
This is the **published version** of the bachelor thesis:

Sabater Bibiloni, Alejandro; Bolta Torrell, Helena, dir. Zenital file validation and compilation. 2024. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/290085>

under the terms of the  license

zenital file validation and compilation

Alejandro Sabater Bibiloni

Resum—En l'actual context, el món està experimentant un creixent impuls en la generació de dades, amb una enorme quantitat d'informació generada i compartida diàriament per a la seva utilització i manipulació. La digitalització i l'automatització han convertit la gestió i el processament de les dades en un aspecte crític. La validació de fitxers i del seu contingut abans de transferir-los a qualsevol destinació ja sigui en un entorn intern o amb tercers esdevé vital. D'aquí sorgeix zenital file validation i compilation, en endavant, Excellenses, la qual parteix d'un enfocament diferencial per resoldre aquesta necessitat, garantint que les dades a analitzar siguin fiables. Aquest objectiu es aconseguit mitjançant un procés de validació i transformació dels arxius desitjats. A més no només garanteix fiabilitat si no que prevé la propagació d'errors per les tipologies, estalviant temps i recursos.

Paraules clau—Excel, validacions, transformacions, consolidacions, dades, Python, Vue, JavaScript, API, Azure, Auth0.

Abstract—Versió en anglès del resum. In the current context, the world is experiencing a growing surge in data generation, with an immense amount of information being created and shared daily for its use and manipulation. Digitalization and automation have turned data management and processing into a critical aspect. The validation of files and their content before transferring them to any destination, whether it be an internal environment or with third parties, becomes vital. Hence arises the zenital file validation and compilation, henceforth known as Excellenses, which starts from a differential approach to solve this need, ensuring that the data to be analyzed is reliable. This objective is achieved through a process of validation and transformation of the desired files. Moreover, it not only guarantees reliability but also prevents the spread of errors across types, saving time and resources.

Index Terms—Excel, validations, transformations, consolidations, data, Python, vue, JavaScript, API, Azure, Auth0.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

ACTUALMENT el món està experimentant un creixent impuls en la generació de dades, amb una gran quantitat d'informació generada i compartida diàriament per a la seva utilització, manipulació i anàlisi. És per aquest motiu, que l'automatització ha convertit la gestió i el processament de les dades en un aspecte crític no menys important. En aquest sentit, la validació i la transformació de fitxers i del seu contingut abans de transferir-los a qualsevol destinació, ja sigui en un entorn intern o amb tercers esdevé vital. Les dades incorrectes, com podria ser un arxiu Excel amb dades dins una mateixa columna de diferents tipus quan haurien de ser totes del mateix tipus, o columnes incompletes, les quals tenen valors nuls, poden tenir un impacte significatiu en la presa de decisions; la qualitat de les dades a oferir i l'eficiència operativa.

Per reduir tots aquests aspectes negatius i aconseguir l'impacte positiu, sorgeix d'aquí zenital file validation and compilation, en endavant, Excellenses. La plataforma, la qual parteix d'un enfocament diferencial per resoldre aquesta necessitat, garantint que les dades a analitzar

siguin fiables i precises des de l'inici, independentment del tipus d'informació. Això no només prevé la propagació d'errors per les tipologies, sinó que també estalvia temps i recursos en minimitzar les correccions manuals i les revisions.

Excellenses es basa en una aplicació web senzilla en la qual l'usuari s'ha d'autenticar, per en primer lloc triar el fitxer Excel que vol validar i transformar. Seguidament l'usuari pot triar sobre les columnes quines validacions realitzar i un cop aquesta ha estat exitosa, pot triar quines columnes amb dades desitja transformar per aconseguir el fitxer final. Finalment, l'usuari pot triar a qui desitja enviar un missatge utilitzant la plataforma, decidir si vol descarregar el fitxer validat i transformat, i finalment decidir si guardar les dades dins una base de dades del sistema corporatiu.

- E-mail de contacte: alexsb98@gmail.com
- Menció realitzada: Enginyeria del Software
- Treball tutoritzat per: Helena Boltà (Ciències de la Computació)
- Curs 2023/24

2 OJECTIUS

El projecte té com a objectiu principal desenvolupar una interfície d'usuari que permeti carregar els arxius a validar i visualitzar-ne els errors en cas d'existents. Això es complementa amb la implementació d'un sistema d'inici de sessió que assegura la seguretat i la personalització de l'experiència d'usuari.

Seguint amb els següents objectius, es vol establir un conjunt de processos de validació automàtica per garantir la qualitat dels arxius, incloent la comprovació de requisits mínims que els arxius han de complir com per exemple, files i columnes a l'inici d'un arxiu completament buides.

Una part crucial del projecte és la creació d'un sistema que notifiqui als usuaris de forma intuïtiva sobre qualsevol error existent, proporcionant detalls específics sobre la columna i la fila en la qual hi ha l'error. Això permet als usuaris modificar ràpidament els errors de l'arxiu per a tornar a començar el procés de validació.

Un altre objectiu clau és que el sistema permeti transformar les columnes de les dades carregades, és a dir, ja validades a un altre format de manera coherent i estructurada, assegurant-se que les dades transformades mantinguin la seva integritat i utilitat.

Per facilitar l'accés i l'ús dels arxius, el sistema ha d'incloure un mètode per descarregar els arxius, ja validats i transformats, preparats per al seu ús posterior.

Per a l'emmagatzematge de fitxers, es desenvoluparà un sistema que no només guardi els arxius validats, sinó que també permeti el seu emmagatzematge automàtic, eliminant la necessitat d'intervenció manual i agilitzant així tot el procés.

Finalment, el projecte permet personalitzar les validacions per adaptar-se a les necessitats específiques de cada cas d'ús, oferint així una solució versàtil i adaptable a diferents contextos i requisits.

3 ESTAT DE L'ART

Dins del context actual, on la generació i utilització de dades es produeix a una escala sense precedents, la necessitat d'eines automatitzades per a la validació i transformació de dades és més crítica que mai. Excellenses emergeix com una solució innovadora en aquest panorama, oferint una plataforma web que simplifica i assegura la integritat de les dades des de la seva entrada inicial fins a la seva destinació final.

El software i les eines de darrera generació juguen un paper crucial en aquesta àrea, i s'ha fet ús d'una combinació de tecnologies de avantguarda per construir l'aplicació.

Front End: Utilitzant el llenguatge de programació JavaScript juntament amb el framework Vue.js, he creat una interfície d'usuari intuïtiva i interactiva. Vue.js és molt útil per la seva facilitat d'ús i per simplificar el desenvolupament de pàgines web interactives, permetent una ràpida integració.

Back End: Per al processament de dades, s'han seleccionat llibreries robustes com Pandas i Numpy, les quals són estàndards en el món de la ciència de dades per la seva capacitat de manipulació i anàlisi eficient de grans volums de dades. Aquestes llibreries ofereixen funcions avançades per a la validació i transformació de dades, essencials per a les operacions que realitzam amb Excellenses.

Connexió i interacció amb la base de dades: S'ha utilitzat SQLAlchemy, una llibreria ORM de Python que permet d'una manera més senzilla manipular les bases de dades. Aquesta llibreria facilita la connexió i la gestió de les transaccions de la base de dades d'una manera segura i eficient.

Pel desenvolupament dels EndPoints: Amb l'ajuda de la llibreria Flask, s'ha establert una arquitectura d'API[1] robusta i escalable (objectiu que teníem també pel desenvolupament). Flask és coneguda per la seva lleugeresa i flexibilitat, permetent la creació d'APIs de tipus RESTful (utilitzant arquitectura REST i mètodes HTTP) amb un esforç relativament baix i una gran llibertat en la definició d'EndPoints.

Gestió d'arxius amb Azure[2]: Per al tractament i emmagatzematge dels arxius, s'ha fet ús de les eines que ofereix Azure, en particular la llibreria de Python Utils Azure. Això permet una integració amb els serveis en el núvol de Microsoft, facilitant un emmagatzematge segur i un accés eficient als fitxers.

Aquestes tecnologies conformen la base sobre la qual Excellenses opera, proporcionant una solució completa que no només valida i transforma les dades, sinó que també permet la seva gestió i distribució d'una manera centralitzada i automàtica.

La selecció d'eines mencionada respon a les tendències actuals en el desenvolupament de software, on la capacitat de l'aplicació per escalar i adaptar-se a diferents necessitats és fonamental. A més, el fet de tenir una API independent del Front End proporciona una flexibilitat considerable en termes de futur desenvolupament i la possibilitat de personalitzar l'experiència d'usuari segons les necessitats específiques.

La combinació d'aquestes tecnologies posa de manifest la capacitat de Excellenses per ser una eina potent en el context de l'empresa zenital, optimitzant la gestió de dades i la presa de decisions basades en informació fiable i precisa.

4 REQUISITS

Pel desenvolupament del projecte s'han tingut en compte els següents requisits funcionals i no funcionals.

4.1 REQUISITS FUNCIONALS

Sobre els requisits funcionals que es van plantejar, podem comentar el següent:

El sistema dissenyat proporciona una sèrie de funcionalitats integrades per a una gestió eficient de dades. Començant amb una funcionalitat d'inici de sessió a través d'Auth0[3], ofereix una plataforma segura pels usuaris. Un cop autenticats, els usuaris poden carregar arxius que requereixen validació. El sistema implementa processos de validació automàtica, sempre assegurant que les dades siguin precises i fiables.

A més, el sistema ofereix la flexibilitat de personalitzar les validacions d'acord amb requisits específics, augmentant la seva aplicabilitat en diversos contextos i/o tipus de negoci.

A més, Excellenses permet als usuaris visualitzar els resultats de les validacions de manera clara i accessible. En cas de detectar-se errors, el sistema notifica als usuaris, proporcionant-los la informació necessària per a que puguin prendre les mesures necessàries per modificar-les. Més enllà de la validació, el sistema té la capacitat de transformar les dades seleccionades a un altre tipus de manera coherent, adaptant-se a les diverses necessitats i formats requerits.

Un cop els arxius han estat validats i confirmats com a aptes per al seu ús, el sistema permet la seva descàrrega. Això es complementa amb un espai dedicat per a l'emmagatzematge dels arxius validats i transformats (emmagatzematge dins Blob Storage[4]), contribuint a una organització eficaç i segura de les dades ja que es troben a l'entorn Cloud d'Azure.

En termes d'eficiència, el sistema està dissenyat per assolir una precisió de validació de dades superior al 95%. Això, juntament amb l'automatització del procés, es tradueix en una reducció significativa del temps necessari per validar un arxiu, arribant a una millora del 98% en comparació amb els mètodes manuals.

4.2 REQUISITS NO FUNCIONALS

Sobre els requisits no funcionals que es van plantejar, podem comentar el següent:

El sistema està dissenyat per respondre de manera ràpida i eficient, optimitzant el rendiment i assegurant que l'experiència de l'usuari sigui lleugera. Per a la interfície d'usuari, s'ha desenvolupat un Front End atractiu (colors opacs), intuïtiu i fàcil d'utilitzar que simplifica la interacció amb les funcionalitats de l'aplicació.

La seguretat de les dades és una prioritat màxima; per això, s'han implementat mesures rigoroses per prevenir accessos no autoritzats, garantint així la protecció de la informació emmagatzemada i gestionada.

Seguint els requisits el Back End ha estat desenvolupat amb codi Python, aprofitant la seva potència i flexibilitat per manejar el processament de dades. El Front End, per la seva banda, ha estat creat amb HTML, CSS i JavaScript juntament amb el Framework Vue.js[5], una combinació clàssica que facilita la creació d'una experiència d'usuari rica i dinàmica. A més, un dels altres requisits era que el sistema disposés d'una API robusta que permet una comunicació fluida i segura entre el Front End i el Back End, assegurant que les dades i les sol·licituds de l'usuari siguin processades de manera efectiva i sense interrupcions.

5 METODOLOGIA

Per assolir l'objectiu final del projecte, es va decidir adoptar la metodologia Agile Lean Startup[6], estratègia la qual fusiona les millors característiques Agile i Lean Startup.

Gràcies a la seva naturalesa iterativa, on els nostres cicles d'iteracions han estat de 3 setmanes, basada en la retroalimentació ràpida dels stakeholders, ens ha permet ajustar i refinar l'aplicació Excellenses en funció de les necessitats i canvis que han anat sorgint per tal d'aconseguir el mínim producte viable (MPV). Gràcies a la simplicitat i flexibilitat d'aquesta metodologia l'ha fet especialment adequada per afrontar els reptes que han anat sorgint.

Com a eina per usar aquesta metodologia s'ha utilitzat el Trello[7], amb el qual de forma molt intuïtiva he pogut anar creant diferents taulells amb les diferents tasques per dur un control sobre el que quedava en BackLog, el que estava en desenvolupament i el que ja estava pràcticament desenvolupat.

Per dur un control de versions del codi desenvolupat, he utilitzat l'eina GitHub[8].

Amb GitHub a més de portar un control sobre les diferents versions tant del Front End com del Back End he pogut realitzar de forma continua el desplegament continu. GitHub està ben implementat amb Azure, cosa que m'ha permès de forma fàcil (un cop configurat) anar desplegant als servidors d'Azure tot el codi que ja estava preparat per l'entorn productiu, de tal forma que solament realitzant un commit podia veure en qüestió de segons si el desplegament era satisfactori així com els resultats.

6 DESENVOLUPAMENT

Per tal d'arribar al MPV final es varen crear petites tasques de desenvolupament tenint en compte els requisits funcionals i no funcionals de l'aplicació i a la vegada, l'objectiu final. Aquestes tasques es van començar a realitzar amb la investigació sobre els sistemes existents de validació de dades l'11 de setembre. Per una part es varen crear les tasques específiques pel Front End, les quals incloïen les tasques de la 2 a la 6, la tasca 9 i les tasques de la 11 a la 14. Es van

crear tasques per a realitzar el Back End (tasques 2-10, 12-14 i la tasca 23). Seguidament es van crear les tasques destinades a testejar el producte, tasques 15, 16 i 17 i finalment varies tasques destinades a realitzar els informes i documentació del projecte, tasques 19-22.

Aquestes tasques van quedar de la següent forma:

ID	Tasca
1	Investigació sobre eines i validacions de dades
2	Desenvolupament de regles lògiques per la validació estàndard.
3	Implementar una funcionalitat de càrrega de fitxers
4	Implementar una funcionalitat per la visió dels resultats
5	Implementar una funcionalitat per la visió dels errors.
6	Implementar una funcionalitat per la descarrega dels resultats validats correctament
7	Desenvolupar i configurar l'emmagatzematge de fitxers carregats en una ubicació designada.
9	Desenvolupar una funcionalitat d'inici de sessió mitjançant Auth0.
10	Desenvolupament del sistema per validar les regles estàndard.
11	Desenvolupament de la interfàç (Front End)
12	Desenvolupament del sistema per validar les regles personalitzades.
13	Desenvolupament del sistema encarregat transformar les dades a la tipologia triada.
14	Desenvolupament sistema de notificació a usuaris proporcionant informació sobre les validacions.
15	Realització de proves per identificar qualsevol error o bug. (Testing)
16	Realitzar proves d'experiència d'usuari per tal de trobar millores en la UX i UI.
17	Resolució d'errors o de millores.
18	Informe Inicial
19	Informe de progrés (I)
20	Informe de progrés (II)
21	Informe final
22	Presentació
23	Desenvolupament API

Taula (1)

Un cop vaig tenir el llistat de tasques, vaig elaborar el cronograma, podem veure les tasques que s'han realitzat i en quins períodes temporals.

ID	SETMANA																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1																			
2																			
10																			
12																			
13																			
19																			
23																			
3																			
4																			
5																			
6																			
7																			
9																			
11																			
15																			
16																			
17																			
20																			
21																			
22																			

Taula (2)

El treball s'ha anat realitzant per iteracions o cicles de 3 setmanes on s'ha anat avaluant l'aplicació desenvolupada i testejant fins el moment per tal de rebre el feedback i les propostes dels canvis que fossin necessaris.

La relació entre les iteracions, setmanes i les hores dedicades a cada setmana es pot trobar a la següent taula:

Iteració	Tasca (ID)	Setmanes	Període	Hores totals
1	2, 10, 12 i 13.	5, 6 i 7	9 Oct. – 29 Oct.	64
2	13, 19 i 23.	8, 9 i 10	30 Oct. – 19 Nov.	53
3	3, 4, 5, 6, 7 i 11.	11, 12, 13	20 Nov. – 10 Dec.	51
4	11, 9, 15, 16, 17 i 20.	14, 15, 16	11 Dec. – 31 Dec.	60
5	11, 17, 21 i 22	17, 18 i 19	1 Gen. – 21 Gen.	(30)* 45

Taula (3)

Les hores del darrer període han augmentat en 15 hores (anteriorment 30 hores)*, degut a que es va fer un mal càlcul sobre les hores dedicades a la tasca 11, desenvolupament del Front End. En un principi creia que el desenvolupament del Front End seria molt més fàcil, però la dificultat sobre aquest ha sigut el desencadenant de que augmentessin les hores. Com ha estat el darrer període i teníem temps de sobra, encara que s’endarrerís uns dies la data final del projecte, no va causar cap problema de força major.

6.1 PROBLEMES DESENVOLUPAMENT

Degut a la complexitat del projecte, vaig elaborar a l’inici del projecte una secció de riscos i diferents plans de contingències per tal que si en algun moment algun risc s’arribava a materialitzar, ja teníem present un pla per tal que l’impacte fos el menys greu possible; i poder actuar de la forma més ràpida i eficient possible modificant al mínim la cronologia temporal de les tasques precedents.

Risc	Pla de contingències
Resultats no satisfactoris	Si a la investigació inicial s'utilitza més temps del previst, es reavaluarà el cronograma del projecte interferint el mínim amb les següents fases i activitats.
Canvis en els requisits	Si es produeixen canvis en els requisits, es realitzarà una avaluació de l'impacte. S'ajustarà el cronograma en cas necessari.
Problemes d'integració	Si s'enfronta a problemes d'integració durant el desenvolupament, es buscaran solucions alternatives si s'escau.
Erroros crítics no detectats	Si es descobreixen errors crítics durant les proves, s'abordaran de manera urgent per tal d'interferir el mínim amb el cronograma. Seguidament es realitzaran tests i proves addicionals per garantir l'efectivitat i l'eficàcia del PMV d'Excellenses.

Taula (4)

Durant el desenvolupament de l’aplicació, varen sorgir dos problemes principals.

En primer lloc, no vaig tenir en compte que s’hauria de necessitar una API per la comunicació entre el Front End i el Back End. Degut a això vam haver de crear la tasca. Respecte a aquest punt no va haver cap problema més ja que ens trobàvem pràcticament a la fase inicial de desenvolupament, i simplement es va crear la tasca i es van augmentar les hores a dedicar en el període.

El següent problema que va aparèixer va ser un canvi

en un dels requisits, en un principi es va definir que l’autenticació havia de realitzar-se mitjançant Azure AD, actual Entra ID. Posteriorment es va modificar el requisit degut a que utilitzar aquesta primera opció era obligar a tothom qui volgués utilitzar l’aplicació a utilitzar un compte Microsoft. Degut a aquesta problemàtica, es va decidir optar per implementar l’autenticació mitjançant Auth0. Utilitzant aquest servei podem permetre iniciar sessió mitjançant Google i Microsoft, entre d’altres.

Aquest canvi en el requisit no va suposar cap problema malgrat que sí que vaig haver de fer investigació profunda sobre com integrar Auth0 amb el que ja havia desenvolupat.

Finalment, cal dir que més que un problema va ser un aprenentatge, vaig subestimar les capacitats del servidor d’Azure. En el procés de desenvolupament, va arribar un moment en el qual apareixien errors estranys al Back End, bàsicament teníem contractat un hardware basic en el qual solament tenia de memòria 1,75 GB. Quan em vaig adonar d’aquest problema vaig escalar verticalment el servidor per tal de poder aprovisionar de més memòria el servidor.

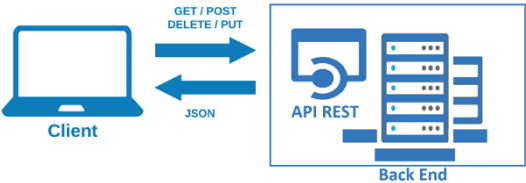
6.2 FASES DE PROVA D’UNITAT I D’ACCEPTACIÓ

Com ja s’ha comentat anteriorment, cada 3 setmanes es realitzava unes proves d’acceptació amb l’equip per tal de rebre el feedback necessari. Entre els notables canvis que es van realitzar va ser el canvi de color de blau (color corporatiu) al color verd més propi i proper als arxius Excels.

Sobre les proves d’unitat, s’ha anat realitzant testing durant tot el desenvolupament sobre el que s’anava desenvolupant.

7 ARQUITECTURA I DISSENY

L’aplicació Excellenses està composta per un Front End allotjat a un servidor i a la vegada un Back End allotjat a un servidor diferent. Per altra banda tenim la base de dades SQL Server per emmagatzemar les dades de forma estructurada i dos contenidors de tipus “blob storage” en el qual s’emmagatzemen els arxius validats i transformats. Tots els serveis estan allotjats al núvol de Microsoft Azure.



Imatge (1)

7.1 FRONT END

El Front End s’ha desenvolupat en la seva totalitat en “Vue.js”, degut a ser bastant simple i molt flexible. Aquest m’ha permès crear una interfície d’usuari interactiva i dinàmica de forma fàcil malgrat la curva d’aprenentatge que

presenta. Cal dir que al principi era una mica complicat, però gràcies a que el front end d'Excellenses no es gens complex no m'ha resultat cap esforç extra aprendre.

El Front End està allotjat com a Aplicació Web Estàtica[9] a un servidor d'Azure.

El Front End segueix el tipus d'estructura C-BA, Component-Based Architecture, on cada part de la interfície està modulada o encapsulada en els seus propis components. D'aquesta forma poden ser reutilitzats a altres projectes sense grans complicacions.

A la vegada segueix l'estructura SFCs, Single File Components, on bàsicament cada fitxer conté un template html, l'script amb les funcions javascript les quals donen sentit a les funcionalitat com fer cridades a la API del back end i l'estil CSS.

Entre els diferents arxius hi trobem:

App.vue. Fitxer en el qual es crea una instància Vue integrada amb el servei d'autenticació d'Auth0 (Fitxer main.sj)

Entre els components que donen sentit a l'aplicació hi trobem els següents:

FileUpload.vue. Component en el qual es carrega el fitxer desitjat per l'usuari.

ValidationsComponent. Component encarregat de mostrar les diferents columnes de l'arxiu carregat i els diferents tipus de validació que es poden realitzar. S'encarrega també de mostrar els diferents errors (en cas que n'hi hagi) de la validació.

TransformationsComponent. Component encarregat de mostrar les diferents columnes de l'arxiu carregat i ja validat juntament amb les seves possibles transformacions que es poden realitzar.

CommunicationsSection. Component en el qual l'usuari pot escriure un missatge al manager triat. A la vegada pot descarregar el fitxer ja validat i transformat i per últim pot decidir si enviar el fitxer a la base de dades SQL.

MenuItemComponent. Component encarregat de mostrar les icones i el text de cada un dels diferents nivells o menús. S'encarrega també de modificar l'estat.

SidebarComponent.vue. Component encarregat de definir la barra lateral i d'incorporar el MenuItemComponent.

MainContentComponent. Component encarregat de gestionar el contingut principal de l'aplicació (diferents components).

7.2 BACK END

El Back End s'ha desenvolupat completament utilitzant Python. Gràcies a aquest codi, excel·lent per poder manipular dataframes de dades amb Pandas[10] (framework) he desenvolupat els mòduls de les validacions i transformacions.

Per la banda de l'API, he utilitzat el framework Flask[11] el qual es molt popular degut a la seva versatilitat i integració amb Python.

Finalment per a realitzar les diferents operacions sobre la base de dades he utilitzat SQL Alchemy Python. Vaig triar aquest degut a l'ORM que presenta ja que facilita la manipulació de dades i la interacció amb la base de dades de manera eficient i segura.

Finalment, cal dir que el Back End està allotjat com un App Service[12] a un servidor d'Azure.

7.3. BLOB STORAGE

Blob storage es el servei d'Azure en el qual s'emmagatzemen tots el arxius que passen pel servei back end.

En el primer contenidor, emmagatzemem tots els fitxers validats mentre que en el segons hi guardem els fitxers transformats. D'aquesta manera tenim tots els arxius completament organitzats.

<input type="checkbox"/>	excellenses	SQL Server
<input type="checkbox"/>	Excellenses (excellenses/Excellenses)	Base de datos SQL
<input type="checkbox"/>	excellenses	Cuenta de almacenamiento
<input type="checkbox"/>	excellenses	App Service
<input type="checkbox"/>	excellensesfrontend	Aplicación web estática

Imatge (2)

7.4. ENDPOINTS

- /upload. EndPoint el qual rep el fitxer des del Front End, s'encarrega de retornar al Front End les columnes d'aquest fitxer juntament amb les validacions aplicables.
- /startValidation. EndPoint el qual rep la informació en format JSON sobre les validacions a realitzar sobre les columnes. Aquesta informació es la desitjada per l'usuari final. Si la validació resulta favorable, és a dir, no conté cap mena d'error, el fitxer es pujat al blob storage pertinent i l'usuari pot continuar amb el procés de transformacions. Si les dades del fitxer contenen errors, aquests son retornats en llenguatge natural a través d'un

JSON al Front End per tal de mostrar a l'usuari quins son els errors que cal modificar, en aquest punt no es realitza cap pujada d'arxiu al blob storage ja que no té cap sentit guardar informació mal tipada i errònia.

- /startTransformation. EndPoint el qual rep la informació en format JSON sobre les transformacions a realitzar sobre les columnes. Aquesta informació es la desitjada per l'usuari final. L'EndPoint s'encarrega de realitzar aquestes transformacions, en cas de que sigui possible. Una vegada les dades han estat transformades també es guarda una copia del dataframe al blob storage, aquesta es la versió de les dades que més sentit té ja que han estat tant validades com transformades a la tipologia que ha triat l'usuari final.
- /startConsolidation. EndPoint el qual rep la informació en format JSON des del front end. Aquesta informació inclou, l'usuari al que s'ha d'enviar el correu electrònic, el contingut del correu electrònic i per últim la discissió de voler afegir les dades a la base de dades corporativa (com a taula de base de dades racional). En el cas de l'enviament del correu electrònic, vaig desenvolupar una petita funcionalitat la qual rep el tipus de missatge en format html i aquest realitza l'enviament del missatge al destinatari triat.
- /download_excel. EndPoint encarregat de rebre la request per part de l'usuari al prémer el botó de descarregar Excel, aquest retorna el fitxer transformat en un format Excel.

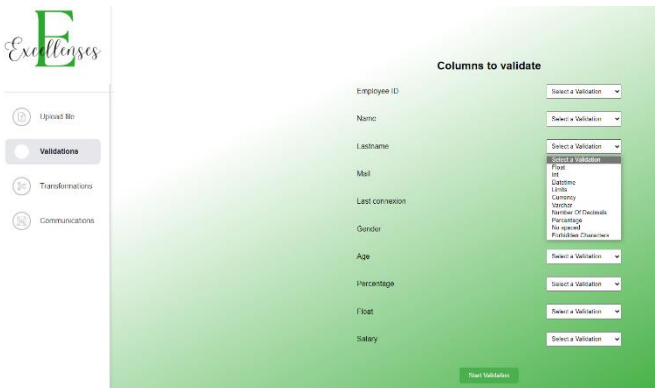
sistema que ofereix la solució per a la validació i transformació de fitxers de dades. Aquest sistema s'ha dissenyat per millorar significativament l'eficiència en el tractament de les dades dins de l'organització, reduint els errors i optimitzant el temps necessari per a les tasques de gestió de dades.



Imatge (3)

Les característiques més destacables d'Excellenses inclouen:

- Interfície d'usuari intuïtiva: El Front End proporciona una experiència d'usuari bona, amb una navegació senzilla la qual segueix un sol i únic flux i una estètica agradable que facilita l'adopció per part dels usuaris finals.



Imatge (4)

7.5 LOGIN

Un dels objectius era tenir un login completament funcional amb el qual poder tenir constància de qui era la persona que realitzava l'execució de validació, transformació i posteriorment guardava les dades a la base de dades.

Per a desenvolupar aquest sistema s'ha fet ús del servei Auth0. A través del seu SDK, la seva integració amb l'aplicació Excellenses ha estat bastant senzilla.

Ara per ara, es permet el login utilitzant compte Google i Microsoft.

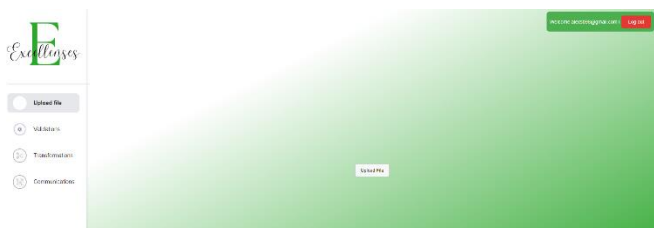
Un cop el login s'ha realitzat i ha estat satisfactori, s'accedeix a l'aplicació web on a un pop-up situat a la part superior dreta, podem veure amb quin compte esteim logejats i a la vegada tancar aquesta sessió.

8 RESULTATS

Els resultats assolits en el desenvolupament del Treball de Fi de Grau han culminat amb l'MPV d'Excellenses, un

- Validació i transformació: A través dels seus processos de validació i transformació, Excellenses garanteix l'exactitud de les dades, minimitzant a 0 la necessitat de revisió manual.

- Seguretat de Dades: El sistema aplica protocols de seguretat per a l'emmagatzematge i el tractament de les dades, incloent protecció contra accessos no autoritzats.



Imatge (5)

Tot i que les proves d'acceptació s'han realitzat per assegurar que el sistema compleix amb els requisits i expectatives, no s'ha realitzat documentació sobre aquesta ja que era més orientada a realitzar petits canvis. No obstant això, els indicadors basats en l'ús per part dels usuaris finals (treballadors de zenital) suggereixen una millora substancial en el temps requerit per realitzar tasques de validació i transformació de dades. El feedback obtingut apunta a una recepció positiva del sistema per part de l'equip.

Aquestes millores són especialment notables en comparació amb els processos manuals anteriors, que sovint eren propensos a errors i requereixen més temps per a la seva realització (es podia tardar d'entre 2 a 3 hores en revisar cada un dels fitxers amb els que treballem actualment). Amb Excellenses, s'ha aconseguit reduir aquests temps de manera significativa.

En conclusió, els resultats del TFG reflecteixen l'èxit en el desenvolupament de l'aplicació.

9 CONCLUSIONS I TREBALL FUTUR

Com a conclusió principal del treball realitzat durant aquests 4 mesos, podem dir que hem obtingut l'MPV (Mínim producte viable) que es plantejava per part de l'empresa.

S'ha aconseguit desenvolupar una aplicació funcional, amb la qual qualsevol usuari pot iniciar sessió mitjançant el seu compte de Google o Microsoft en cas de tenir compte.

S'ha aconseguit desenvolupar la funcionalitat de validació del fitxer desitjat mitjançant qualsevol de les regles disponibles. En cas que hi hagi errors de validació, aquests es mostren de forma senzilla indicant quins són els camps amb errors.

S'ha aconseguit desenvolupar la funcionalitat de transformació dels fitxers a la tipologia de dades necessària.

S'ha desenvolupat un mètode de consolidació/comunicació en el qual es pot enviar un missatge a qualsevol manager disponible a la base de dades comunicant que el fitxer ja està disponible per la seva descarrega.

S'ha aconseguit desenvolupar finalment la possibilitat de descarregar dels fitxers mitjançant un botó de descarrega.

Un dels avantatges d'haver desenvolupat la API de forma independent del Front End, és a dir, en servidors diferents, es que podem realitzar peticions GET/POST i obtenir els mateixos resultats. Això ens permet en un futur diverses coses, en primer lloc podem crear un Front End completament diferent segons les regles necessàries o de negoci existents. En segon lloc podem seguir amb el desenvolupament de la API i finalment utilitzar-la inclòs sense necessitat de tenir un Front End.

A més, una de les conclusions més satisfactòries es l'aprenentatge que he pogut aconseguir sobre el Cloud d'Azure.

Excellenses és el principi pràcticament d'una aplicació que pot anar creixent amb el temps segons les necessitats finals de l'empresa zenital.

Actualment ja s'està utilitzant de forma interna i s'espera poder introduir millores respecte el disseny del front end i més funcionalitats respecte el back end.

Ara per ara, el treball realitzat i aconseguit és el que es va proposar com treball de fi de grau i és el que finalment s'ha aconseguit.

AGRAÏMENTS

En primer lloc m'agradaria fer menció a tota la meua família, amics i parella; persones que han estat allà tant en els moments mes estressants com en els millors moments com és l'actual.

Ja son quasi dos anys que duc treballant a zenital i poder realitzar el TFG dins la mateixa empresa a la qual vaig començar a treballar ha sigut un orgull.

Vull fer especial menció a l'equip que ha fet possible aquest projecte: Rui Correia, Bruno Costa, l'Íngrid Rodríguez i Anxo Armada.

Gràcies Íngrid per tot el que m'has ensenyat i ser una referent.

Dels creadors de "Por ahora no puedo hacer nada más, ya que mi turno ha terminado" i de "he revisado la información y entiendo que deseas discutir XXXXX", endavant i sense freno que moltes coses bones estan per venir!

Gràcies!

BIBLIOGRAFIA

- [1] API [What is an Application Programming Interface \(API\)? | IBM](#) [Febrer, 2023]
- [2] Azure [Qué es Azure: Servicios en la nube de Microsoft | Microsoft Azure](#) [Febrer, 2023]
- [3] Auth0 [¿Qué es Auth0 y para qué sirve? - Platzi](#) [Febrer, 2023]
- [4] Blob Storage [Introduction to Blob \(object\) Storage - Azure Storage | Microsoft Learn](#) [Febrer, 2023]
- [5] Vue.js [Vue.js - The Progressive JavaScript Framework | Vue.js \(vuejs.org\)](#) [Febrer, 2023]
- [6] Agile Lean Startup [Cuál es la diferencia entre Agile, Lean Startup](#)

- [y Design Thinking \(deloitte.com\)](#) [Febrer, 2023]
- [7] Trello [Manage Your Team's Projects From Anywhere | Trello](#)
- [8] GitHub [GitHub](#) [Febrer, 2023]
- [9] Aplicació Web Estàtica [¿Qué es Azure Static Web Apps? | Microsoft Learn](#) [Febrer, 2023]
- [10] Pandas [pandas - Python Data Analysis Library \(pydata.org\)](#) [Febrer, 2023]
- [11] Flask [Welcome to Flask — Flask Documentation \(3.0.x\) \(palletsprojects.com\)](#) [Febrer, 2023]
- [12] App Service [Overview - Azure App Service | Microsoft Learn](#) [Febrer, 2023]