

# Implementació d'un sistema d'emmagatzematge de Documents per al nou projecte WEB de SPI

Cristian Pérez Rodríguez

**Resum** — En aquest treball es desenvoluparà una nova funcionalitat per a una aplicació GMAO[1] WEB que està creant l'empresa "Sistemas de Publicaciones Informáticas, SL." Aquesta aplicació està enfocada al manteniment i la gestió interna d'una empresa. Principalment, es centra en el manteniment rutinari, com ara la gestió d'estocs d'un magatzem o la generació d'albarans, però també s'encarrega de l'assignació de tasques per als seus treballadors i la realització d'aquestes. Aquesta aplicació WEB es desenvolupa sobre un framework anomenat Durandal, el qual està dissenyat per generar SPAs per a la part del client. Pel que fa al servidor, s'utilitza una API en .Net que fa servir un Entity Framework per gestionar la base de dades que hi ha al SQL Server. En relació al disseny, tant la part del servidor com la del client, utilitzen sempre un patró MVC. D'altra banda, cada secció d'aquesta aplicació GMAO WEB està composta per 2 apartats: el primer, és una llista principal on es mostren els elements X i, el segon, una secció on es mostra la informació d'un element seleccionat i on es pot interactuar amb ell. La nova funcionalitat que es presenta en aquest treball consisteix en un sistema de manteniment de documents que serà compartit per a totes les seccions i que tindrà la seva pròpia secció per a la gestió de les etiquetes assignades als documents. Aquesta nova funcionalitat permetrà carregar, eliminar i modificar múltiples documents, alhora que també facilitarà l'assignació d'una o múltiples etiquetes per a la seva classificació.

**Paraules clau** – Durandal, Manteniment de Documents, Gestió d'etiquetes, ASP.NET Web API.

**Abstract** — In this project, a new functionality will be developed for a GMAO WEB application created by the "Sistemas de Publicaciones Informáticas, SL." company. This application is focused on a company's maintenance and internal management. Principally, it focuses on routine maintenance, such as the inventory management of a warehouse or the delivery notes' generation, but it is also responsible for assigning tasks to its workers and the performance of them. This WEB application is developed on a framework called Durandal, which is designed to generate SPAs for the customer part. For the server, a .Net API is used that uses an Entity Framework to manage the database on SQL Server. In relation to the design, both the server side and the client side always use an MVC pattern. On the other hand, each section of this GMAO WEB application is composed of 2 sections: the first, is a main list where the X elements are displayed and, the second, a section where the information of a selected element is displayed and where you can interact with it. The new functionality presented in this work consists of a documents maintenance system that will be shared for all sections and will have its own section for label managing of the labels assigned to documents. This new feature will allow you to load, delete and modify multiple documents, while also making it easier to assign one or more labels for classification.

**Key words** – Durandal, Document Maintenance, Label Managing, ASP.NET Web API.

---

## 1 INTRODUCCIÓ

**A**CTUALMENT l'empresa treu tots els seus beneficis de la comercialització d'un únic software de gestió anomenat LINX el qual està desenvolupat en Delphi i únicament apte per entorns Windows.

Delphi es un llenguatge antic que cada vegada s'aprèn menys en el món laboral, poc a poc s'està quedant obsolet, es poc sostenible i cada cop els programadors Delphi estan més cotitzats.

També presentava un altre gran inconvenient i era que no-mes tenien una versió, la qual era massa completa i cara tant de preu de compra com el preu de manteniment anual, això feia que no es poguessin expandir entre les petites empreses les quals mes de la meitat de les funcionalitats no les utilitzarien.

Davant aquesta situació de mercat, l'empresa va prendre la decisió de desenvolupar de nou una aplicació amb tecnologies actuals, amb una major facilitat per poder-la mantenir, actualitzar, millorar, amb una versió paral·lela mes reduïda per a les petites empreses i que a la vegada sigues multi plataforma.

Durant l'últim any hem estat desenvolupant aquesta nova aplicació i farà uns mesos hem facilitat a uns quants clients una versió alfa per a provar i tenir un feedback real amb els clients.

D'aquests clients hem rebut diverses recomanacions o peticions per a implementar en futures versions i hem pogut observar que la petició mes repetida es la necessitat que tenen de poder assignar documents a qualsevol tipus d'element de l'aplicació, el qual ha donat pas a poder realitzar aquesta proposta de TFG per tal d'implementar-la.

- 
- E-mail de contacte: [cristian.perezr@e-campus.uab.cat](mailto:cristian.perezr@e-campus.uab.cat)
  - Menció realitzada: *Enginyeria del Software.*
  - Treball tutoritzat per: *Marc Talló (departament)*
  - Curs 2020/21

## 2 ESTAT DE L'ART

Actualment existeixen bastantes plataformes GMAO en el mercat, de les gratuïtes que he pogut provar personalment com exemple "OutSmart<sup>[2]</sup>", la gestió de documents es molt bàsica i només tenen opció d'afegir i eliminar.

Desafortunadament les plataformes més competitives tenen un cost molt elevat (com la nostra) i només ens hem pogut guiar de les explicacions que ens han facilitat els nostres clients que després d'utilitzar varies plataformes GMAO han acabat utilitzant la nostra i tots coincidien en el mateix, actualment cap GMAO té la funcionalitat de classificar els documents per a diferenciar-los.

Així doncs podríem dir que el desenvolupament del sistema de gestió de documents que plantegem podria ser una part decisiva per als futurs clients a l'hora de decantar-se per la nostra plataforma.

## 3 OBJECTIUS

L'objectiu principal del projecte consisteix en la implementació d'un sistema de manteniment de fitxers per a cada element de l'aplicació.

A continuació es presenten la resta d'objectius ordenats de forma prioritària de manera que els primers són els més importants per a realitzar el projecte.

### 3.1 Objectius principals

- I. Creació del llistat de documents amb la gestió mínima d'afegir i eliminar.
- II. Creació bàsica de la secció de gestió d'etiquetes.
- III. Afegir l'opció al llistat de documents de gestionar les etiquetes de cada document.
- IV. Afegir l'opció en la secció de gestió d'etiquetes de tenir etiquetes pare / fill i visualitzar les etiquetes en forma d'arbre.
- V. Afegir el mode de visualització/navegació al llistat de document com si les etiquetes fossin carpetes com es fa a Google Drive.

### 3.2 Objectius secundaris

- I. Visualitzar cada document de tipus imatge afegit al llistat de documents en un carrusel d'imatges amb la possibilitat de veure-les en gran.
- II. Possibilitat d'ordenació d'aquestes imatges per a la previsualització en ordre.

## 3.3 Criticitat

Objectius	Criticitat
<b>Objectius Principals</b>	
I	
II	
III	
IV	
V	
<b>Objectius Secundaris</b>	
I	
II	

Alta	Mitja	Baixa
------	-------	-------

## 4 ANÀLISIS DE REQUERIMENTS

### 4.1 Funcionals

- A la web en la secció de gestió d'un element s'ha de mostrar una nova pestanya per la gestió dels documents d'aquest element.
- Cada element tindrà els seus propis documents independents.
- La pestanya de gestió de documents ha de mostrar un llistat dels documents de l'element actual.
- Al llistat de documents hi ha d'haver un boto per afegir un document local.
- Al afegir un document al llistat de documents d'un element se li ha de poder assignar una o múltiples etiquetes.
- Al llistat de documents d'un element al clicar sobre un document s'ha de poder descarregar.
- Al llistat de documents s'ha de poder esborrar un document.
- Al llistat de documents d'un element, per a cada document s'ha de mostrar el tipus de document, el nom del document i totes les seves etiquetes.
- A la web hi ha d'haver una pàgina sencera de gestió d'etiquetes de fitxers.
- A la pàgina de gestió d'etiquetes de fitxers s'han de poder crear etiquetes.
- A la pàgina de gestió d'etiquetes de fitxers s'han de poder eliminar etiquetes.
- A la pàgina de gestió d'etiquetes de fitxers s'han de poder modificar etiquetes.
- L'etiqueta ha de tenir un codi d'identificació.
- L'etiqueta ha de tenir un nom que ha de ser multi llenguatge.
- Una etiqueta ha de poder ser filla d'un altra etiqueta.
- A la pàgina de gestió d'etiquetes la visualització d'aquestes ha de ser en mode d'arbre per una millor

visualització de les famílies.

- Al llistat de documents d'un element, s'ha de poder visualitzar com si les etiquetes fossin carpetes.
- Al llistat de documents d'un element, en el mode de visualització de les etiquetes com a carpetes, s'ha de poder navegar entre carpetes.

## 4.2 No funcionals

- Els documents es guardaran físicament al servidor sense el seu nom original, es guardaran amb el nom del seu identificador de la tupla de la BD.
- La informació bàsica de cada document quedarà gravada a una taula de la BD.
- La informació de les etiquetes quedarà gravada a una taula de la BD.
- El camp de la taula de la BD que guardarà el nom de l'etiqueta es guardarà en estructura JSON en tots els idiomes de l'empresa.
- La relació entre un document, les seves etiquetes i l'element associat es guardarà en una altra taula de la BD.
- A l'afegir un document ja existent al servidor però a un element diferent, per tal de no ocupar més espai del necessari al servidor, no s'afegirà al servidor i compartiran el mateix document.
- A l'eliminar un document d'un element que està sent compartit per més elements, no s'eliminarà físicament, només s'eliminarà la relació amb aquest element.
- A l'eliminar un document d'un element que no està sent compartit, s'eliminarà físicament del servidor i les seves tuples de relació a la BD.

## 4.3 Tècnics

- Servidor extern on estarà corrent l'API on es faran les peticions des de l'aplicació client.
- El servidor extern ha de tenir una bona capacitat d'emmagatzematge per a poder guardar els documents.
- El servidor haurà de córrer sobre un sistema Windows per a poder córrer l'api sobre un IIS.
- La base de dades haurà de ser SQL Server versió 19 o superior.
- El servidor haurà de tenir instal·lat Microsoft .NET Framework 3.5 o superior.
- El servidor haurà de poder acceptar peticions des de l'exterior de la seva xarxa.

## 5 METODOLOGIA

La metodologia escollida per a dur a terme aquest projecte ha sigut el model incremental<sup>[3]</sup>, explicada anteriorment a l'assignatura de Sistemes d'Informació i que consisteix en

dividir el projecte en objectius que s'aniran assolint en terminis de 2 setmanes.

Aquesta metodologia és la més adient per aquest projecte ja que s'aniran fent reunions periòdiques de seguiment fent referència amb els objectius marcats per a cada termini amb el tutor de l'empresa i anar veient l'evolució, si son requerides altres funcionalitats i amb la possibilitat d'adaptació a nous canvis i requeriments. Cada termini anirà relacionat directament amb cada un dels punts dels objectius anomenats anteriorment.

Per a portar un control d'aquests terminis, farem servir el nostre mateix programa GMAO Linx en el qual se m'assignaran un nombre d'ordres de treball prevists per a realitzar cada objectiu i on hauré d'anar imputant les hores de feina realitzades en cada iteració.

Per a poder mantenir un control del codi, també usarem l'eina Git<sup>[4]</sup> tant per la part del client com per la part del servidor.

### 5.1 Control de versions

Per a que el desenvolupament del projecte de software sigui satisfactori, ha sigut necessari utilitzar un mètode de control de totes les versions generades durant el desenvolupament.

#### 5.1.1 GIT

En aquest projecte, s'ha utilitzat GIT, un software de control de versions pensat per a fer possible el desenvolupament i manteniment de codi. En aquest projecte, principalment l'ús que se li ha donat, ha sigut el de poder recuperar antics canvis, d'aquesta manera reduïm l'impacte que pot tenir esborrar codi important, instal·lació de llibreries que posteriorment donen peu a errors de compilació, etc. Concretament s'ha dut a terme el disseny "Git-flow"<sup>[5]</sup>, un model de ramificació estricta dissenyat per a estructurar el control de versions de forma eficient amb GIT.

Aquest model de ramificació consisteix en l'ús de dues branques anomenades "Màster" on trobem tots els commits de producció i "Develop" on hi hauran les versions del software que estan encara en desenvolupament i que en un futur formaran part de producció. A més d'aquestes dues branques, es proposen les següents branques auxiliars:

- Feature: neix de "Develop" i que està destinada a afegir noves característiques i funcionalitats al software.
- Release: utilitzada per incorporar el codi que sortirà a producció. Aquesta branca neix a "Develop" per a incorporar-se un cop finalitzada a "Màster".
- Hotfix: es fan servir per corregir errors existents a la branca "Master".

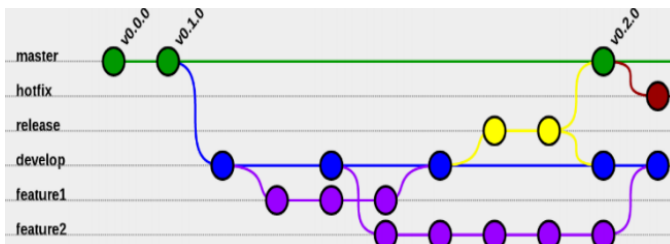


Figura 1: Exemple de flux amb GitFlow WorkFlow.

## 5.2 Model incremental

S'ha escollit aquest model ja que te com a objectiu un creixement progressiu de les funcionalitats del projecte, nombrades anteriorment com objectius a assolir (funcionalitats), combina elements del model en cascada amb la filosofia interactiva de construcció de prototips.

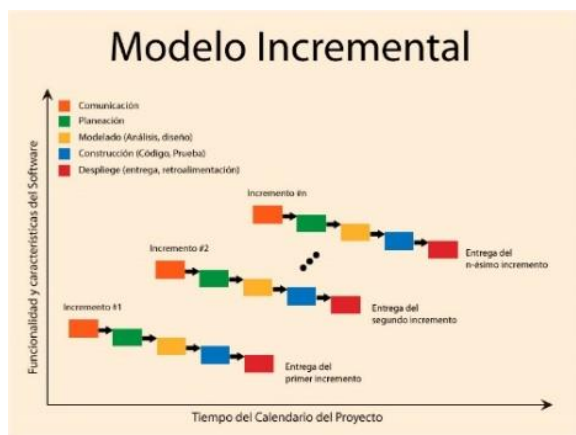


Figura 2: Exemple model Incremental.

Aquest model aplica seqüències lineals de forma escalonada mentre es progressa en la planificació, cada seqüència lineal produeix un increment en el projecte, es centra en l'entrega d'un producte operatiu a cada increment.

### Veneficis:

- Els increments són petits el qual faciliten la seva depuració i test.
- Faciliten l'administració de les tasques a cada increment.
- La inversió de temps en el projecte es materialitza més ràpidament, lo qual si el client vol veure com va el desenvolupament es fàcilment demostrable.
- Queda registre en el control de versions molt clar de les funcionalitats desenvolupades.

- S'adapta fàcilment a les necessitats que puguin anar sortint a cada iteració.

## 5.3 Control dels increments

Per a realitzar un bon seguiment dels objectius a assolir durant cada increment, hem fet us de la pròpia aplicació de l'empresa anomenada GMAO Linx, la qual s'està intentant desenvolupar una versió multi plataforma en format WEB ja que l'anterior estava desenvolupada en Delphi i només era compatible amb sistemes Windows.

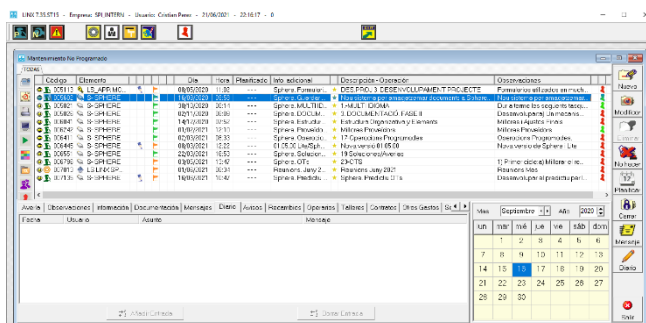


Figura 3: Exemple d'assignació d'increment a GMAO Linx.

## 6 SOFTWARE UTILITZAT

Per l'implementació d'aquest projecte s'han utilitzat diverses eines.

### Visual Studio Code

Es tracta d'un editor de codi font desenvolupat per Microsoft molt sofisticat que admet moltes funcionalitats pràctiques a l'hora de treballar amb el codi i que porta un control integrat de Git. Algunes d'aquestes funcionalitats són:

- Multiplataforma: creat i dissenyat per funcionar tant en Windows, Linux com Mac OS.
- Plugins: possibilitat d'afegir una gran quantitat de plugins que ens faciliten la codificació.
- Open Source: podem trobar aquesta eina a GitHub, per tant, podem descarregar, analitzar i modificar el projecte.
- Intellisense: l'editor té la capacitat de preveure la instrucció que volem escriure i auto completar-la. Això ens fa més productius a més de reduir la possibilitat de tenir errors de sintaxis.
- Consola de comandes integrada que permet treballar amb més comoditat.

S'ha utilitzat per al desenvolupament de la part del client del projecte.

### Microsoft Visual Studio

Es un entorn de desenvolupament integrat amb nombroses característiques que es poden usar per molts aspectes del desenvolupament del software. En aquest cas per a poder editar, depurar y compilar el codi i finalment publicar el desenvolupament en una aplicació. Igual que el Visual Studio Code, també esta desenvolupat per Microsoft i per tant comparteixen característiques molt útils com pot ser l'IntelliSense.

### Microsoft SQL Server Management Studio

Es un entorn integrat per l'administració de qualsevol infraestructura SQL. Proporciona eines per a la configuració, supervisió i l'administració d'instàncies d'SQL Server i la consulta, el disseny i l'administració de bases de dades tant en local com a través d'Internet.

## 7 TECNOLOGIES UTILITZADES

S'han utilitzat diverses tecnologies per a poder estructurar l'aplicació.

### Back End

#### *Servidor WEB IIS (Internet Information Services)*

Es tracta d'un servidor web de Microsoft Windows destinat a oferir serveis, que en el nostre cas seran tots HTTP/HTTPS mitjançant un entorn ASP.NET, creant una API Web. La principal pega es que només es pot utilitzar mitjançant un sistema Windows.

#### *ASP.NET Web API*

Es un marc per crear una API Web sobre el framework de Microsoft .NET.

#### *SQL Server*

Es tracta d'un sistema de gestió de bases de dades relacionals de Microsoft que està dissenyat per a l'entorn empresarial. SQL Server s'executa en T-SQL (Transact-SQL), un conjunt d'extensions de programació de Sybase i Microsoft que afegeixen diverses característiques a SQL estàndard, incloent control de transaccions, excepció i maneig d'errors, processament fila, així com variables declarades.

### Front End

#### *Durandal*

Es un petit framework dissenyat per muntar SPAs (Single Page Application) simple i elegant el qual té un gran suport a MVC, MVP i MVVM.

Durandal pot proporcionar una productivitat espectacular mentre que t'ajuda a mantenir unes practiques de codificació solides, gracies a que es fa valer de RequireJs com a base y una capa molt fina de convencions a seguir.

## 8 DISSENY I EXECUCIÓ DE L'APLICACIÓ WEB

Descriurem les principals característiques relacionades amb el desenvolupament del projecte.

### 8.1 Requeriments

#### *Funcionals*

- A la web en la secció de gestió d'un element s'ha de mostrar una nova pestanya per la gestió dels documents d'aquest element.
- Cada element tindrà els seus propis documents independents.
- La pestanya de gestió de documents ha de mostrar un llistat dels documents de l'element actual.
- Al llistat de documents hi ha d'haver un boto per afegir un document local.
- Al afegir un document al llistat de documents d'un element se li ha de poder assignar una o múltiples etiquetes.
- Al llistat de documents d'un element al clicar sobre un document s'ha de poder descarregar.
- Al llistat de documents s'ha de poder esborrar un document.
- Al llistat de documents d'un element, per a cada document s'ha de mostrar el tipus de document, el nom del document i totes les seves etiquetes.
- A la web hi ha d'haver una pàgina sencera de gestió d'etiquetes de fitxers.
- A la pàgina de gestió d'etiquetes de fitxers s'han de poder crear etiquetes.
- A la pàgina de gestió d'etiquetes de fitxers s'han de poder eliminar etiquetes.
- A la pàgina de gestió d'etiquetes de fitxers s'han de poder modificar etiquetes.
- L'etiqueta ha de tenir un codi d'identificació.

- L'etiqueta ha de tenir un nom que ha de ser multi llen-guatge.
- Una etiqueta ha de poder ser filla d'un altra etiqueta.
- A la pàgina de gestió d'etiquetes la visualització d'aquestes ha de ser en mode d'arbre per una millor visualització de les famílies.
- Al llistat de documents d'un element, s'ha de poder visualitzar com si les etiquetes fossin carpetes.
- Al llistat de documents d'un element, en el mode de visualització de les etiquetes com a carpetes, s'ha de poder navegar entre carpetes.

### No funcionals

- Els documents es guardaran físicament al servidor sense el seu nom original, es guardaran amb el nom del seu identificador de la tupla de la BD.
- La informació bàsica de cada document quedarà gravada a una taula de la BD.
- La informació de les etiquetes quedarà gravada a una taula de la BD.
- El camp de la taula de la BD que guardarà el nom de l'etiqueta es guardarà en estructura JSON en tots els idiomes de l'empresa.
- La relació entre un document, les seves etiquetes i l'element associat es guardarà en una altra taula de la BD.
- A l'afegir un document ja existent al servidor però a un element diferent, per tal de no ocupar més espai del necessari al servidor, no s'afegirà al servidor i compartiran el mateix document.
- A l'eliminar un document d'un element que està sent compartit per més elements, no s'eliminarà físicament, només s'eliminarà la relació amb aquest element.
- A l'eliminar un document d'un element que no està sent compartit, s'eliminarà físicament del servidor i les seves tuples de relació a la BD.

### Tècnics

- Servidor extern on estarà corrent l'API on es faran les peticions des de l'aplicació client.
- El servidor extern ha de tenir una bona capacitat d'emmagatzematge per a poder guardar els documents.
- El servidor haurà de córrer sobre un sistema Windows per a poder córrer l'api sobre un IIS.
- La base de dades haurà de ser SQL Server versió 19 o

superior.

- El servidor haurà de tenir instal·lat Microsoft .NET Framework 3.5 o superior.
- El servidor haurà de poder acceptar peticions des de l'exterior de la seva xarxa.

## 8.2 Estructura de l'aplicació

El sistema està format per un Servidor WEB (IIS) [6] que conté un Client WEB amb accés públic.

El servidor utilitza un entorn ASP.NET que exposa una sèrie de serveis Web API per realitzar les gestions de la BD i dels Documents mitjançant crides API REST[7].

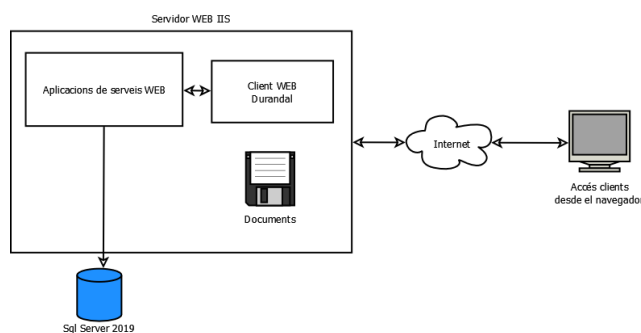


Figura 4: Estructura aplicació.

## 8.3 Disseny de la base de dades

Estructura relacional de les taules de la base de dades:

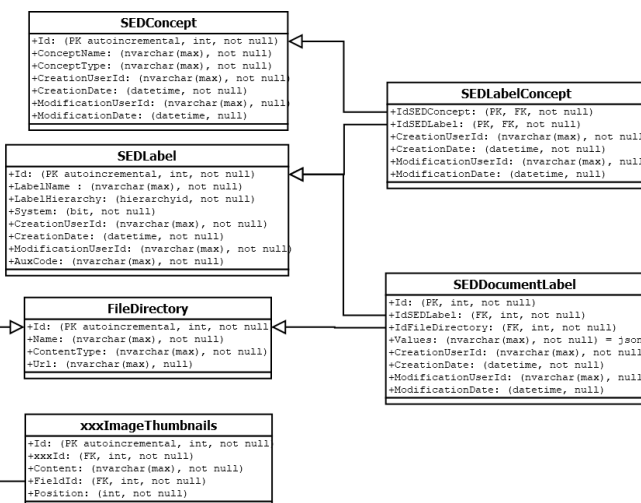


Figura 5: Estructura BBDD



### 8.3.1 Definició de les taules

- **SEDConcept:** S’afegiran conceptes on haurem de guardar el nom del concepte (que després servirà per montar el json a la columna “Values” de la taula “SEDDocumentLabel”) i de quin tipus de concepte serà (boolea, numèric, text, etc).
- **SEDLabel:** S’afegiran les Etiquetes introduint el nom de l’etiqueta, la jerarquia que tindrà en l’arbre general d’etiquetes i si es una etiqueta de Sistema o no.
- **SEDLabelConcept:** Relació entre cada Etiqueta i els Conceptes que conte cada etiqueta.
- **FileDirectory:** Taula on guardarem el nom del fitxer i el seu tipus.
- **SEDDocumentLabel:** Informació final per a cada document, conte la relació amb l’Etiqueta i els seus valors dels conceptes guardat en un camp de text amb estructura “json”.
- **xxxImageThumbnails:** Relació dels thumbnails amb els fitxers originals.  
Taula d’emmagatzament de les imatges comprimides(thumbnails) en format base64, també guarda l’ordenació de la previsualització.  
Es crearà una taula per a cada secció de la web on es vulgui utilitzar aquesta funcionalitat.  
Ex: *SparePartImageThumbnails*

## 9 RESULTATS

En aquesta secció es mostra de manera visual els resultats obtinguts, primer es mostraran els resultats de l’administració de les etiquetes i seguidament el funcionament de la previsualització dels arxius de tipus imatge. En aquest cas utilitzarem la secció de recanvis per a realitzar la implementació de la nova funcionalitat d’administració de documents. Per implementar les funcionalitats en un element, simplement hem d’anar al model d’aquest i afegir les 3 funcionalitats que hem desenvolupat. En la següent figura veurem les 3 funcionalitats afegides al model i quina secció de la pantalla generarà automàticament durant la visualització de l’element. Les 3 funcionalitats es complementen i es pot optar per no utilitzar les que no es necessitin amb la restricció de que una posterior es necessita de la anterior per a funcionar. Es podria optar per no utilitzar ni l’ordenació ni el visor de thumbnails però no es podria optar per utilitzar l’ordenació sense la visualització dels thumbnails.

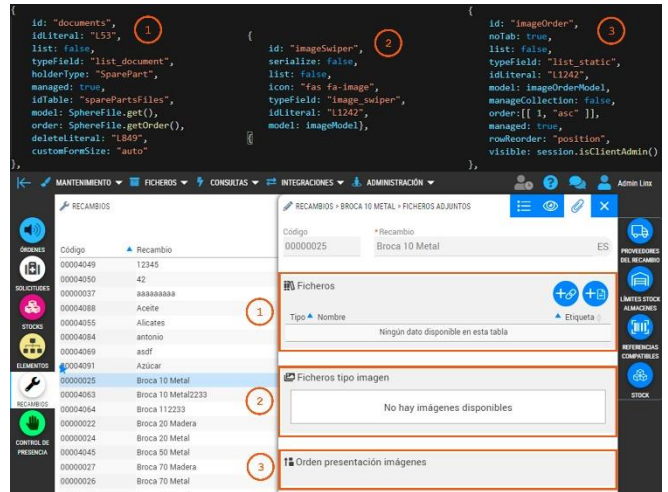


Figura 6: Implementació al model del recanvi i la seva visualització gràfica.

Com podem observar, cada funcionalitat té un “TypeField” diferent. Aquest atribut és el que determina la lògica i la visualització gràfica que ha de tenir aquest camp quan es generi per pantalla.

Per altra part esta la secció de l’administració de les etiquetes en la qual s’ha utilitzat la llibreria Fancytree[11] per aconseguir una visualització de les etiquetes de manera jeràrquica.

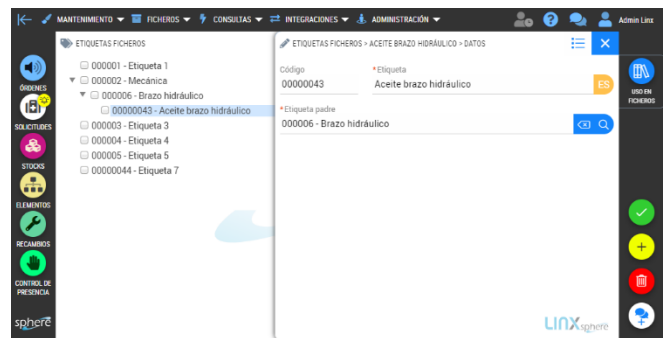


Figura 7: Visualització de l’administració d’etiquetes.

Una vegada mostrades les 2 gran parts principals de l’implementació del projecte podem mostrar un exemple pràctic d’us.

### Adjuntar un fitxer

Al adjuntar un fitxer, podem veure com sens mostra una icona per a poder assignar etiquetes al document que a posteriori també ens ajudaran a mostrar els arxius adjunts visualitzant-los depenentment de la jerarquia de l’etiqueta a mostrar-los com si fossin carpetes, tal i com realitza Google Drive.



Figura 8: Arxiu adjuntat.

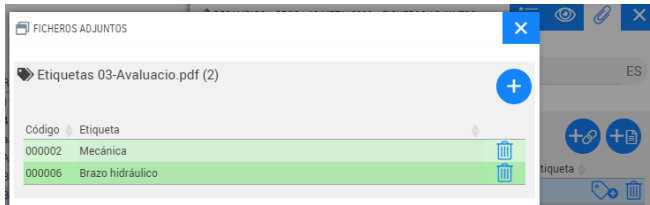


Figura 9: Selecció d'una o varies etiquetes.

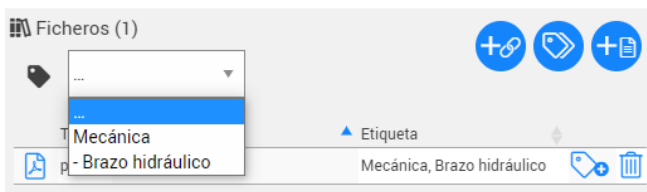


Figura 8: Mode de visualització de navegació per "carpetes"

### Adjuntar varis documents format d'imatge

Al adjuntar imatges, una vegada ja guardades al servidor, quan tornem a demanar l'informació del element, ens mostra una previsualització de les imatges amb una resolució reduïda per evitar grans temps de resposta en les peticions i que tot i així es poden seleccionar i visualitzar en gran resolució fent una petició de l'arxiu en concret. També ens dona l'opció d'alterar l'ordre en que es vol que es guardin les imatges per a la seva previsualització ja que aquests apartats es solen utilitzar per guardar imatges de la realització d'un procés concret o guia.

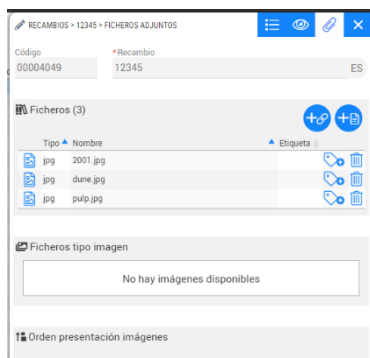


Figura 9: Imatges adjuntes abans de guardar l'informació.

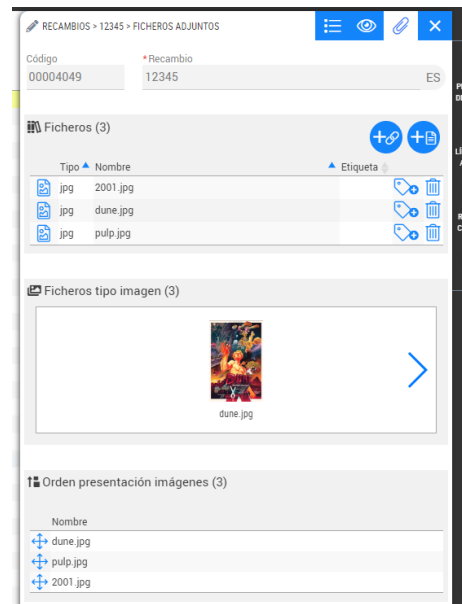


Figura 10: Informació rebuda al demanar l'informació del element una vegada ja s'han guardat les imatges al servidor.

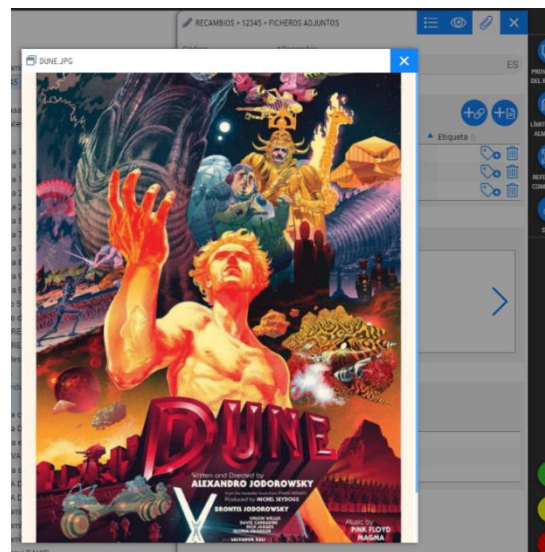


Figura 11: Visualitzador d'imatges.

## 10 PROBLEMES SUPERATS

Per sort o per desgràcia, durant el desenvolupament no ens hem trobat amb gaires problemes més enllà del desconeixement del funcionament de certes llibreries / tecnologies a l'hora d'implementar-les al projecte que han requerit d'un termini d'aprenentatge previ que han suposat una demora de temps considerable i un perfecte exemple podria ser l'implementació dels testos de Karma<sup>[8]</sup> per a la part del client, la llibreria Fancytree<sup>[9]</sup> per fer visualitzacions en forma d'arbre o el pluguin de JQuery anomenat Datatables<sup>[10]</sup> per a la visualització i tractament de dades en forma de taules.



Per altra part, un dels problemes mes importants que s'han hagut de solucionar ha sigut durant l'implementació de l'objectiu secundari "Visualitzar cada document de tipus imatge afegit al llistat de documents en un carrusel d'imatges amb la possibilitat de veure-les en gran".

Una vegada implementat l'objectiu, durant el període de prova de l'increment, es va reportar un baix rendiment i un alt temps de resposta relacionat amb la quantitat d'imatges que un element tenia adjunt a l'hora de visualitzar la seva informació.

Aquesta baixada de rendiment i augment en el temps de resposta del servidor quan un element tenia adjunts un gran nombre d'imatges venia degut a causa de que realitzàvem una descarrega total de les imatges adjuntes del element en qüestió per a la seva previsualització. Això ens va dur a replantejar el mètode de previsualització de les imatges.

Finalment es va prendre la decisió de que la solució mes practica i eficient era que en el mateix moment de guardar les imatges al servidor s'hauria de crear una versió comprimida i reduïda de cada una d'elles la qual seria la que carregariem per a la previsualització i que si es volia veure en una major qualitat es faria una descarrega individual. Aquesta decisió va comportar una millora de rendiment considerable. A continuació es mostrarà un exemple de millora amb només 3 imatges.

Tamany de les imatges guardades en el servidor(3081kB)

Nombre	Fecha de modifica...	Tipo	Tamaño
2968	31/05/2021 21:29	Archivo	524 KB
2969	31/05/2021 21:29	Archivo	1.336 KB
2967	31/05/2021 21:29	Archivo	1.221 KB

Figura 6: Imatges al servidor

Tamany total de la petició al servidor que retorna tota la informació de l'element amb la previsualització de les 3 imatges(74kB):

Name	Status	Type	Initiator	Size	Time
4049/	200	xhr	jquery.min.js4	73.6 kB	
4049/	200	preflight	Preflight	0 B	

Figura 7: Petició de l'informació de l'element

Thumbnails a la BD:

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT * FROM [Sphere].[dbo].[SparePartImageThumbnails] where FileId in (2969, 2968, 2967)
    
```

Id	SparePartId	Content	Field	Position
3	4049	IVBORw0KGgoAAAANSUkEulgAAAEcAAABkCAYAAADUIC2HAAAA...	2967	1
4	4049	IVBORw0KGgoAAAANSUkEulgAAAEIAAABkCAYAAAYoebDAAAA...	2968	2
5	4049	IVBORw0KGgoAAAANSUkEulgAAAD8AAABkCAYAAAXFuJrAAAAA...	2969	3

Figura 8: Thumbnails reduïts en base64

Aquesta decisió va comportar un baixada en el cost de transferència d'un 97.6%.

## 11 CONCLUSIONS

L'objectiu principal d'aquest projecte era crear una nova funcionalitat d'una aplicació mentre aquesta s'anava desenvolupant en paral·lel. Durant el principi del desenvolupament d'aquest projecte es va veure reflexada la meua falta de comunicació amb els demes integrants de l'equip de desenvolupament amb modificacions en el codi que inutilitzaven els meus progressos provocant pèrdues innecessàries de temps. Això ens va fer veure la importància que tenia el que tots els integrants de l'equip de desenvolupament tinguessin una idea general del que els demes integrants estaven desenvolupant a l'hora de prendre decisions sobre el disseny de l'aplicació principal. També, per evitar aquest tipus de problemàtiques, s'ha vist la importància de crear tests, per tal d'evitar futures modificacions que involuntàriament afectin a funcionalitats ja finalitzades i que creient que ja funcionaven correctament, no ens donem compte fins que se li instal·la al client i ho reporta.

D'altra banda també s'ha vist la importància que te no només crear un disseny funcional, sinó eficient. Com hem pogut veure a l'hora de dissenyar la funcionalitat de previsualitzacions de les imatges adjuntes.

Per finalitzar, he observat com tenir un equip amb tanta experiència al costat que pugui aportar diferents perspectives a l'hora d'abordar nous projectes, impulsa i facilita l'aprenentatge de noves tecnologies.

## 12 AGRAÏMENTS

Per acabar, vull agrair tota l'ajuda i consideració que he rebut per part del meu tutor de treball de fi de grau, Marc Talló, que tot i les meves tardances en les entregues dels informes ha confiat en mi en tot moment. També vull donar les gràcies als meus companys de pis que m'han estat recolzant en tot moment i quan em veien apurat i amoïnats realitzaven per mi les tasques quotidianes com pot ser el cuinar per tal de donar-me mes temps per a la realització del projecte. També vull agrair als meus pares que tot i no poder passar 2 setmanes sense veurem, durant el tram final d'aquest projecte m'han donat l'espai i el temps que necessitava per poder finalitzar a temps. I finalment vull agrair al meu tutor en l'empresa on estic treballant a jornada completa, en Ramon Andrés, que m'ha estat tutoritzant, m'ha guiat durant tot el procés i m'ha aportat tots els coneixements i habilitats que he hagut d'aprendre per poder desenvolupar totes les funcionalitats d'aquest projecte.

## 13 BIBLIOGRAFIA

- [1] GMAO  
[https://es.wikipedia.org/wiki/Gesti%C3%B3n\\_de\\_mantenimiento\\_asistido\\_por\\_computadora](https://es.wikipedia.org/wiki/Gesti%C3%B3n_de_mantenimiento_asistido_por_computadora)
- [2] OurSmart  
<https://out-smart.com/es>
- [3] Modelo incremental  
<https://www.javatpoint.com/software-engineering-incremental-model>
- [4] Git  
<https://es.wikipedia.org/wiki/Git>
- [5] Git-Flow  
<https://aprendegit.com/que-es-git-flow/>
- [6] IIS  
<https://docs.microsoft.com/es-es/iis/get-started/whats-new-in-iis-10-version-1709/new-features-introduced-in-iis-10-1709>
- [7] API REST  
[https://es.wikipedia.org/wiki/Transferencia\\_de\\_Estado\\_Representacional](https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional)
- [8] Karma  
<https://karma-runner.github.io/latest/index.html>
- [9] Fancytree  
<https://github.com/mar10/fancytree>
- [10] Datatables  
<https://datatables.net/>