      <parameterid>part1</parameterid>
      <name>Text Settings</name>
      <value/>
    </parameter>
    <parameter>
      <parameterid>str</parameterid>
      <name>Text</name>
      <value>Estàs cansat? &#13;Estàs saturat? </value> ----> Contingut
del subtítol (&#13; defineix el salt de línia)
    </parameter>
    <parameter>
      <parameterid>font</parameterid>
      <name>Font</name>
      <value>Verdana</value> ----> Nom de la tipografia
    </parameter>
    <parameter>
      <parameterid>style</parameterid>
      <name>Style</name>
      <valuemin>1</valuemin>
      <valuemax>4</valuemax>
      <valuelist>
        <valueentry>
          <name>Plain</name>
```

```

        <value>1</value>
    </valueentry>
    <valueentry>
        <name>Bold</name>
        <value>2</value>
    </valueentry>
    <valueentry>
        <name>Italic</name>
        <value>3</value>
    </valueentry>
    <valueentry>
        <name>Bold/Italic</name>
        <value>4</value>
    </valueentry>
</valuelist>
<value>1</value> ----> Estil del text(1,2,3,4)
</parameter>
<parameter>
    <parameterid>align</parameterid>
    <name>Alignment</name>
    <valuemin>1</valuemin>
    <valuemax>3</valuemax>
    <valuelist>
        <valueentry>
            <name>Left</name>
            <value>1</value>
        </valueentry>
        <valueentry>
            <name>Center</name>
            <value>2</value>
        </valueentry>
        <valueentry>
            <name>Right</name>
            <value>3</value>
        </valueentry>
    </valuelist>
    <value>2</value> ----> Alineació
</parameter>
<parameter>
    <parameterid>size</parameterid>
    <name>Size</name>
    <valuemin>0</valuemin>
    <valuemax>200</valuemax>
    <value>19</value>----> Mida de la tipografia
</parameter>
<parameter>
    <parameterid>track</parameterid>
    <name>Tracking</name>
    <valuemin>0</valuemin>
    <valuemax>100</valuemax>
    <value>2</value> ----> Tracking
</parameter>
<parameter>
    <parameterid>lead</parameterid>
    <name>Leading</name>
    <valuemin>-100</valuemin>
    <valuemax>100</valuemax>

```

```

    <value>0</value> ----> Leading
</parameter>
<parameter>
  <parameterid>aspect</parameterid>
  <name>Aspect</name>
  <valuemin>0</valuemin>
  <valuemax>4</valuemax>
  <value>1</value> ----> Aspect
</parameter>
<parameter>
  <parameterid>linewidth</parameterid>
  <name>Line Width</name>
  <valuemin>0</valuemin>
  <valuemax>200</valuemax>
  <value>12</value> ----> Line Width
</parameter>
<parameter>
  <parameterid>linesoft</parameterid>
  <name>Line Softness</name>
  <valuemin>0</valuemin>
  <valuemax>100</valuemax>
  <value>18</value> ---->Line Softness
</parameter>
<parameter>
  <parameterid>textopacity</parameterid>
  <name>Text Opacity</name>
  <valuemin>0</valuemin>
  <valuemax>100</valuemax>
  <value>100</value> ---->Text Opacity
</parameter>
<parameter>
  <parameterid>center</parameterid>
  <name>Center</name>
  <value>
    <horiz>0</horiz> ---->Posició X
    <vert>0.333333</vert> ---->Posició Y
  </value>
</parameter>
<parameter>
  <parameterid>textcolor</parameterid>
  <name>Text Color</name>
  <value>
    <alpha>255</alpha> ---->Text Color Alpha
    <red>230</red> ----> Text Color Vermell
    <green>230</green>---->Text Color Verd
    <blue>230</blue> ----> Text Color Blau
  </value>
</parameter>
<parameter>
  <parameterid>supertext</parameterid>
  <name>Text Graphic</name>
</parameter>
<parameter>
  <parameterid>linecolor</parameterid>
  <name>Line Color</name>
  <value>
    <alpha>255</alpha> --> Line Color Alpha

```

```

        <red>0</red> --> Line Color Vermell
        <green>0</green> --> Line Color Verd
        <blue>0</blue> --> Line Color Blau
    </value>
</parameter>
<parameter>
    <parameterid>superline</parameterid>
    <name>Line Graphic</name>
</parameter>
<parameter>
    <parameterid>part2</parameterid>
    <name>Background Settings</name>
    <value/>
</parameter>
<parameter>
    <parameterid>xscale</parameterid>
    <name>Horizontal Size</name>
    <valuemin>0</valuemin>
    <valuemax>200</valuemax>
    <value>0</value> ----> Background Size x
</parameter>
<parameter>
    <parameterid>yscale</parameterid>
    <name>Vertical Size</name>
    <valuemin>0</valuemin>
    <valuemax>200</valuemax>
    <value>0</value> ----> Background Size y
</parameter>
<parameter>
    <parameterid>xoffset</parameterid>
    <name>Horizontal Offset</name>
    <valuemin>-100</valuemin>
    <valuemax>100</valuemax>
    <value>0</value> ----> Background offset x
</parameter>
<parameter>
    <parameterid>yoffset</parameterid>
    <name>Vertical Offset</name>
    <valuemin>-100</valuemin>
    <valuemax>100</valuemax>
    <value>0</value> ----> Background offset y
</parameter>
<parameter>
    <parameterid>backsoft</parameterid>
    <name>Back Soft</name>
    <valuemin>0</valuemin>
    <valuemax>100</valuemax>
    <value>0</value> ----> Back Softness
</parameter>
<parameter>
    <parameterid>backopacity</parameterid>
    <name>Back Opacity</name>
    <valuemin>0</valuemin>
    <valuemax>100</valuemax>
    <value>50</value> ----> Back Opacity
</parameter>
<parameter>

```

```

        <parameterid>backcolor</parameterid>
        <name>Back Color</name>
        <value>
            <alpha>255</alpha> ---> Back Color Alpha
            <red>255</red> ----> Back Color Blau
            <green>255</green> --> Back Color Verd
            <blue>255</blue> ----> Back Color Blau
        </value>
    </parameter>
    <parameter>
        <parameterid>superback</parameterid>
        <name>Back Graphic</name>
    </parameter>
    <parameter>
        <parameterid>crop</parameterid>
        <name>Crop</name>
        <value>FALSE</value> ----> Crop
    </parameter>
    <parameter>
        <parameterid>autokern</parameterid>
        <name>Auto Kerning</name>
        <value>TRUE</value> ----> Autokerning
    </parameter>
</effect>
<sourcetrack>
    <mediatype>video</mediatype>
</sourcetrack>
</generatoritem>

```

Lectura dels subtítols – Final Cut Pro Interchange Format v 1.0

En aquest apartat, analitzarem quines són els paràmetres que s'han de tenir en compte per realitzar la lectura del subtítols quan un usuari penja com a arxiu principal un arxiu .XML de Final Cut 7 en qualsevol de les versions.

Per realitzar la lectura, si agafem com a referència l'exemple anterior, haurem de buscar tots els <generatoritem> que es troben dins d'un <track> i haurem de llegir les etiquetes següents:

- <timebase> de la <sequence>: quin és la velocitat de fotogrames de la seqüència
- <start> de cada subtítol: codi de temps d'entrada
- <end> de cada subtítol: codi de temps de sortida

- `<text><value>` de cada subtítol: text de cada subtítol. S'haurà de comprovar el nombre de línies.
- `<style><value>` de cada subtítol: saber quin estil té el subtítol.

Ara bé, a l'hora de fer la lectura dels subtítols en l'arxiu .XML poden aparèixer alguns dels problemes que hem de tenir en compte.

- Que els subtítols estiguin editats en pistes de vídeo diferents
- Que s'hagin utilitzat presets diferents per editar els textos
- Que hi hagi títols que no són subtítols, poden ser títols de crèdit o altres.

Com que els subtítols en realitat són clips de vídeo generats per un editor de textos, no podem saber si les etiquetes `<generatoritem>` amb id "Outline Text" o "Text" són subtítols o altres títols. Per tant, en principi l'aplicació llegirà tots els `<generatoritem>` amb id "Outline Text" o "Text" que hi hagi dins de l'XML.

En el cas que es trobin `<generatoritem>` en pistes diferents o `<generatoritem>` amb presets diferents, s'avisarà l'usuari i se li suggerirà que exporti l'.xml amb tots els subtítols en una sola pista de vídeo, que en aquella pista només hi hagin clip de text que facin referència als subtítols i que usi els presets "Outline Text" o "Text".

En verd es marquen les etiquetes que ens ajudaran a implementar el mecanisme de comprovació.

```
<track>  
  <generatoritem id="Outline Text">  
    <name>Outline Text</name>  
    <duration>3000</duration>  
    <rate>  
      <ntsc>FALSE</ntsc>  
      <timebase>24</timebase>  
    </rate>
```

```
<in>1375</in>
<out>1435</out>
<start>1400</start>
<end>1460</end>
<enabled>TRUE</enabled>
<anamorphic>FALSE</anamorphic>
<alphatype>black</alphatype>
<effect>
  <name>Outline Text</name>
  <effectid>Outline Text</effectid>
  <effectcategory>Text</effectcategory>
  <effecttype>generator</effecttype>
  <mediatype>video</mediatype>
</effect>
```

Esriptura dels subtítols – Final Cut Pro Interchange Format v 1.0

Per escriure els subtítols, dins de l'etiqueta <media> crearem una pista de vídeo nova amb l'etiqueta <track> i la situarem posarem com a últim fill. –d'aquesta manera es situarà a dalt de tot del timeline del programa d'edició.–

A dins dels <tracks> crearem per ordre tants <generatoritem> com subtítols haguem d'escriure. Dins de cada <generatoritem> escriurem tota la informació de cada subtítol i tots els paràmetres que l'usuari hagi definit.

Paràmetres d'estil – Final Cut Pro Interchange Format v 1.0

En el cas de l'xml de FCP7, la definició de paràmetres és essencial, atès que el programari no permet editar els paràmetres de més d'un clip de text alhora. Per exemple, si volem variar el color de 600 subtítols, ho haurem de fer un a un. Per aquesta raó, l'aplicació ha de permetre a l'usuari definir tots aquells paràmetres que afecten a l'estil dels subtítols.

També, ha de permetre definir els paràmetres de vídeo com la resolució o la velocitat de fotogrames que defineixen la seqüència, però només en el cas que l'usuari no pengi un arxiu .xml de referència.

En el document ***Memoria/Fitxer adjunts/FCP7_xml_parametres.xls*** es troben els paràmetres que l'usuari pot definir i els valors que poden tenir.

Annex 2 - Final Cut Pro X Format (FCPXML)

▪ CARACTERÍSTQUES DEL FORMAT

- **Arxius i extensió:** arxius XML amb extensió .fcpxml.
- **Nombre de versions:** d'aquest format existeixen 6 versions que s'implementen en les diferents versions del programari. La versió actual del programari és la 10.3.2 i la versió de l'XML és la 1.6.
 - A partir de la 10.0.3 s'implementa XML 1.1
 - A partir de la 10.0.6 s'implementa XML 1.2
 - A partir de la 10.1.0 s'implementa XML 1.3
 - A partir de la 10.1.2 s'implementa XML 1.4
 - A partir de la 10.2.0 s'implementa XML 1.5
 - A partir de la 10.3.0 s'implementa XML 1.6

Per una banda, ens trobem que les versions noves del programari són compatibles amb els arxius .fcpxml antics. Per l'altra, que l'etiquetatge és diferent segons la versió de l'XML i, això, afecta als subtítols, atès que es varia lleugerament l'estructura de l'arxiu, la definició dels clips de text i, sobretot, el nombre de paràmetres que es poden emmagatzemar.

Fins la versió 1.2 s'etiqueta d'una manera i a partir de la 1.3 d'una altra. Per tant, haurem de crear dues possibilitats diferents d'exportació, la 1.1 i la 1.3. Implementarem les següents funcionalitats:

- L'aplicació podrà llegir les 6 versions de l'arxiu .xml.
 - Si l'usuari penja un .fcpxml de referència, l'aplicació exportarà el .fcpxml en la mateixa versió que hagi penjat l'usuari.
 - Si l'usuari vol crear un .fcpxml nou des de l'aplicació, aquest podrà escollir entre la versió 1.1 i la 1.3.
 - Els subtítols o els títols en Final Cut Pro X es creen, principalment, amb els presets següents de l'editor de textos: Custom i Basic Text en la versió 1.1, i, Basic Text, Textup Regular i Textup Box a partir de la 1.3. Es possible que s'utilitzin d'altres però, en el cas de la lectura admetrem només aquests quatre i, en el cas de l'escriptura, utilitzarem "Basic Text", atès que ens permet definir els paràmetres adequats per generar subtítols.
- **FINAL CUT PRO X FORMAT versió 1.1**

Estructura general del fitxer

Com es pot comprovar en l'exemple que hi ha més endavant, l'arrel de l'arxiu està formada per una sola etiqueta: <fcpxml>, que identifica la versió de l'arxiu. Dins hi trobem <project>, on es guarda tota la informació del projecte.

L'etiqueta <project> té dos fills principals: <resources>, on es defineixen els recursos que hi ha al projecte, com el format de la seqüència de vídeo, els clips de vídeo i els efectes que hi participen, i <sequence>, on es defineixen les característiques dels elements que la formen, com els clips de vídeo, àudio o els clips text. Aquestes dues són imprescindibles alhora d'etiquetar qualsevol element present al timeline.

A diferència dels .xml de FC7, en els fcpxml s'usen atributs per definir els paràmetres i no etiquetes. *En groc es marquen aquelles etiquetes i atributs que contenen informació que haurem de tenir en compte sobre el projecte.*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fcpxml>
<fcpxml version="1.1"> ---> Versió del .fcpxml
  <project name="New Project 3" (Nom del projecte) uid="5AF70453-360F-4024-92FE-
E59ECE99BD1C" eventID="F6A3231B-681C-4E9A-93FE-67647DBF605F" location="">
    <resources>
      <format id="r1" (Nom del projecte) name="FFVideoFormat1080p24" (nom del còdec de
vídeo) frameDuration="100/2400s" (frame rate) width="1920" (Resolució vídeo ample)
height="1080" (Resolució vídeo alçada) />
      <asset id="r2" name="CALLBACK MASTER CLEAN PROXY"
uid="1BF75474A8765DB639B2410B3FF1E74D" projectRef="r2" src=" " start="0s"
duration="116524/24s" hasVideo="1" hasAudio="1" audioSources="1" audioChannels="2"
audioRate="48000"> ---> Defineix un clip de vídeo
        <effect id="r3" name="Basic Title" uid=".../Titles.localized/Bumper:Opener.localized/Basic
Title.localized/Basic Title.moti" (identificador de l'editor de text usat)/>
        <effect id="r4" name="Custom" uid=".../Titles.localized/Build
In:Out.localized/Custom.localized/Custom.moti" (identificador de l'editor de text usat)/>
      </resources>
      <sequence duration="31300/2400s" (Duració seqüència) format="r1" (recurs usat)
tcStart="0s" (Timecode d'inici) tcFormat="NDF" audioLayout="stereo" audioRate="48k">
    </sequence>
  </project>
</fcpxml>
```

Estructura dels subtítols – Final Cut Pro Format v 1.1

Dins de l'etiqueta <sequence> existeix l'etiqueta <spine> que és l'encarregada de vertebrar els continguts de la seqüència. Dins d'aquesta hi trobarem les pistes de vídeo <clip>. i els clips de text <text>.

Cada subtítol anirà dins d'una etiqueta <title>. Aquesta tindrà com a atributs el nom, la duració, codi de temps d'entrada i el recurs(<resource>) que usa. Dins de <title> hi trobem dues etiquetes: <text>, on es guarda el contingut del text i <adjust-transform>, on es guarden els paràmetres de posició del

clip. En les versions 1.1 i 1.2 del fcpxml només podrem emmagatzemar aquest paràmetre. Per tant, no podrem definir l'estil del text.

En groc es marquen aquelles etiquetes que defineixen els diferents paràmetres.

```
<sequence duration="31300/2400s" format="r1" tcStart="0s" tcFormat="NDF"
audioLayout="stereo" audioRate="48k">
  <spine>
    <title offset="0s" (Codi de temps d'inici) ref="r2" (Recurs usat) name="Title1" (Nom del
clip) duration="11900/2400s" (Duració del clip) start="3600s">
      <adjust-transform position="0 -34.6296"/> ----> Posició
      <text>- Hola, com estas?
- Bé, tot bé.</text> ----> Contingut textutal del clip
    </title>
    <gap offset="11900/2400s" name="Gap" duration="7300/2400s" start="3600s"/>
    <title offset="8s" ref="r3" name="Title2" duration="12100/2400s" start="3600s">
      <adjust-transform position="0 -34.6296"/>
      <text>- Hola, com estas?
- Bé, tot bé.</text>
    </title>
  </spine>
</sequence>
```

En l'exemple veiem l'etiqueta <gap> entre els dos títols. Aquesta etiqueta és un concepte nou a FCPX i serveix per definir si hi ha un espai buit a la seqüència. A la pràctica el que fa és generar un clip buit entre dos clips. Aquesta etiqueta no és imprescindible, però ens podrà ser útil utilitzar-la en el moment de l'escriptura.

Un canvi important respecte a FCP7 és que el concepte de tracks(pistes de vídeo) funciona diferent. En comptes de haver-hi pistes de vídeo separades, hi ha etiquetes embrancades que defineixen la col·locació dels clips al timeline. Sempre que hi ha un recurs associat a un altre, el recurs secundari és fill del principal.

Per exemple, si els títols(<title>) van associats a un clip de vídeo, aquests aniran dins de l'etiqueta del clip de vídeo(<clip>). En aquesta situació, apareix un nou element: l'atribut "lane", que ens diu a quina alçada es troben els elements respecte el seu element contenidor.

Veiem l'exemple:

```
<sequence duration="11652400/2400s" format="r1" tcStart="0s" tcFormat="NDF"
audioLayout="stereo" audioRate="48k">
  <spine>
    <clip offset="0s" name="CALLBACK MASTER CLEAN PROXY" duration="11652400/2400s"
tcFormat="NDF"> ----> Etiqueta del clip de vídeo principal
      <video offset="0s" ref="r3" duration="11652400/2400s">
        <audio lane="-1" (Alçada de l'etiqueta d'àudio, en aquest cas -1 perquè està per sota
el vídeo) offset="0s" ref="r3" duration="233048000/48000s" role="dialogue"/>
      </video>
      <title lane="1" (Alçada de l'etiqueta d'àudio, en aquest cas 1 perquè està per sobre el
vídeo) offset="94400/2400s" ref="r4" name="Title1" duration="24100/2400s" start="3600s">
        <text>- Hola, com estàs?
- Bé, molt bé</text>
      </title>
      <title lane="1" offset="141900/2400s" ref="r4" name="Title2" duration="24100/2400s"
start="3600s">
        <adjust-transform position="0 -30.0926"/>
        <text>- Hola, com estàs?
- Bé, molt bé</text>
      </title>
    </clip>
  </spine>
</sequence>
```

L'absència de pistes de vídeos i l'estructura de l'etiquetatge dels subtítols dificulta moltíssim l'escriptura de clips de text en un .fcxmpl de referència. Per exemple, si un usuari penja un .fcpxml amb moltíssims clips editats i espais buits serà molt complicat saber dins de quins clips s'ha de col·locar cada clip de text i a quina alçada s'ha de situar. Per tant, de moment, no permetrem que l'usuari pugui penjar un .fcpxml de referència. Només, podrà exportar-ne un creat des de zero. Si ho comparem amb la versió de FCP7, en aquesta funcionalitat dona molt la feina per el resultat que genera.

Lectura dels subtítols – Final Cut Pro Format v 1.1

Per llegir els subtítols haurem de buscar totes aquelles etiquetes <title> que hi hagi dins del <spine> i llegir els atributs i etiquetes següents:

- offset a <title> : Codi de temps d'entrada
- duration a <title>: Duració del subtítol.
- <text>: Text de cada subtítol. S'haurà de comprovar el nombre de línies.

En les versions 1.1 i 1.2 no es defineix l'estil del text, per tant, aquest sempre serà pla.

No obstant, haurem de tenir en compte possibles problemes:

- Que hi hagi títols que no són subtítols, poden ser títols de crèdit o altres.
- Que els subtítols s'hagin generat amb un preset diferent als que contemplem

Com que els subtítols en realitat són clips de vídeo generats per un editor de textos, no podem saber si les etiquetes <title> amb id "Basic Text" o "Custom" són subtítols o altres títols. Per tant, en principi l'aplicació llegirà tots els <title> amb id "Basic Text" o "Custom" que hi hagi dins de l'XML.

En el cas que es trobin <title> a alçades diferents o <title> amb presets diferents, s'avisarà l'usuari i se li suggerirà que exporti l'.xml amb tots els subtítols a la mateixa alçada, i que usi els presets "Basic Text" o "Custom".

En verd es marquen les etiquetes i atributs que ens ajudaran a implementar el mecanisme de comprovació.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fcpxml>
<fcpxml version="1.2">
  <project name="New Project 3" uid="5AF70453-360F-4024-92FE-E59ECE99BD1C"
eventID="F6A3231B-681C-4E9A-93FE-67647DBF605F" location="">
    <resources>
      <format id="r1" name="FFVideoFormat1080p24" frameDuration="100/2400s" width="1920"
height="1080"/>
      <effect id="r2" name="Basic Title" uid=".../Titles.localized/Bumper:Opener.localized/Basic
Title.localized/Basic Title.moti"/>
      <effect id="r3" name="Custom" uid=".../Titles.localized/Build
In:Out.localized/Custom.localized/Custom.moti"/>
    </resources>
    <sequence duration="31300/2400s" format="r1" tcStart="0s" tcFormat="NDF"
audioLayout="stereo" audioRate="48k">
      <spine>
        <title offset="0s" ref="r2" name="Title1" duration="11900/2400s" start="3600s">
          <adjust-transform position="0 -34.6296"/>
          <text>- Hola, com estas?
- Bé, tot bé.</text>
        </title>
        <gap offset="11900/2400s" name="Gap" duration="7300/2400s" start="3600s"/>
        <title offset="8s" ref="r3" name="Title2" duration="12100/2400s" start="3600s">
          <adjust-transform position="0 -34.6296"/>
          <text>- Hola, com estas?
- Bé, tot bé.</text>
        </title>
      </spine>
    </sequence>
  </project>
</fcpxml>

```

Esriptura dels subtítols – Final Cut Pro Format v 1.1

L'absència de pistes de vídeos i l'estructura de l'etiquetatge dels subtítols dificulta l'escriptura de clips de text en un .fcpxml de referència. Per exemple, en el cas que l'usuari pengi un arxiu amb múltiples clips editats a diferents alçades.

Per tant, primer mirarem si l'<spine> està buit o ple. En el cas que no estigui buit, haurem de comprovar si el codi de temps d'entrada del subtítol coincideix amb algun clip de vídeo o recurs situat dins de l'<spine>. Si és el cas, crearem un element <title> dins d'aquest recurs i el situarem en l'espai que li correspon. Sinó coincideix amb cap recurs, crearem un <gap> i el situarem el <title> a dins.

En el cas que l'<spine> estigui buit, els <title> és situaran per ordre un darrera l'altre dins de l'<spine>.

Paràmetres d'estil – Final Cut Pro Format v 1.0

En aquest cas, a banda dels paràmetres de vídeo que defineixen la seqüència, l'usuari només podrà definir la posició del subtítols.

En el document *Memoria/Fitxers adjunts/FCPX_fcpxml1_1_parametres.xls* es troben els paràmetres que l'usuari pot definir i els valors que poden tenir.

▪ FINAL CUT PRO X FORMAT versió 1.3

Estructura general del fitxer

L'estructura general de la versió 1.3 és igual que la de la 1.2. Ara bé, hem de tenir en compte que a partir de la versió 1.4 apareixen dues noves etiquetes dins de l'arrel: <library> i <event> que són pare i avi de <project> i avi i besavi de <sequence>. En principi, cap de les dues ens farà cap servei ni ens hauria de complicar la lectura o l'escriptura.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fcpxml>
```

```
<fcpxml version="1.5">
  <resources>
    <format id="r1" name="FFVideoFormat1080p2398" frameDuration="1001/24000s"
width="1920" height="1080"/>
```

```

    <effect id="r2" name="Título básico" uid=".../Titles.localized/Bumper:Opener.localized/Basic
Title.localized/Basic Title.moti"/>
  </resources>
  <library location="file:///Users/vhtorner/Movies/Sin%20ti%CC%81tulo.fcpbundle/">
    <event name="Proyecto sin título" uid="3239EE17-FBFD-4AD1-9CE8-70F6A56B82B8">
      <project name="Proyecto sin título" uid="512FB56D-EF52-4240-9B9C-27F974741528">
        <sequence duration="4694690/120000s" format="r1" tcStart="0s" tcFormat="NDF"
audioLayout="stereo" audioRate="48k">
          </sequence>
        </project>
      </event>
    </library>
  </fcpxml>

```

Estructura dels subtítols – Final Cut Pro Format v 1.3

A partir de la versió 1.3, l'estructura varia bàsicament en la definició dels clips de text, atès que aquesta versió permet emmagatzemar més informació sobre els paràmetres que afecten al text.

Com en la versió 1.1, cada clip de text es genera amb una etiqueta <title> amb els atributs nom, duració, codi de temps d'entrada i recurs. Dins de l'etiqueta <title> la cosa comença a canviar. Les <title> tenen associades les etiquetes <text> i <text-style-def> on es defineixen tots els atributs de l'estil del text. Dins de <text>, hi ha <text-style> on es guarda el contingut i l'identificador de l'estil.

En groc es marquen aquelles etiquetes que defineixen els diferents paràmetres.

```

<sequence duration="4694690/120000s" format="r1" tcStart="0s" tcFormat="NDF"
audioLayout="stereo" audioRate="48k">
  <spine>
    <title offset="0s"(Codi de temps d'inici) ref="r2"(Recurs usat) name="Title1"(Nom del
clip) duration="11900/2400s"(Duració del clip) start="3600s">
      <text>
        <text-style ref="ts1" (referència a l'estil del text) >- Hola
- Adiós</text-style> ---> Contingut textutal del clip
      </text>
      <text-style-def id="ts1"> ---> id = identifica l'estil del text
        <text-style font="Helvetica"(familia de la tipografia) fontSize="63"(mida de la
tipografia) fontFace="Regular"(tipus de tipografia) fontColor="1 0.999974 0.999991 1"(color i
transparència de la tipografia) italic="1"(Si té estil itàlic) alignment="center"(alineació)

```



```

lineSpacing="3"(Interlineat) strokeColor="0.0329639 0.00992625 7.46671e-05 1"(color i
transparència del farcit) strokeWidth="3"(ample del farcit) shadowColor="0 0 0 0.75"(color i
transparència de l'ombra) shadowOffset="5 315" (distància i angle de l'ombra)
shadowBlurRadius="1.92"(Desenfoc) />
</text-style-def>
<adjust-transform position="0 -34.6296"/> ----> Posició
</spine>
</sequence>

```

Com en la versió 1.1, si hi ha un clip de referència en el mateix codi de temps, la capa de text <title> haurà d'anar dins del <clip>.

Lectura dels subtítols – Final Cut Pro Format v 1.3

Per llegir els subtítols haurem de buscar totes les etiquetes <title> que hi hagi dins del <spine> i llegir els atributs i etiquetes següents.

- L'atribut offset dins de <title> : Codi de temps d'entrada
- L'atribut duration dins de <title>: Duració del subtítol.
- <text>: Text de cada subtítol. S'haurà de comprovar el nombre de línies.
- Si existeix l'atribut itàlic dins de <text-style>: Si el text té un estil itàlic.

No obstant, com en la versió 1.1, haurem de tenir en compte possibles problemes com:

- Que hi hagi títols que no són subtítols, poden ser títols de crèdit o altres.
- Que els subtítols s'hagin generat amb un preset diferent als que contemplem

En aquest cas, implementarem la mateixa solució i el mateix mecanisme de comprovació que en la versió 1.1, però acceptarem dos presets de text més: "Textup Regular" i "Textup Box"

Esriptura dels subtítols – Final Cut Pro Format v 1.3

En l'escriptura utilitzarem la mateixa metodologia que en la versió 1.1, l'única diferència, com és obvi, serà que el contingut de <title> serà diferent.

Paràmetres d'estil – Final Cut Pro Format v 1.3

A diferència de FCP7, FCPX permet editar els paràmetres de varis clip de text alhora. Això, provoca que la definició dels paràmetres en aquest format no sigui tant cabdal.

Per altra banda, els .fcpxml no permeten emmagatzemar tots els paràmetres que es poden definir en l'editor de textos. A més, ens hem adonat que hi ha paràmetres que un cop exportats en l'fcpxml, després, per error, no són ben llegits pels programari com per exemple l'opacitat i el desenfocament de la línia de farcit.

Per totes aquestes raons, decidim que l'usuari pugui definir els paràmetres més rellevants que poden ser exportats i importats. En el document ***Memoria/Fitxers adjunts/FCPX_fcpxml1_3_parametres.xls*** es troba un llista amb els paràmetres que l'usuari pot definir i els valors que aquests poden tenir.

Annex 3 - Altres extractes de codi

- **Creació d'un nou arxiu .srt.**

```

public function createNewFile($subtitlesList=null,
$OriginalFrameRate=null, $targetFrameRate=null) {
// COMPROVEM SI ELS FRAME RATE D'ENTRADA I DE SORTIDA SÓN DIFERENTS.
// SI HO SÓN, CANVIEM EL FRAME RATE I MODIFIQUEM ELS OBJECTES SUBTÍTOL
if (!(($targetFrameRate == 'thesame' || $targetFrameRate ==
$OriginalFrameRate))
{
    $frameRate = $targetFrameRate;
    foreach ($subtitlesList as $sub) {
        $newTcIn = $this->changeFrameRate($sub ->getTcIn(),
        $OriginalFrameRate, $targetFrameRate);

        $newTcOut = $this->changeFrameRate($sub ->getTcOut(),
        $OriginalFrameRate, $targetFrameRate);

        $ sub ->setTcIn($newTcIn);
        $ sub ->setTcOut($newTcOut);
    }
} else {
    $frameRate = $OriginalFrameRate;
}
// CREEM UN NOU FITXER
$filename = 'Subtitle_Converter_New.srt';
$file = fopen($filename,"w");
// PER CADA SUBTÍTOL, ESCRIVIM LES LINIES AL FITXER
foreach ($subtitlesList as $ sub) {
    fwrite($file, $ sub ->getId().PHP_EOL);
    fwrite($file,$this->framesToMiliseg($sub ->getTcIn(),
    $frameRate).' --> '.$this->framesToMiliseg($sub ->getTcOut(),
    $frameRate).PHP_EOL);

    // COMPROVEM SI EL TEXT ÉS EN CURSIVA ABANS D'ESCRIURE'L
    if ($sub ->getItalic() == true)
    {
        fwrite($file, '<i>'.$ sub ->getText().'</i>'.PHP_EOL);
    } else {
        fwrite($file, $ sub ->getText().PHP_EOL);
    }
    // ESCRIVIM UNA LÍNIA EN BLANC DESPRÉS DE CADA SUBTÍTOL
    fwrite($file, PHP_EOL);
}
// TANQUEM EL FITXER
fclose($file);
return $filename;
}

```

- **Lectura d'un arxiu .srt.**

```

public function getSubtitlesList()
{
    // DEFINIM LES CONSTANTS QUE S'ASSOCIARAN A CADA TIPUS DE LÍNIA
    define('LINE_ID', 0);
    define('LINE_TIME', 1);
    define('LINE_TEXT', 2);
    define('LINE_BLANK', 3);
    // ESTAT ACTUAL
    $state = LINE_ID;
    // INICIALIZEM LES VARIABLES ON DESEM LA INFORMACIÓ DE CADA SUBTÍTOL
    $subNum = 0;
    $subText = '';
    $subTime = '';

    $totalLines = count($this->fileLines);
    // RECORREM TOTES LES LÍNIES
    foreach($this->fileLines as $key => $line) {
        //COMPROVEM L'ESTAT
        switch($state) {
            case LINE_ID:
                $subNum = trim($line);
                $state = LINE_TIME;
                break;
            case LINE_TIME:
                $subTime = trim($line);
                $state = LINE_TEXT;
                break;
            case LINE_TEXT:
                //SI LA LÍNIA ESTÀ BUIDA O ÉS L'ÚLTIMA I TÉ CONTINGUT
                //CREAREM UN OBJECTE SUBTÍTOL
                if ((trim($line) == '') || ($key == $totalLines-1 &&
                !(trim($line) == ''))) {
                    // SI ÉS L'ÚLTIMA I TÉ CONTIGUT, GUARDEM EL TEXT
                    if ($key == $totalLines-1 && !(trim($line) == '')){
                        $subText .= $line;
                    }
                    // GUARDEM LA LÍNIA DE L'ID
                    $id = $subNum;
                    // SEparem LA LÍNIA DE CODI DE TEMPS EN DOS
                    $tc = explode ("-->", $subTime);
                    $tcIn = $this->milisegToFrames(trim($tc[0]),
                    $this->frameRate);
                    $tcOut = $this->milisegToFrames(trim($tc[1]),
                    $this->frameRate);
                    // COMPROVEM SI ÉS CURSIVA
                    $italic = false;
                    $cadena1 = '<i>';
                    $check1 = stripos($subText, $cadena1);

```

```

        $cadena2 = '</i>';
        $check2 = stripos($subText, $cadena2);
        if ($check1 !== false || $check2 !== false )
        {
            $italic = true;
        }
        // GUARDEM EL TEXT COMPROVANT LA CODIFICACIÓ
        if ($this->charset != 'utf-8') {
            $text=iconv($this->charset,'utf-8//IGNORE',
                trim(strip_tags($subText)));
        } else {
            $text = trim(strip_tags($subText));
        }
        // CREEM UN NOU OBJECTE SUBTITLE DINS DE L'ARRAY
        $this->subtitlesList[] = new Subtitle($id,
            $tcIn, $tcOut, $text, $italic);
        // INICIALIZEM LA VARIABLE TEXT I L'ESTAT
        $subText = '';
        $state = LINE_ID;
    } else {
        // SI LA LÍNIA NO ESTÀ BUIDA I NO ÉS L'ÚLTIMA
        // AFEGIM EL TEXT A LA VARIABLE
        $subText .= $line;
    }
    break;
}
}
return $this->subtitlesList;
}

```

- **Lectura d'un arxiu .xml.**

```

public function getSubtitlesList()
{
    // ACCEDIM A TOTS ELS NODES TRACK
    $trackNumbers = $this->xmlSimple->sequence->media->video->track;
    $count = 1;
    // RECORREM TOTS ELS NODES TRACK
    foreach ($trackNumbers as $track) {
        // DINS DE CADA TRACK, RECORREM CADA GENERATORITEM
        foreach ($track->generatoritem as $generatoritem) {
            // SI EL GENERATOR ÉS UN PRESET DE TEXT, HI ACCEDIM
            if ($generatoritem->effect->effectid == 'Outline Text' ||
                $generatoritem->effect->effectid == 'Text'){
                //RECUPEREM TOTES LES DADES DEL SUBTÍTOL
                //L'ID
                $id = $count;
                //EL CODI DE TEMPS
                $tcIn = $this->framesToTc($generatoritem->start,

```

```
        $this->frameRate);
        $tcOut = $this->framesToTc($generatoritem->end,
        $this->frameRate);
        //EL TEXT
        if (strtolower($this->charset) != 'utf-8') {
            $text = iconv($this->charset, 'utf-8//IGNORE',
            trim($generatoritem->effect->parameter[1]->value));
        } else {
            $text=trim($generatoritem->effect->parameter[1]
            ->value);
        }
        // EL SUBTÍTOL ÉS CURSIVA
        if ($generatoritem->effect->parameter[3]->value == '3'){
            $italic = true;
        } else {
            $italic = false;
        }
        // CREEM UN NOU OBJECTE SUBTÍTOL
        $this->subtitlesList[] = new Subtitle($id, $tcIn,
        $tcOut, $text, $italic);
        $count += 1;
    }
}
}
return $this->subtitlesList;
}
```

Annex 4 - Programari i eines utilitzades

- SublimeText. Creació i desenvolupament del codi.



- Microsoft Word. Creació del contingut textual.

2.2 Diagrames UML

2.2.1 Casos d'ús

En la nostra aplicació, l'actor principal sempre és l'usuari. A continuació, es mostra un diagrama de casos d'ús que simplifica els tres casos d'ús principals: el login, l'edició de subtitols i l'exportació tant a un arxiu de subtitols com a un arxiu de text.

Figura 1. Diagrama d'ús general.

Per no estendre'ns massa, deixarem de banda els casos d'ús de nivell tasca més habituals com Login, registrar-se o recuperar la clau i ens centrarem en el cas d'importar un arxiu de subtitols. Aquest cas d'ús, inclou dos subcasos: per una banda, penjar un arxiu de subtitols i, per l'altra, seleccionar la codificació textual. En el segon cas, la idea és que el sistema sigui capaç de trobar la codificació de l'arxiu, si no és possible, el tenim en compte que estem creant una aplicació professional, demanarem la codificació textual a l'usuari.

CAS D'ÚS: Importar un arxiu de subtitols

Actor principal: Usuari.

Nivell: Tasca.

Àmbit: Sistema.

Escenari principal d'èxit:

1. El sistema mostra la pantalla d'importació
2. L'usuari penja un arxiu de subtitols
3. L'usuari selecciona la codificació textual de l'arxiu
4. El sistema comprova que l'extensió de l'arxiu, llegeix l'arxiu i recupera el seu contingut.

Extensions:

4b. El sistema troba un error en la comprovació

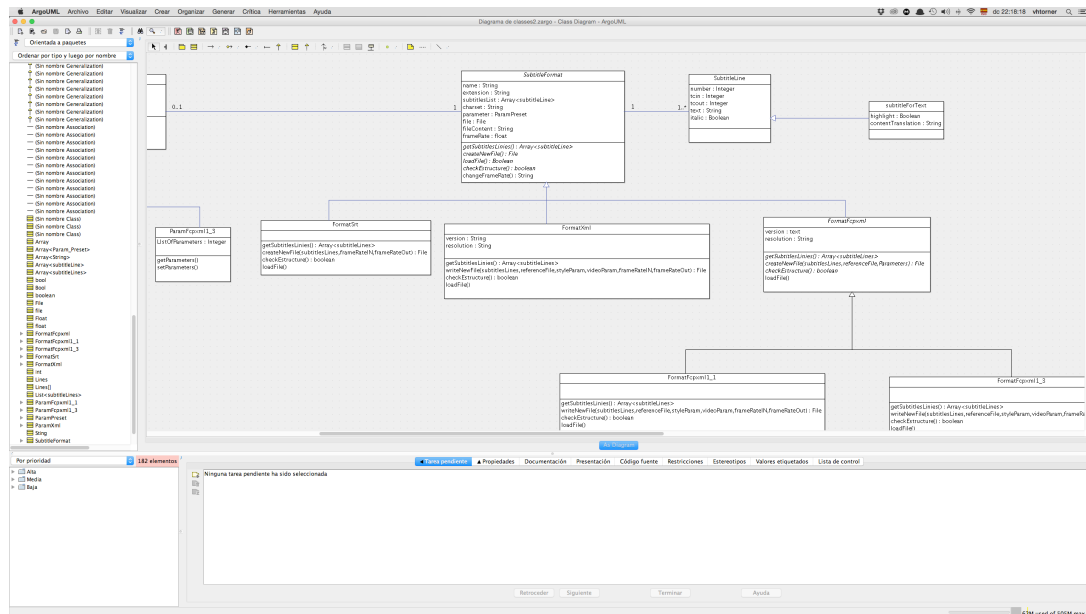
4b.1 Es mostra un missatge d'error específic i tornem al punt 1

Els casos d'edició de subtitols, exportar arxiu de subtitols i exportar arxiu de text són casos molt generals que inclouen i estenen altres casos d'ús més concrets. Per fer-ho més comprensible, separarem els casos relacionats amb l'edició dels d'exportacions.

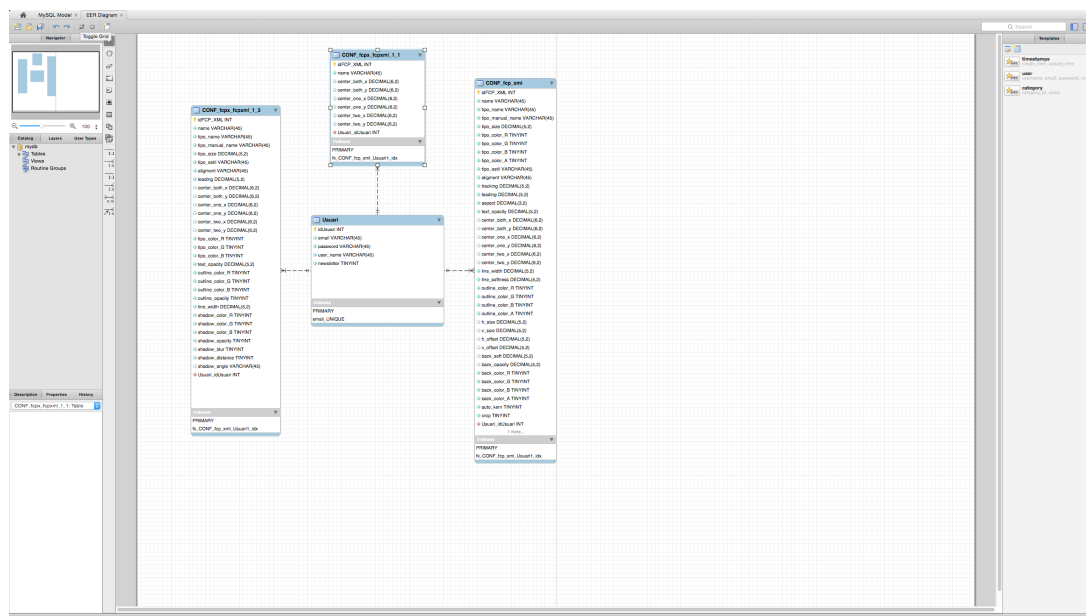
Figura 2. Diagrama d'ús de l'edició de subtitols.

Un cop importat un arxiu, el sistema mostra un llistat amb la informació dels subtitols a la finestra d'edició. En aquest punt, és on l'usuari pot començar a editar la informació dels seus subtitols. El cas d'ús d'edició de subtitols bàsicament

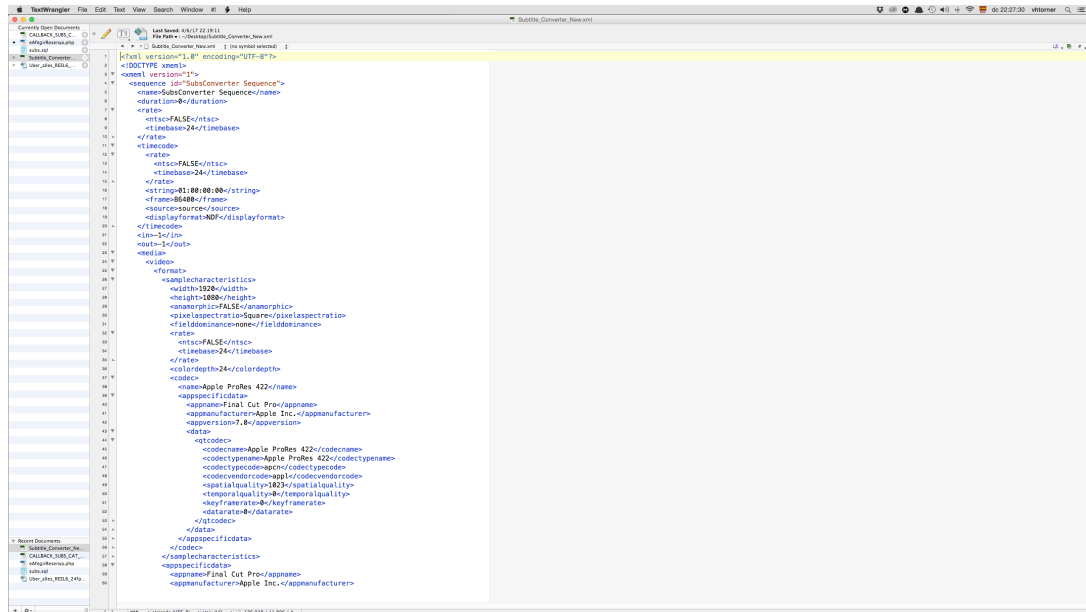
- **ArgoUML.** Creació dels diagrames UML.



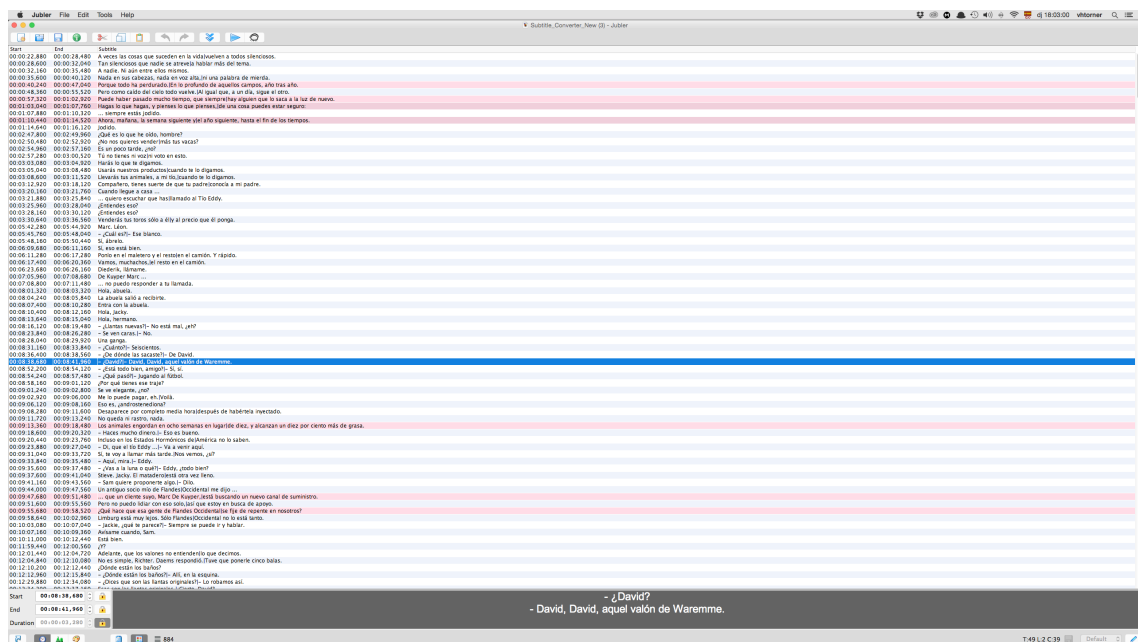
- **MySQLWorkBench.** Creació de les taules i del model relacional de la base de dades.



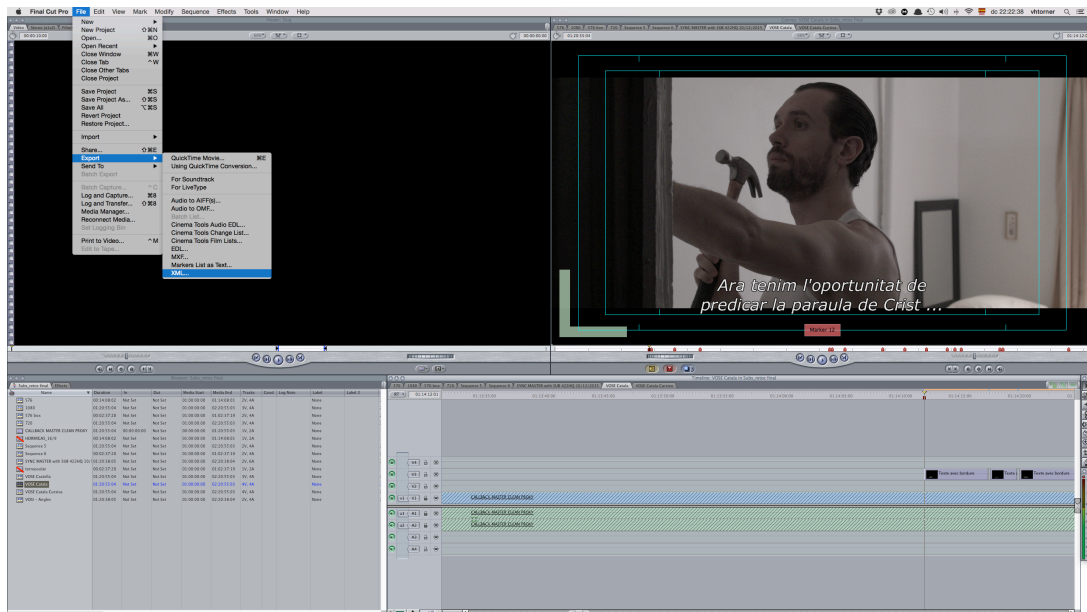
- **Text Wrangler.** Comprovació de l'estructura dels fitxers XML.



- **Jubler.** Comprovació de l'estructura dels fitxers SRT.



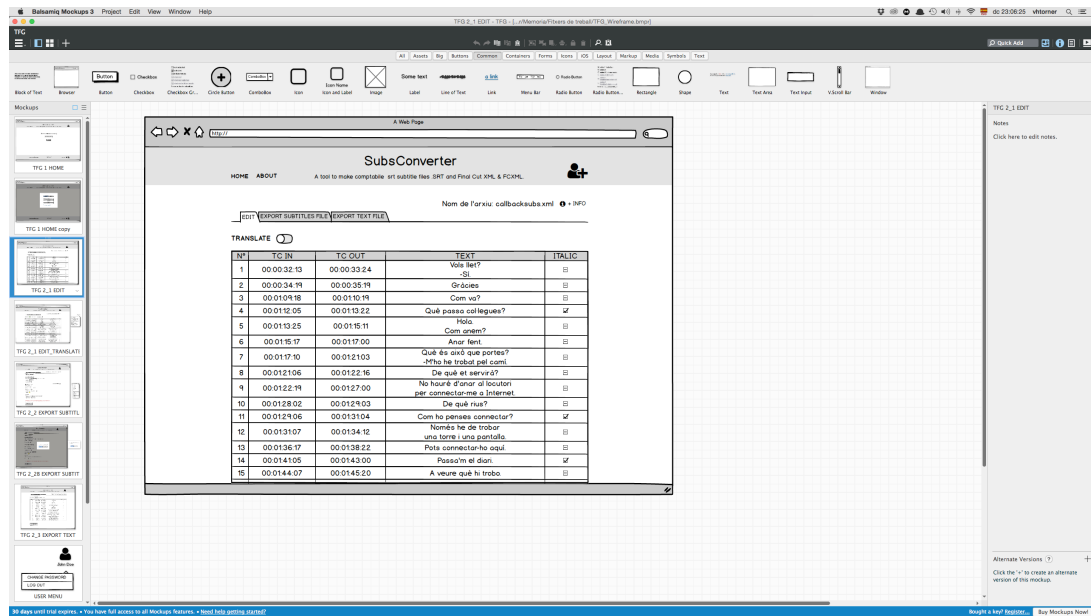
- **Final Cut Pro 7.** Creació d'arxius XML i comprovació del funcionament dels arxius exportats.



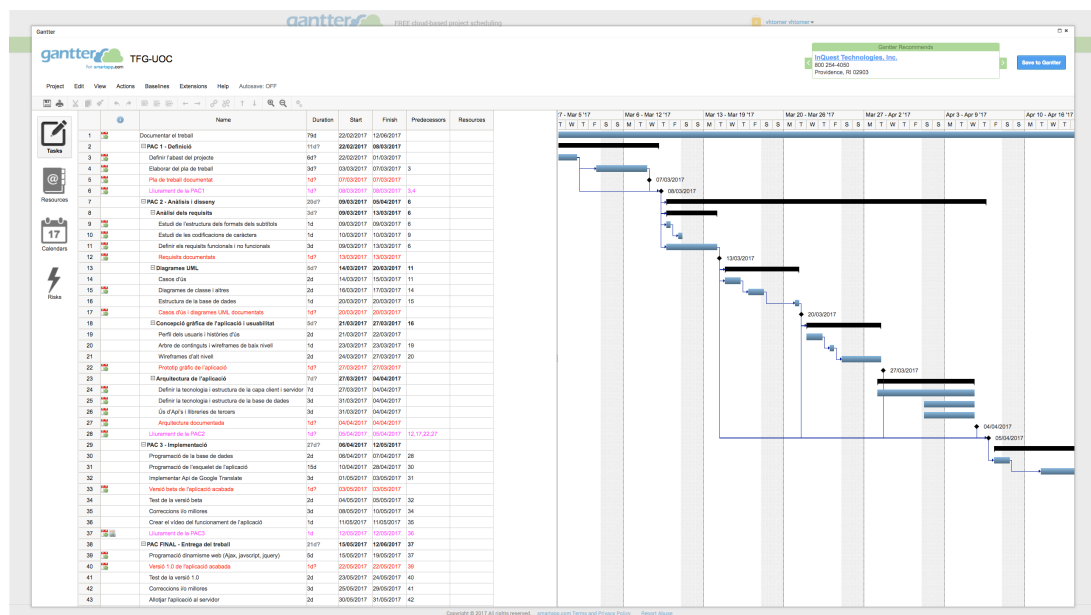
- **Final Cut Pro X versions.** Creació d'arxius FCPXML i comprovació del funcionament dels arxius exportats.



- Balsamiq Mockup 3. Creació dels wireframes.



- Ganttter.com. Creació del diagrama de Gantt.



- **Adobe Photoshop.** Creació del Wall concept.

