



Treball de Fi de Grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica

Universitat de Barcelona

SegMyBill: Desglossament de factures

Nil Viñals Nin

Director: Josep Vañó Chic

Realitzat a: Departament de
Matemàtiques i Informàtica

Barcelona, 13 de juny de 2022

Resum

El projecte, consisteix en l'anàlisi, el disseny, la implementació i la posada en marxa d'un lloc web que automatitza el desglossament de factures d'una manera eficient.

Per a la seva implementació, s'ha utilitzat el framework ASP NET.Core de C#, Flask de Python, una sèrie de llibreries de Python, els llenguatges de desenvolupament web HTML, CSS, el framework de base TypeScript que ens dóna Angular i per la persistència de dades Firebase Authentication i Firebase Firestore.

El lloc web permet que els usuaris puguin separar tota la informació més important d'una factura segons una plantilla, fent així que sigui una feina molt més àgil, donat que cada empresa té un tipus de factura diferent. La millor forma de solucionar-ho, és a partir de col·laboració, per això aquest el contingut de la web és totalment accessible per qualsevol usuari registrat.

Les dues funcions que aporten més valor a la web són, la possibilitat de penjar plantilles i la possibilitat d'utilitzar alguna de les plantilles guardades per extreure la informació d'unes factures en concret. Cal dir que la visualització de tota de la informació relacionada amb la gestió de les plantilles o dels usuaris es restringeix en funció dels permisos d'accés o rols que tingui donats el perfil de l'usuari.

Les funcionalitats implementades a l'espai web proporcionen la possibilitat de crear i gestionar un usuari amb un sistema de tokens per assegurar-se l'autenticació d'aquest, crear i gestionar noves plantilles des de zero i obtenir les dades més rellevants de la factura segons la plantilla indicada. També permeten restringir l'accés de l'usuari segons el seu rol a veure certa informació o funcions tan en el backend com en el front.

Aplicant la lògica "Do it yourself", fent que la comunitat es nodreixi, el servei de la web Api de segmentació és totalment pública, però no tindrà l'avantatge de tenir les plantilles de la comunitat, ja que es publica pensant en integracions personals o en programes de codi obert.

El disseny de la part frontend del lloc web procura que sigui una experiència d'usuari accessible, còmode per tots tipus d'usuaris i adaptable a diferents dispositius, ja que, el contingut de l'espai web és ajustable al tamany de la pantalla del dispositiu.

Resumen

El proyecto consiste en el análisis, el diseño, la implementación y la puesta en marcha de un sitio web que automatiza el desglose de facturas de una forma eficiente.

Para su implementación, se ha utilizado el framework ASP NET.Core de C#, Flask de Python, una serie de librerías de Python, los lenguajes de desarrollo web HTML, CSS, el framework de base TypeScript que nos da Angular y para la persistencia de datos Firebase Authentication y Firebase Firestore.

El sitio web permite que los usuarios puedan separar toda la información más importante de una factura usando una plantilla, haciendo que sea una labor más ágil, dado que cada empresa tiene un tipo de factura completamente diferente al resto de empresas. La mejor manera de solucionarlo, es a partir de la colaboración, por eso el contenido de la web es totalmente accesible para cualquier usuario registrado.

Las dos funciones que aportan más valor a la web son, la posibilidad de colgar plantillas y la posibilidad de usar alguna de las plantillas guardadas para extraer la información de una factura de una empresa en concreto. Hace falta remarcar que la visualización de toda la información relacionada con la gestión de las plantillas o de los usuarios es restringida en función de los permisos de acceso o roles que tenga el usuario.

Las funcionalidades implementadas en el espacio web proporcionan la posibilidad de crear y gestionar un usuario con un sistema de tokens para asegurarse la autenticación de este, crear y gestionar nuevas plantillas desde cero y obtener los datos más relevantes de la factura según la plantilla indicada. También permiten restringir el acceso del usuario según su rol a ver cierta información o funciones tanto en el backend como en el frontend

Aplicando la lógica "Do it yourself", haciendo que la comunidad se nutra, el servicio que de la Web Api de segmentación es totalmente pública, pero no tendrá la ventaja de tener disponible las plantillas de la comunidad, ya que es pública pensando en integraciones personales o en programas de código abierto.

El diseño de la parte del frontend de la web procura que sea una experiencia de usuario accesible, cómoda para todos los usuarios y adaptable a diferentes dispositivos, puesto que, el contenido del espacio web es ajustable al tamaño de la pantalla del dispositivo.

Abstract

This project consists of the analysis, design, implementation and start-up of a website that automates the breakdown of invoices in an efficient way.

For its implementation, I have used the APS NET.Core C# Framework, Python Flask, a series of Python libraries, web development languages like HTML, CSS, the TypeScript base framework provided by Angular and for data persistence, I used Firebase Authentication and Firebase Firestore.

The website allows users to extract all the most important information from an invoice using a template, making it a more agile task, since each company has a completely different type of invoice than the rest of the companies. The best way to solve this is through collaboration, which is why the content of the website is fully accessible to any registered user.

The two functions that add the most value to the website are the capacity for uploading templates and the capacity to use one of the saved templates to extract the information from a specific company invoice. I should emphasize that the visualization of all the information related to the management of templates or users is restricted depending on the role that the user has.

The implemented functionalities on the web page provide the possibility to create and manage a user with a token system to ensure user authentication, to create and manage new templates from scratch and to obtain the most relevant data of the invoice according to the indicated template. It also allows us to restrict the access of the user according to his role in seeing certain information or functions both in the backend and in the frontend.

Applying the "Do it yourself" logic, making the community nurture itself by making the segmentation Web Api totally public, but people who use it will not have the advantages of having the community templates available. Making this Api open to the public is to promote the development of personal projects and open source programs to make this service carry even further.

The design of the frontend seeks to be an accessible user experience, comfortable for all users and adaptable to different devices, since the content of the web page is adjustable to the screen size of the device.

Índex

Resum	2
Resumen	3
Abstract	4
Índex	5
Índex de figures	8
1 Introducció	10
1.1 Motivacions i context del projecte	10
1.2 Descripció del projecte	11
1.3 Objectius	12
1.3.1 Objectius generals	12
1.3.2 Objectius específics	12
1.4 Anàlisi de riscos	13
1.5 Enfocament i mètode seguit	15
1.6 Planificació temporal	16
1.6.1 Distribució del temps	16
1.6.2 Fites i dates de lliurament	16
2 Anàlisi del sistema	18
2.1 Requeriments inicials	18
2.1.1 Funcionals	18
2.1.1.1 Usuaris	18
2.1.1.2 Aprenentatge	18
2.1.1.3 Gestió de plantilles	18
2.1.1.4 Gestió d'usuaris	18
2.1.1.5 Segmentació de factures	19
2.1.2 No funcionals	19
2.1.3 Tècnics	20
2.2 Casos d'ús	21
2.2.1 Diagrama de casos d'ús	21
2.2.2 Descripció dels casos d'ús textuais	22
3 Disseny	28
3.1 Diagrama del projecte	28
3.2 Arquitectura	29
3.2.1 Descripció de les capes:	30
3.2.2 Injecció de dependències	30
3.2.3 Mapping	31
3.4 Prototipat de plantilles	32
3.5 Prototipat gràfic	35
3.5.1 Pàgines públiques	35

3.5.1.1	Pàgina d'inici	35
3.5.1.2	Pàgina de tutorials	36
3.5.1.3	Pàgina de funcionament	37
3.5.1.4	Pàgina d'inici de sessió	37
3.5.1.5	Pàgina de crear usuari	38
3.5.2	Pàgines exclusives per usuaris registrats	38
3.5.2.1	Pàgina de plantilles	39
3.5.2.2	Modal de filtrar plantilles	39
3.5.2.3	Pàgina de creadors	40
3.5.2.4	Modal de filtrar creadors	40
4	Implementació	41
4.1	Esquema organitzatiu dels fitxers	41
4.1.1	Esquema organitzatiu dels fitxers de la web	41
4.1.2	Esquema organitzatiu dels fitxers de la Api de segmentació	43
4.2	La gestió de la base de dades	44
4.2.1	Firestore Database	44
4.2.1.1	Usuaris	44
4.2.1.2	Plantilles	45
4.2.2	Autenticació	45
4.2.2.1	Autorització	46
4.3	Interfície	47
4.3.1	Adaptació a pantalles de diferents mides	47
4.3.2	Actualització del contingut sense refrescar les pàgines	47
4.3.3	Permisos d'accés	48
4.3.4	Pantalles principals	49
4.3.4.1	Fer Login	49
4.3.4.2	Fer Sign up	50
4.3.4.3	Pantalla Plantilles	51
4.3.4.3.1	Afegir plantilla	51
4.3.4.3.2	Previsualitzar plantilla	52
4.3.4.3.3	Eliminar plantilla	52
4.3.4.3.4	Filtrar plantilla	53
4.3.4.3.5	Utilitzar plantilla	53
4.3.4.4	Pantalla Users	54
4.3.4.4.1	Filtrar usuaris	54
4.3.4.4.2	Eliminar usuaris	55
4.4	Segmentació de factures	56
4.4.1	Web Api de Segmentació	56
4.4.2	De pdf a text	56
4.4.3	Plantilla	57
4.4.4	Funcionament del Regex	58
4.4.5	Estructura de dades dins de les plantilles	60
4.4.6	Extracció a JSON	63

5 Conclusions	64
5.1 Futures ampliacions	65
Glossari	66
Bibliografia	67

Índex de figures

Figura 1: Fases del flux de treball.....	15
Figura 2: Fases de desenvolupament en els sprints escollits.....	16
Figura 3: Fases de desenvolupament en els sprints escollits.....	21
Figura 4: Cas d'ús textual: UC01 - Registrar usuari.....	22
Figura 5: Cas d'ús textual: UC02 - Iniciar sessió.....	23
Figura 6: Cas d'ús textual: UC03 - Eliminar Usuari.....	23
Figura 7: Cas d'ús textual: UC04 - Crear usuari amb rol concret.....	24
Figura 8: Cas d'ús textual: UC05 - Filtrar usuarios.....	24
Figura 9: Cas d'ús textual: UC06 - Eliminar plantilla.....	25
Figura 10: Cas d'ús textual: UC07 - Previsualitzar plantilla.....	25
Figura 11: Cas d'ús textual: UC08 - Crear plantilla.....	26
Figura 12: Cas d'ús textual: UC09 - Filtrar plantilles.....	26
Figura 13: Cas d'ús textual: UC10 - Utilitzar plantilla.....	27
Figura 14: Esquema de la distribució del programari i la distribució entre ells.....	28
Figura 15: Esquema de la distribució del Domain Driven Design.....	29
Figura 16: Exemple de factura.....	32
Figura 17: Exemple dades lineals sense estructura.....	33
Figura 18: Exemple de les línies de la factura anterior.....	33
Figura 19: Exemple de taula de la factura anterior.....	34
Figura 20: Prototip de pantalla d'Inici.....	35
Figura 21: Prototip de pantalla de tutorials.....	36
Figura 22: Prototip de pantalla d'explicació tècnica.....	37
Figura 23: Prototip de pantalla d'inici de sessió.....	37
Figura 24: Prototip de pantalla de creació d'usuari.....	38
Figura 25: Prototip de pantalla plantilles.....	39
Figura 26: Prototip de pantalla de filtrar plantilles.....	39
Figura 27: Prototip de pantalla de creadors.....	40
Figura 28: Prototip de pantalla de filtrar creadors.....	40
Figura 29: Organització de fitxers de la pàgina web.....	41
Figura 30: Organització de fitxers de l'Api de Segmentació.....	43
Figura 31: Col·lecció d'usuarios.....	44
Figura 32: Col·lecció de plantilles.....	45
Figura 33: Usuarios amb sessió creada.....	46
Figura 34: Comparativa ordinador-telèfon.....	47

Figura 35: Visió d'un usuari normal i d'un admin de la pàgina de plantilles.....	48
Figura 36: Pantalla d'inici de sessió.....	49
Figura 37: Pantalla de registre d'usuari.....	50
Figura 38: Pantalla de plantilles.....	51
Figura 39: Modal afegir plantilla.....	51
Figura 40: Previsualitzar plantilla.....	52
Figura 41: Filtrar plantilles.....	53
Figura 42: Pantalla d'ús de plantilla.....	53
Figura 43: Pantalla d'usuaris.....	54
Figura 44: Filtrar usuaris.....	54
Figura 45: Plantilla serialitzada en un OrderDict.....	57
Figura 46: Exemple de taules.....	60
Figura 47: Exemple de línies.....	61
Figura 48: Extracció de les línies.....	62
Figura 49: Json final.....	63

1 Introducció

1.1 Motivacions i context del projecte

L'empresa Virmotos és una empresa familiar de reparació i venda de motocicletes i un dels pocs tallers en tota Espanya en estar especialitzats amb reparacions d'Honda Goldwings, fent que vingui gent de qualsevol punt d'Europa.

He tingut el plaer de poder treballar dins d'aquesta empresa durant alguns estius abans de la meva carrera professional dedicada a l'enginyeria de software. Això va coincidir amb les meves primeres passes amb el món de la programació, fent així algunes tasques de desenvolupament per automatitzar gestions dins de l'empresa com per exemple, un petit programa de gestió d'hores pels operaris, per facilitar l'estat de les reparacions i tenir ben definides les hores invertides per cada una d'elles.

Llavors a partir d'aquest temps treballant allà i xerrades amb el propietari, vaig adonar-me d'algunes de les necessitats en empreses d'aquest estil a on generalment el propietari es veu obligat a fer el treball burocràtic de l'empresa o contractar un servei perquè ho facin per ell, però aquesta última opció hi ha persones que degut una mala experiència no es fien.

Degut això, vaig plantejar-me com seria la millor forma d'automatitzar o eliminar el procediment més enrevessat i repetitiu que té, que era l'extracció d'informació de factures i resguards.

La finalitat d'aquest projecte és estalviar temps a una gran quantitat de gent, ja que és molt aplicable a diferents factors, tanmateix, com a digitalització de factures per tenir un historial de moviments econòmics, per facilitar l'extracció de despeses i costos de la declaració de la renda o bé ser l'eina intermediària amb un sistema comptable en línia com per exemple Holded.com, per això l'eina ha de ser totalment modelable, fent que el seu input sigui el més simple possible, el seu output molt fàcil d'interpretar i habilitar a part de la pàgina web una Web API per fer crides directament per poder-ho implementar a una cadena d'operacions més fàcilment.

Degut aquest objectiu i la seva naturalesa, la funcionalitat ha de ser totalment fiable i resolutiva, perquè hi ha diferents factors de riscos i és informació delicada. Per altra banda,

crec molt amb el projecte, pel fet que, he pogut veure de primera mà que útil pot ser en diverses ocasions.

Actualment, el procediment que s'ha de seguir per digitalitzar factures és fer-ho manualment, és a dir, buscar dins de les factures la informació important, sent totes aquestes diferents segons l'empresa, sense tenir cap format comú i després passar la informació a l'ordinador, fent així una tasca repetitiva amb molta possibilitat d'error i de tardar a identificar la posició de la informació.

En el món actual temps equival a diners, és a dir, quan estàs involucrant el teu temps en una feina amb la qual no et veus beneficiat de cap forma per fer-la, es podria dir que estàs perdent diners. Llavors com més temps es pugui invertir millor, en canvi, de perdre's.

La solució amb què competeix aquest servei és amb la contractació d'un comptable o gestor que et faci l'extracció de dades del seguiment de factures per posteriorment fer qualsevol operació, però això representarien costos força elevats en comparació a la dificultat de la feina realitzada.

1.2 Descripció del projecte

Aquest projecte consisteix en l'elaboració d'un lloc i una aplicació web que permet extreure les dades desitjades d'una o diverses factures.

Aquest software deixa a l'abast de tots els usuaris que s'han registrat amb anterioritat a la web puguin penjar les seves plantilles a la pàgina web a on utilitzant aquestes o d'altres que han penjat els altres usuaris, poden extreure la informació que vulgui de dins d'una factura. Els administradors tindran al seu abast la gestió d'usuaris tant com de plantilles de tot el lloc web.

Actualment qualsevol persona té a la seva disposició d'un aparell que pot accedir a internet, a causa d'una sèrie de situacions econòmiques i socials les eines de gestoria, burocràtiques i les factures han passat a ser en línia, igualment dins d'una factura ens interessa una sèrie de dades en concret segons què necessitem fer amb ella i generalment no s'accepten les factures senceres sinó les dades d'elles. Per això ens trobem en diferents moments que hem de passar una a una aquestes dades, fent així una tasca carregosa sobretot si tens una gran quantitat d'elles.

1.3 Objectius

1.3.1 Objectius generals

La finalitat principal d'aquest projecte és cobrir les necessitats tant com de particulars com a petites empreses explicades a la descripció del projecte. Tot el disseny del software desenvolupat s'ha fet amb perspectiva d'evolució al futur, és a dir, l'aplicació de noves funcionalitats i el manteniment del programari ha de ser fàcil i adaptable.

1.3.2 Objectius específics

Inicialitzar-me o perfeccionar diverses tecnologies de desenvolupament d'aplicacions web, com poden ser:

- Web Apis.
- Bases de dades noSQL.
- Angular Framework
- Maquetació
- ASP NET. Core

I descobrir eines totalment noves o millorar l'ús que feia fins ara, per resoldre el problema plantejat:

- YAML i JSON
- Encoders UTF-8 i Unicode
- Parsers
- Regex

1.4 Anàlisi de riscos

Risc	Fer malament les estimacions de temps o no tenir el temps necessari
Descripció	El projecte ha d'estar finalitzat en tres mesos, temps que podria ser just a causa d'estar treballant a jornada completa mentre vaig desenvolupant el treball. A més el projecte conté moltes tasques que representen ser molt meticulós amb el procés a prendre per fer-les, a causa de la sensibilitat de les dades que es requereixen en el programari.
Impacte	Data de finalització i entrega
Solució	Utilitzar dies de vacances laborals per invertir de dedicació total al projecte.

Risc	Dificultat amb l'experiència d'usuari fent que els usuaris no es familiaritzin amb el servei
Descripció	Els usuaris no es familiaritzen amb la interfície d'usuari.
Impacte	Tot el projecte
Solució	Avaluar l'enteniment del disseny de la interfície gràfica a través de diferents A/B Tests utilitzant Figma, ja que, dona la possibilitat de fer dissenys dinàmics i compartir-los. Un cop es tenen els resultats dels A/B Tests, intentar millorar els dissenys encara que siguin els escollits a través dels comentaris dels usuaris testats.

Risc	Arquitectura i disseny del software poc escalable
Descripció	A causa de la naturalesa del projecte hi ha moltes funcionalitats que es poden implementar pel futur, un cop que s'hagi finalitzat el projecte. Per tant, la implementació ha de facilitar la modificació i l'adaptabilitat a nous canvis.
Impacte	El projecte sencer
Solució	Fer un disseny molt detallat i tenir idea molt clara de l'organització del codi

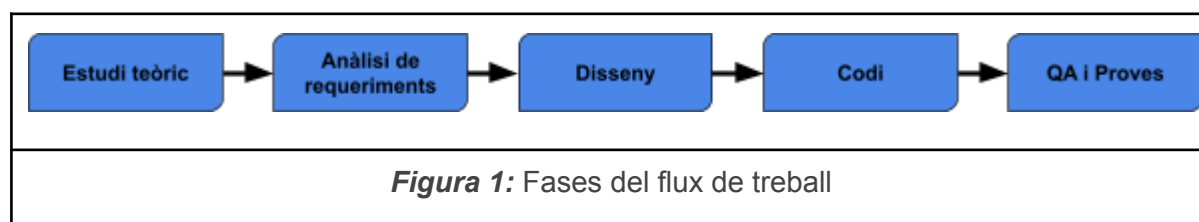
des de l'inici de la fase d'escriure codi.

Risc	Creació de plantilles poc intuïtiva
Descripció	A causa de la naturalesa del projecte, és necessari que la màxima quantitat de persones siguin capaces de generar plantilles funcionals i pujar-les a la base de dades, així que les plantilles han de ser fàcils d'entendre i crear.
Impacte	Tenir poques plantilles a la base de dades, fent que poca gent accedeixi.
Solució	Donar-li la màxima quantitat d'eines a l'usuari perquè compregui la creació de plantilles.

Risc	Mala elecció de tecnologies pel desenvolupament.
Descripció	L'elecció de tecnologies és una part complexa de l'inici del projecte que marcarà el camí que es farà amb aquest, una mala elecció potser crítica.
Impacte	Pèrdua de temps o reinventar la roda.
Solució	Com que un dels objectius del projecte és crear un programari escalable és necessari que abans d'aplicar una nova tecnologia fer una investigació amb els recursos que seran necessaris en el futur, fent així un programari més organitzat, òptim i flexible.

1.5 Enfocament i mètode seguit

L'enfocament del projecte serà orientat a la planificació, remarcar al principi els seus límits i els terminis de temps per cada una de les etapes del flux de treball establert. Aquestes etapes es poden veure en la **Figura 1**.



La primera etapa del flux fa referència a la projecció i estudi teòric per poder fer les aproximacions de temps de la resta de les etapes, si fem un paral·lelisme amb la metodologia Agile aquesta fase seria l'etapa de plantejament i valoració d'història d'usuaris, ja que sense coneixement de causa no podem fer cap aproximació de cap mena.

La segona etapa és quan s'especifica les característiques operacionals del programari per la solució de les necessitats o condicions donades per un producte nou o el seu manteniment.

La tercera etapa és dissenyar aquesta solució plantejada a l'etapa anterior amb projecció d'optimitzar tant com computacionalment, com millorar el plantejament a futur, per tenir una aplicació organitzada, escalable i entenedora.

La quarta etapa s'aplica el disseny plantejat al codi.

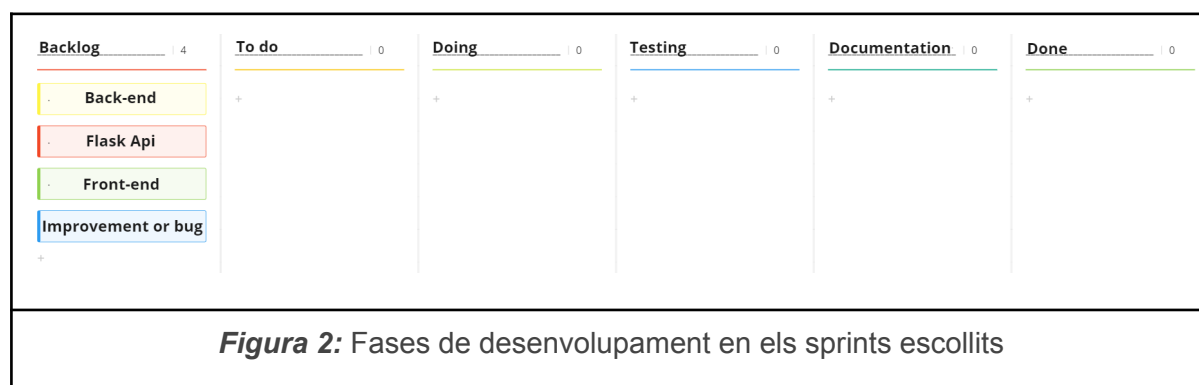
L'última fase plantejada al nostre flux, és **Quality Assurance** a on es comprova que el codi sigui sòlid i no tingui falles o bugs i per últim, proves per assegurar-nos l'adaptabilitat i nivell de comprensió.

No s'ha afegit l'última etapa real que és el manteniment, degut a la llargada del projecte i que actualment no seria necessària.

1.6 Planificació temporal

1.6.1 Distribució del temps

La planificació ha sigut amb metodologia Agile, en concret un sistema he aplicat un SCRUM de sprints de dues setmanes amb monitoratge d'històries d'usuari en una taula Kanban per tenir ben fixats els objectius de cada setmana, aquesta taula es separa pels apartats de desenvolupament que es poden veure a continuació i el tipus de tasca que és segons el seu color:



1.6.2 Fites i dates de lliurament

1. Disseny de la plantilla amb els camps necessaris - (23/2/22)
2. Extreure l'expressió regular dels camps en un string normal - (10/3/22)
3. Fer implementació de pdftotext i codificació - (13/3/22)
4. Extreure els valors estàtics, els regex bàsics i configuracions - (20/3/22)
5. Crear una plantilla amb una factura completa d'HolaLuz - (21/3/22)
6. Utilitzar la plantilla creada per la serialització de la plantilla en el script- (31/3/22)
7. Fer la segmentació de taules i afegir a la plantilla - (7/4/22)
8. Fer la segmentació de línies i afegir a la plantilla - (16/4/22)
9. Extreure el contingut final en un json - (18/4/22)
10. Envoltar el script de segmentació amb una web api de Flask i generar una crida que ens retorna el json final - (21/4/22)
11. Crear el projecte per la pàgina web amb arquitectura DDD amb injecció de dependències - (28/4/22)
12. Configuració d'objectes, mappings, logger i de base de dades - (30/4/22)
13. Crear les consultes necessàries de gestió plantilles dins de la base de dades - (1/5/22)
14. Crear les consultes necessàries de gestió usuaris dins de la base de dades - (1/5/22)
15. Configuració de jwt i gestió d'autenticació (Login i Sign up) - (8/5/22)

- 16.** Assignar rols al jwtoken i fer que les crides més sensibles no siguin cridades per tokens sense permís - (9/5/22)
- 17.** Desenvolupar pantalles bàsiques sense permisos - (12/5/22)
- 18.** Desenvolupar una pantalla de login i sign up i fer-les operables - (15/5/22)
- 19.** Desenvolupar pantalles d'usuaris i plantilles - (17/5/22)
- 20.** Mostrar totes les plantilles dins de la taula - (18/5/22)
- 21.** Previsualitzar plantilla des del frontend - (19/5/22)
- 22.** Afegir template des del frontend - (20/5/22)
- 23.** Eliminar template des del frontend - (20/5/22)
- 24.** Crear la crida en el backend a la Flask Api - (22/5/22)
- 25.** Descarregar json de factura segmentada des del frontend - (22/5/22)
- 26.** Omplir la taula amb els usuaris creats - (19/5/22)
- 27.** Eliminar usuaris des del frontend - (23/5/22)
- 28.** Filtrar dades de les taules - (27/5/22)
- 29.** Fer que només admins puguin accedir a la pàgina users i puguin veure el creador de la plantilla - (1/6/22)

2 Anàlisi del sistema

2.1 Requeriments inicials

2.1.1 Funcionals

Els requisits funcionals són les funcionalitats principals que ha d'oferir el programari. Aquests estan dividits segons el seu camp d'ús i tema:

2.1.1.1 Usuaris

- S'ha de crear un usuari per poder accedir a les funcionalitats principals de la web.
- S'ha de tenir referència d'aquest usuari amb una identificació única.
- S'ha de tenir la capacitat de moderar els usuaris.
- L'usuari hauria de tenir un rol assignat segons els seus permisos.

2.1.1.2 Aprenentatge

- L'aprenentatge és clau en aquesta web i es necessita la contribució de moltes persones per tenir més potència, per això s'ha de donar un servei que aclareixi tots els dubtes i doni la quantitat més gran d'eines a l'usuari perquè el procés d'ús sigui el més senzill possible per ell.

2.1.1.3 Gestió de plantilles

- S'ha de tenir la capacitat de visualitzar una plantilla en qualsevol moment.
- Crear una plantilla a través d'un fitxer.
- Poder obtenir totes les plantilles que tenim guardades a la base de dades.
- Eliminar qualsevol de les plantilles guardades a la base de dades.
- Tenir la capacitat de poder filtrar per nom i data de creació de la plantilla.

2.1.1.4 Gestió d'usuaris

- Crear un usuari a través del sign up.
- Poder obtenir tots els usuaris que tenim guardades a la base de dades.
- Eliminar qualsevol dels usuaris guardats a la base de dades.
- Tenir la capacitat de poder filtrar pels camps característics de l'usuari.

2.1.1.5 Segmentació de factures

- Necessitem un sistema de detecció de patrons dins de text.
- Desenvolupar una sèrie de possibles representacions de dades a dins de les factures i com extreure en aquell cas concret la informació, taules, línies...
- Tenir algun sistema de culturització segons certes coses culturals com podria ser el separador de decimals, la moneda...
- Reunir totes les coincidències trobades dels patrons dins d'un json.

2.1.2 No funcionals

Aquests requeriments s'associen a característiques del programari que no ens aporta valor en l'àmbit de funcionalitat, però facilita o s'assegura d'un bon ús del programari.

2.1.2.1 Seguretat

- L'accés a la gestió de la informació tant com a usuaris com a les plantilles serà delimitada segons el rol de qui ho sol·licita, extret per l'inici de sessió d'aquest.
- Tenir autenticades les crides que requereixen l'inici de sessió a través d'un sistema de tokens.
- Tenir la contrasenya totalment coberta i que no sigui accessible per ningú.

2.1.2.2 Interfície

- Pàgines adaptables a diferents resolucions de pantalla.
- Visió clara i contundent des de qualsevol dispositiu.

2.1.3 Tècnics

Les eines de desenvolupament escollides en les quals s'implementarà i es gestionarà tant la pàgina web i la web Api seran:

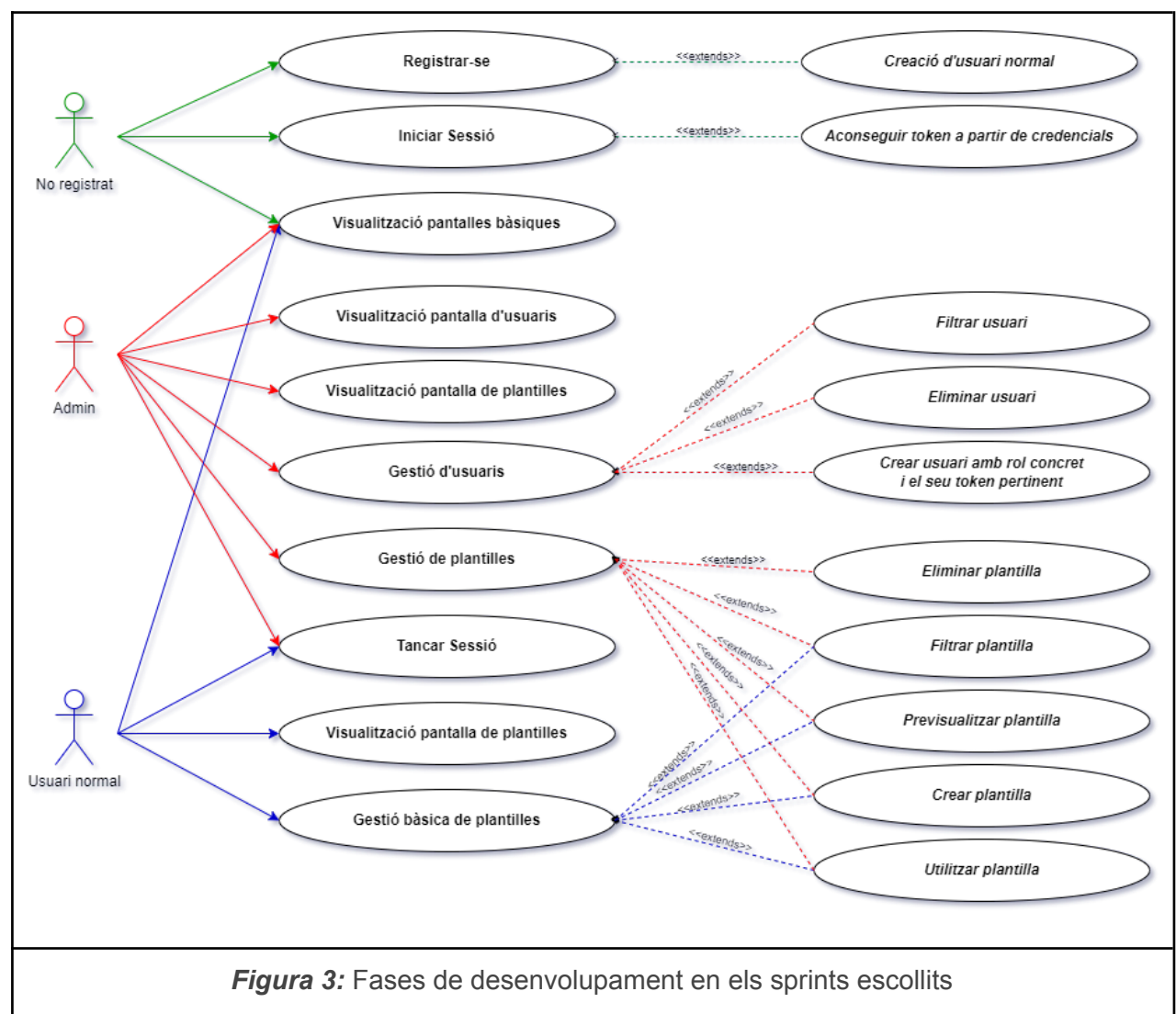
- **Backend de la pàgina web:**
 - ASP .Net Core 5, és un marc multiplataforma de codi obert d'alt rendiment que té com a finalitat la compilació d'aplicacions modulars connectades a Internet i habilitades pel Cloud, basat en C#.
- **Frontend de la pàgina web:**
 - Angular, és un Framework de JavaScript de codi obert escrit en TypeScript.
 - HTML5, és un llenguatge d'estructuració de textos, que defineixen l'estructura de l'aspecte de la pàgina web.
 - CSS, és un llenguatge de fulls d'estils que descriu l'aspecte i format de la web.
 - Bootstrap, és una biblioteca multiplataforma de codi obert que dona disponibilitat a components del front ja fets.
- **Base de dades:**
 - Firebase, és un servei freemium que ofereix Google que conté diferents tipus de bases de dades segons les nostres preferències en el projecte, en el meu cas he utilitzat el servei d'autenticació i la base de dades noSQL de documents Firestore.
- **Backend de la API de segmentació:**
 - Python 3.9, és un llenguatge de programació d'alt nivell, en aquest projecte he implementat un entorn generat per Anaconda amb totes les llibreries necessàries per la segmentació i l'Api.
 - Flask, és un framework escrit en Python que permet generar una Web Api a través de les eines que et dona.

2.2 Casos d'ús

És necessari definir els usuaris del sistema i les accions que es poden dur a terme. Això requereix el desenvolupament d'un diagrama de casos d'ús i la descripció d'aquests casos d'ús. A través d'aquests casos d'ús podrem observar si el programari satisfà alguns dels requeriments mencionats anteriorment.

2.2.1 Diagrama de casos d'ús

El següent diagrama de casos d'ús mostra els tipus actors que poden interactuar en el programari, és a dir, els possibles usuaris i les funcionalitats que tenen la capacitat de fer cada un d'ells.



2.2.2 Descripció dels casos d'ús textuais

Descripció: UC01 - Registrar usuari	
Actors:	Usuari no registrat
Precondicions:	L'usuari ha d'haver accedit al lloc web i no tenir cap usuari donat d'alta en el sistema.
Flux bàsic:	1. Accedir a la pàgina d'inici de sessió a través del botó de més a la dreta de la barra de navegació. 2. Fer clic sobre el botó de crear un compte. 3. Omplir el formulari amb la informació desitjada i fer clic sobre el botó de crear un compte 4. Redirigeix a la pàgina de plantilles amb l'inici de sessió fet.
Flux alternatiu:	1b i 2b. Accedir a partir de l'URL amb l'extensió /signup. 3c. S'omple el formulari, però les contrasenyes no són iguals i es fa clic sobre el botó de crear un compte. 4c. No fa cap acció fins que es posin les dades bé i es torni a clicar. Apunt. Es pot fer una crida en el backend de la web amb un token amb drets d'administrador passant-li un json amb els atributs de l'usuari que vulguem crear i així es poden crear administradors, d'altre forma sempre es creen usuaris normals.
Postcondicions:	L'usuari passa a estar amb la sessió oberta.

Figura 4: Cas d'ús textual: UC01 - Registrar usuari

Descripció:	UC02 - Iniciar sessió
Actors:	Usuari no registrat
Precondicions:	Tenir un usuari donat d'alta en el sistema
Flux bàsic:	1. Accedir a la pàgina d'inici de sessió a través del botó de més a la dreta de la barra de navegació. 2. Omplir el formulari amb les credencials corresponents a un usuari i es fa clic sobre el botó Entrar. 3. Redirigeix a la pàgina de plantilles amb la sessió iniciada.
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió /login. 2c. Omplir el formulari amb les credencials que no corresponen a un usuari i es fa clic sobre el botó Entrar. 3c. No fa res fins que es posi la informació correcta i es torni a fer clic. 2,5. Si es dóna a la checkbox Recorda'm, mantindrà la sessió iniciada.
Postcondicions:	L'usuari passa a estar amb la sessió oberta.

Figura 5: Cas d'ús textual: UC02 - Iniciar sessió

Descripció:	UC03 - Eliminar Usuari
Actors:	Administrador
Precondicions:	Estar amb una sessió iniciada d'administrador.
Flux bàsic:	1. Accedir a la pàgina d'usuaris 2. En la taula d'usuaris donar-li al botó amb la icona d'una brossa dins de la fila de l'usuari que volem eliminar.
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió app/community.
Postcondicions:	L'usuari desapareix de la taula i ja no pot iniciar sessió amb el compte eliminat.

Figura 6: Cas d'ús textual: UC03 - Eliminar Usuari

Descripció:	UC04 - Crear usuari amb rol concret
Actors:	Administrador
Precondicions:	Tenir un token amb permisos d'administrador
Flux bàsic:	1. Fer una crida al backend de la pàgina web amb un token d'administrador. 2. Passant per paràmetre en el còs de la petició un json amb els atributs i els seus valors de l'objecte usuari. 3. Enviar la petició.
Flux alternatiu:	2b. Si posem un correu electrònic ja existent no crearà cap usuari.
Postcondicions:	S'afegeix l'usuari amb els permisos i els valors dels atributs que li haguem donat dins del Firebase.

Figura 7: Cas d'ús textual: UC04 - Crear usuari amb rol concret

Descripció:	UC05 - Filtrar usuarios
Actors:	Administrador
Precondicions:	Estar amb una sessió iniciada d'administrador.
Flux bàsic:	1. Accedir a la pàgina d'usuaris 2. Fer clic sobre el botó "Filtrar" 3. A la modal desplegada fer la cerca segons els paràmetres que vulguis. 4. Clicar el botó "Aplicar filtros"
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió app/community.
Postcondicions:	En la taula sortiran només aquells usuaris que compleixin la cerca feta en la modal passada.

Figura 8: Cas d'ús textual: UC05 - Filtrar usuarios

Descripció:	UC06 - Eliminar plantilla
Actors:	Administrador
Precondicions:	- Estar amb una sessió iniciada d'administrador. - Hi hagi una plantilla mínim.
Flux bàsic:	1. Accedir a la pàgina de plantilles 2. En la taula de plantilles donar-li al botó amb la icona d'una brossa dins de la fila de la plantilla que volem eliminar.
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió app/templates.
Postcondicions:	La plantilla desapareix de la taula i ja no forma part de la base de dades.

Figura 9: Cas d'ús textual: UC06 - Eliminar plantilla

Descripció:	UC07 - Previsualitzar plantilla
Actors:	Administrador i usuari normal.
Precondicions:	- Estar amb una sessió iniciada. - Hi hagi una plantilla mínim.
Flux bàsic:	1. Accedir a la pàgina de plantilles 2. En la taula de plantilles donar-li al botó en el qual hi ha el text "Previsualitzar" dins de la fila de la plantilla que volem veure.
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió app/templates.
Postcondicions:	S'obre una modal a on es pot veure el contingut del yaml.

Figura 10: Cas d'ús textual: UC07 - Previsualitzar plantilla

Descripció:	UC08 - Crear plantilla
Actors:	Administrador i usuari normal.
Precondicions:	Estar amb una sessió iniciada.
Flux bàsic:	1. Accedir a la pàgina de plantilles. 2. Fer clic sobre el botó amb el text “Afegir Plantilla”. 3. S’obre una modal per pujar un fitxer. 4. Arrosseguem un fitxer de tipus YML dins de l'àrea amb interlineat de la modal. 5. Li fem clic sobre el botó Acceptar.
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió app/templates. 4c. Fem clic sobre el botó “Busqueu el fitxer”. 4.1c. Seleccionem en l'explorador d'arxius el YML.
Postcondicions:	S'afegeix a la base de dades la plantilla i s'actualitza la taula.

Figura 11: Cas d'ús textual: UC08 - Crear plantilla

Descripció:	UC09 - Filtrar plantilles
Actors:	Administrador i usuari normal.
Precondicions:	Estar amb una sessió iniciada.
Flux bàsic:	1. Accedir a la pàgina de plantilles. 2. Fer clic sobre el botó “Filtrar” 3. A la modal desplegada fer la cerca segons els paràmetres que vulguis. 4. Clicar el botó “Aplicar filtres”
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió app/templates.
Postcondicions:	En la taula sortiran només aquelles plantilles que compleixin la cerca feta en la modal passada.

Figura 12: Cas d'ús textual: UC09 - Filtrar plantilles

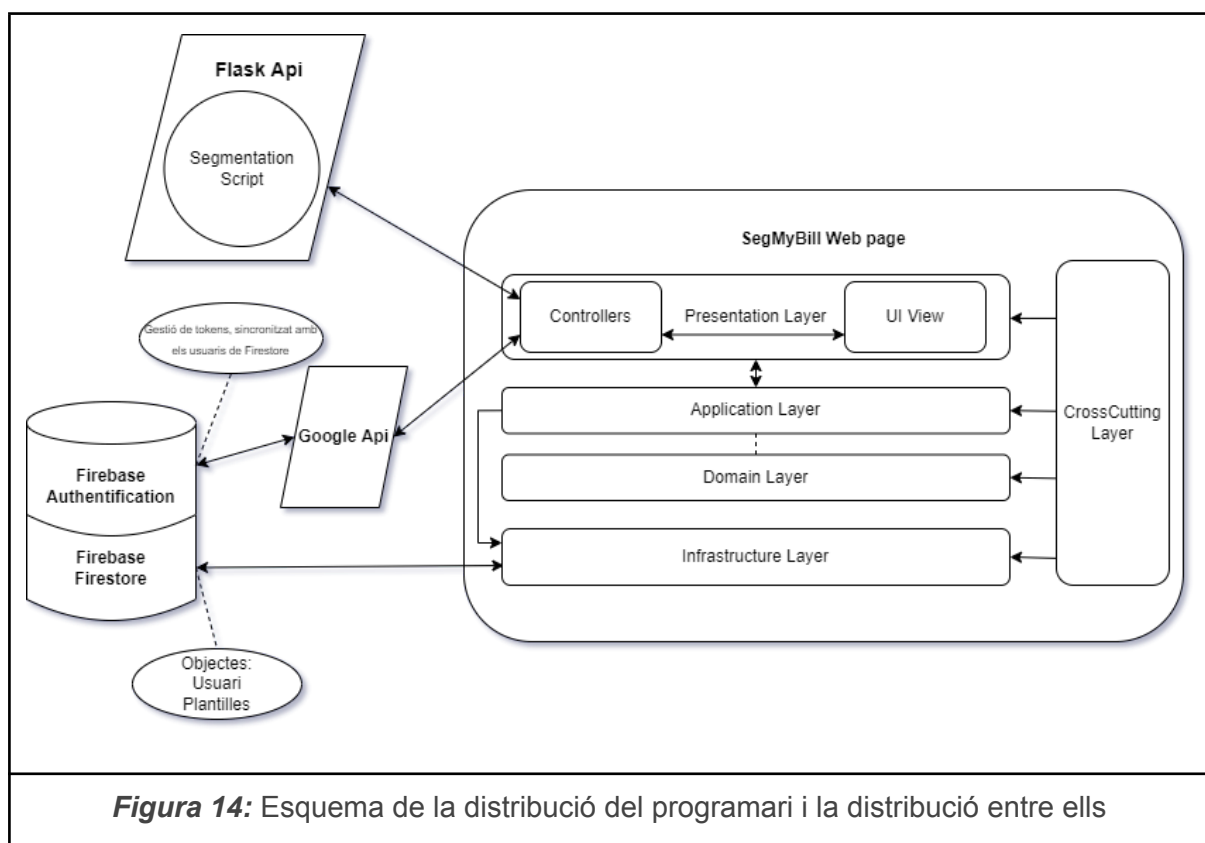
Descripció:	UC10 - Utilitzar plantilla
Actors:	Administrador i usuari normal.
Precondicions:	- Estar amb una sessió iniciada. - Hi hagi una plantilla mínim.
Flux bàsic:	1. Accedir a la pàgina de plantilles 2. Fer clic sobre el botó amb el text "Afegir Utilitzar" en la fila de la plantilla que vulguem utilitzar. 3. S'obre una modal per pujar un fitxer PDF. 4. Arrosseguem un fitxer de tipus PDF dins de l'àrea amb interlineat de la modal. 5. Li fem clic sobre el botó Acceptar.
Flux alternatiu:	1b. Accedir a partir de l'URL amb l'extensió app/templates. 4c. Fem clic sobre el botó "Busqueu el fitxer". 4.1c. Seleccionem en l'explorador d'arxius el PDF de la factura corresponent a la plantilla.
Postcondicions:	Es descarrega un arxiu JSON automàticament amb la resposta de la crida, si tot ha anat correctament el JSON conté la informació de la factura desglossada.

Figura 13: Cas d'ús textual: UC10 - Utilitzar plantilla

3 Disseny

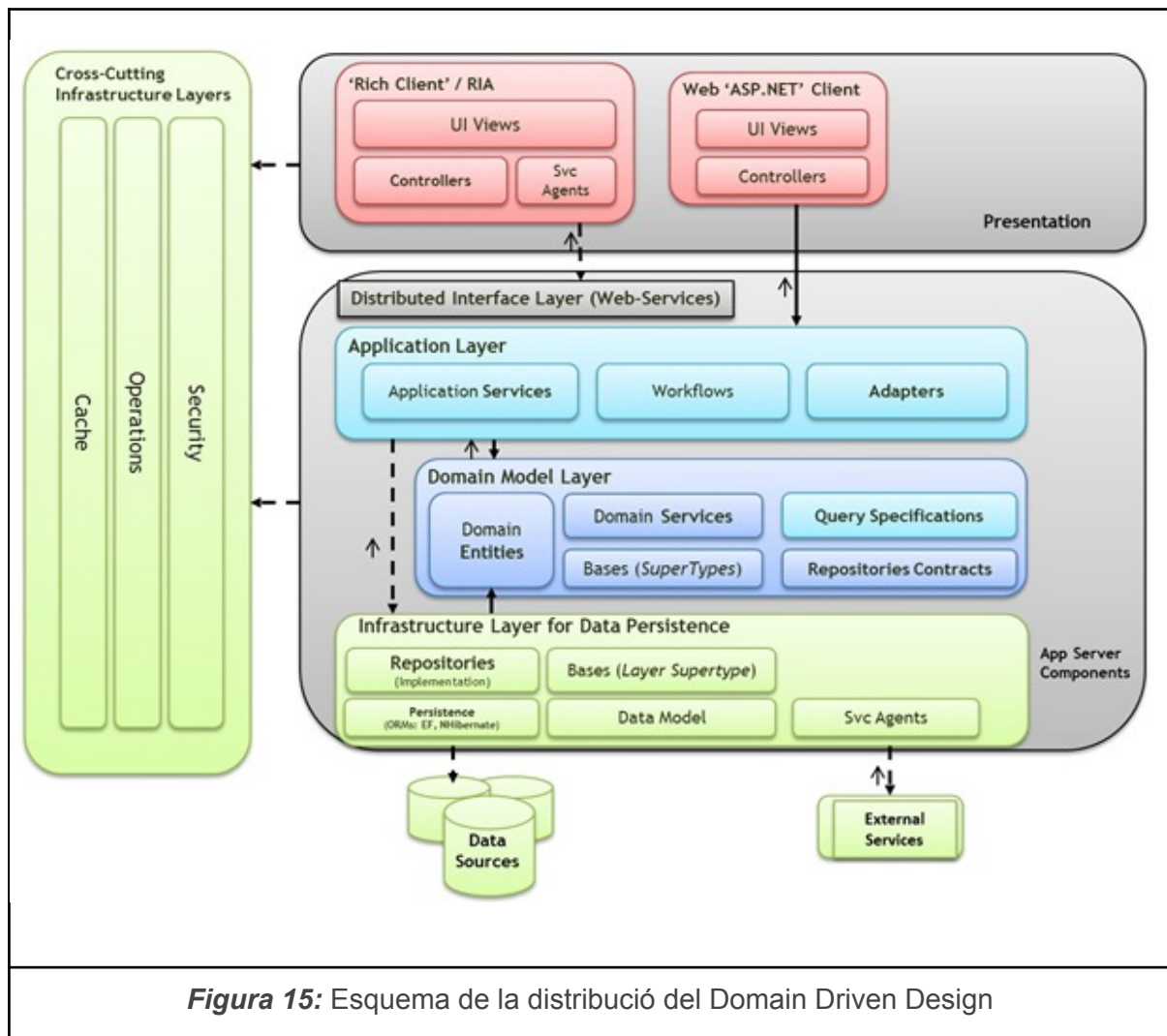
3.1 Diagrama del projecte

Com que vull donar un servei de tipus DIY (do it yourself) i fer el màxim accessible el sistema, he dissenyat una estructura que sigui capaç de separar capacitats i he decidit fer la segmentació a una Web Api, ja que si un desenvolupador necessita l'eina de segmentació li pot donar servei sempre que faci una crida amb les seves pròpies plantilles. Per altra banda, està la pàgina web, que dóna accés a una base de dades amb diferents plantilles per no tenir necessitat de fer-les cada cop que es vulgui utilitzar en una factura d'una nova empresa, d'aquesta forma se li dóna a l'usuari bàsic una ajuda a utilitzar el servei.



3.2 Arquitectura

El disseny del sistema que he utilitzat ha sigut el DDD (Domain Driven Design), aquest disseny ens ajuda, en primer lloc, a definir el domini exhaustivament. Ens permet també conèixer de manera extensa el problema de negoci de manera que puguem dividir-lo en subdominis amb l'objectiu de maximitzar el desacoblament entre els subdominis i assolir així solucions modulars, que premien l'adaptabilitat de cada subdomini i, per tant, l'adaptabilitat de la solució en general.



3.2.1 Descripció de les capes:

- **Presentation Layer:** Defineix els treballs que suposa que el programari ha de fer i dirigir els objectes de domini perquè resolguin els problemes. Les tasques que són responsabilitat d'aquest nivell són importants referent al que aporta com a valor al programari i necessari per a les interaccions entre diferents sistemes.
- **Application Layer:** Defineix les normes de negoci i s'encarrega de fer procediments interns que no tenen a veure amb la persistència de dades.
- **Domain Layer:** Defineix les normes de mètodes de persistència de dades i les entitats que es modificaran i que posteriorment seran usades per la serialització a la base de dades amb el seu model concret de l'entitat.
- **Infrastructure Layer:** És la capa encarregada del control de persistència de dades, implementacions i gestió de serveis externs.
- **Cross-Cutting Layer:** És la capa que serà totalment transversal a les altres capes, aquí és on està majoritàriament les funcions d'utilitat, per exemple, els registres, funcions utilitàries per control d'arxius...

3.2.2 Injecció de dependències

La injecció de dependències és un principi que ens ajudarà a disminuir l'acoblament entre les capes i algunes interfícies de l'arquitectura, això ens millorarà el control d'instàncies d'objectes i fent injectables algunes classes a dins d'altres per extreure la seva utilitat.

Per aplicar aquest principi ASP .Net Core ja té implementat una llibreria que ho pot gestionar, però en aquest cas he utilitzat AutoFac, una llibreria que facilita la gestió de les dependències a través de perfils per capes i a més és importable per generar tests d'implementació.

3.2.3 Mapping

Cada capa principal del programari conté un tipus d'entitat que quan una capa es comunica amb una altra és necessari transformar l'entitat al tipus d'objecte d'aquella capa, això ofereix més flexibilitat i gestió d'informació necessària per capa.

- **Data Transfer Object (DTO):** Ens servirà com a objecte d'entrada i sortida del backend.
- **Entity:** Objecte que ens servirà per a la lògica de negoci de la capa d'aplicació.
- **Model:** Objecte serializable a la base de dades.

3.4 Prototipat de plantilles

He escollit els fitxers tipus YAML, que és un llenguatge de serialització de dades. Inicialment, aquest nom venia de “Yet Another Markup Language”, ja que, a l’inici de la història d’aquest llenguatge suposava una altra solució respecte als llenguatges de marques preferents de l’època, HTML i XML a l’hora de serialitzar dades. La raó principal a l’hora d’escollir aquest tipus de fitxers ha sigut que semànticament sembla més simple i intuïtiu per gent sense un bagatge de programació o informàtica. A més els arxius YAML estan a l’ordre del dia dins del món del Cloud com per exemple a la creació de plantilles en les pipelines CI/CD d’Azure Cloud, fent així que hi ha molt de suport en molts llenguatges diferents.



Tu factura de Gas

Factura número:
2022G03610028

Fecha de emisión:
22/3/2022

Datos del cliente
Cliente:
NIF:
Domicilio:

Datos de suministro
CUPS:
Domicilio:

Tarifa de acceso:
Titular:
NIF:

Datos de pago
Vencimiento:
Forma de pago:
Cuenta bancaria:

Periodo de facturación
1/3/2022 a 4/3/2022

Contrato
5608120220

Cliente:
NIL VIÑALS NIN
41353135G
Calle De La Independencia núm. 2,
04-4

ES0217010003580422WG
Calle De La Independencia núm. 2,
04-4
RL01
Nil Viñals Nin
41353135G

30/03/2022
Recibo domiciliado
1881 xxxx xx xxxxxx0182



900 64 92 92

clientes@holaluz.com

@Holaluzcom

/Holaluzcom

Total a pagar por tu factura de gas del mes de marzo:

11,56 €

Concepto	Base	Precio unitario	Importe
Término fijo	4 días	0,088476 €/día	0,35 €
Término de energía	50,83 kWh	0,177 €/kWh	9,00 €
Impuesto especial sobre hidrocarburos	50,83 kWh	0,00234 €/kWh	0,12 €
Alquiler del contador	4 días	0,020055 €/día	0,08 €
IVA	9,55 €	21%	2,01 €
Total factura			11,56 €

Tus lecturas

Núm. contador
Lectura desde 01-03-2022
Última lectura real 04-03-2022
Lectura a facturar
Gas disfrutado

4104395

1.990,48

1.995,00

1.995,00

4,52 m³

Peajes y Cargos

De acuerdo con lo establecido en la Resolución de 27 de mayo de 2021, de la Comisión Nacional de los Mercados y la Competencia, por la que se establecen los peajes de acceso a las redes de transporte, redes locales y regasificación para el año de gas 2022, los precios de los peajes son:
0,33 € en concepto de peaje por término fijo.
0,85 € en concepto de peaje por término de energía.
De acuerdo con la Orden TED/1023/2021, de 27 de septiembre, por la que se establecen los cargos del sistema gasista y la retribución y los cánones de los almacenamientos subterráneos básicos para el año de gas 2022, los precios de los cargos son:
4 días * 0,001973 €/día = 0,01 € en concepto de cargos por término fijo.
La Tasa de la Comisión Nacional de los Mercados y la Competencia es de 0,140 %
La cuota para la financiación de la retribución del Gestor Técnico del Sistema es de 1,093 %

Figura 16: Exemple de factura

Donada la factura que podem veure a la **Figura 16**, és necessari detectar patrons a partir de la informació que li posem al yaml.

A través d'investigar vaig trobar el concepte RegEx. Aquest nom ve de regular expressió, traduït al català, expressió regular, aquesta és una representació, segons regles sintàctiques d'una porció de text, a on es poden trobar diferents llibreries de detecció de patrons a partir d'aquest concepte. La seva utilització serveix tant per acreditació de camps en un formulari com de cerca d'un patró concret dins del text.

Per acabar, era important establir algun tipus d'estructura dins de la plantilla que segons el seu marcat tingui un comportament o un altre. Els patrons més repetitius en una factura són informació sense estructura, les línies i les taules.

Sense estructura: Es basa d'informació que el seu camp es pot trobar amb un patró lineal i a través d'una referència propera es pot extreure el seu valor.

Datos de suministro	
CUPS:	ES0217010003580422WG
Domicilio:	Calle De La Independencia núm. 2, 04-4
Tarifa de acceso:	RL01
Titular:	Nil Viñals Nin
NIF:	41353135G

Figura 17: Exemple dades lineals sense estructura

Línies: És la repetició de files unes sobreposades a les altres, sense relació entre elles només pels atributs marcats per les columnes i cada fila conté informació relacionada al mateix concepte.

Concepto	Base	Precio unitario	Importe
Término fijo	4 días	0,088476 € /día	0,35 €
Término de energía	50,83 kWh	0,177 € /kWh	9,00 €
Impuesto especial sobre hidrocarburos	50,83 kWh	0,00234 € /kWh	0,12 €
Alquiler del contador	4 días	0,020055 € /día	0,08 €
IVA	9,55 €	21%	2,01 €
Total factura			11,56 €

Figura 18: Exemple de les línies de la factura anterior

Taules: Útils en el cas que la informació està continguda sota del títol d'aquesta, es poden donar casos que es repeteixi exposant-se més casos als seus costats acoblant-se.

Factura número: 2022G03610028	Período de facturación 1/3/2022 a 4/3/2022
Figura 19: Exemple de taula de la factura anterior	

3.5 Prototipat gràfic

Per fer el prototip visual del frontend, he fet segregació implementacions segons l'ús, intentar que sigui una eina comprensible per tothom encara que tingui una dificultat mitjana per iniciar-se i donar eines perquè l'escala d'aprenentatge sigui la més ràpida possible.

He fet el prototipat amb l'eina més coneguda professionalment anomenada Figma. En les figures de continuació es poden veure pantalles del prototipat en diferents resolucions, que corresponen a ordinador, tauleta i telèfon, amb aquest ordre.

S'ha de tenir en compte que no totes les pàgines mostrades estan a la vista de tothom, hi ha tres nivells:

- **Pública:** Tothom que entra a la pàgina pot accedir a elles i interactuar.
- **Usuaris registrats:** Només disponibles pels usuaris que han creat amb anterioritat un compte al lloc web.
- **Administradors:** Només disponibles per usuaris amb el rol d'administrador.

3.5.1 Pàgines públiques

3.5.1.1 Pàgina d'inici



La **Figura 20** il·lustra el prototip de la pàgina d'inici a on es pot veure una petita introducció sobre el projecte i la barra de navegació a on pots accedir a totes les altres pàgines de la web.

3.5.1.2 Pàgina de tutorials



Figura 21: Prototip de pantalla de tutorials

La **Figura 21** il·lustra la pàgina que es dedica de la part d'ensenyar el coneixement bàsic per la creació de plantilles YAML, aquest ensenyament estarà en format de vídeo explicatiu, intentant ser el més lleuger i més entenedor possible. És de vital importància que la majoria de la comunitat tinguin aquests coneixements, per tenir una base de dades gran i per la gent que no és molt habitual, hi hagi més probabilitats que tinguem la plantilla que ell necessiti.

3.5.1.3 Pàgina de funcionament



Figura 22: Prototip de pantalla d'explicació tècnica

La **Figura 22** mostra una pàgina amb una explicació tècnica del funcionament de la web.

3.5.1.4 Pàgina d'inici de sessió



Figura 23: Prototip de pantalla d'inici de sessió

La **Figura 23** il·lustra com és el procés d'inici de sessió amb la possibilitat de registrar-se o iniciar sessió a través del compte de Google.

3.5.1.5 Pàgina de crear usuari



Figura 24: Prototip de pantalla de creació d'usuari

A la **Figura 24** es pot observar l'última de les pàgines públiques, la creació d'usuari dins de la pàgina web, com he mencionat anteriorment també hi ha l'opció de registrar-se a partir del compte de Google segons com et sentis més còmode.

3.5.2 Pàgines exclusives per usuaris registrats

Tots aquests dissenys estan fets a partir del punt de vista d'un administrador, en cas de ser un usuari sense rol d'administrador no tindria capacitat d'eliminar plantilles i no seria capaç de veure la taula d'usuaris (Pantalla de creadors). El plantejament donat respecte a les taules en visualització de telèfon seria el mateix prototip amb un desplaçament horitzontal per poder ser totalment entenedor amb les dades.

3.5.2.1 Pàgina de plantilles

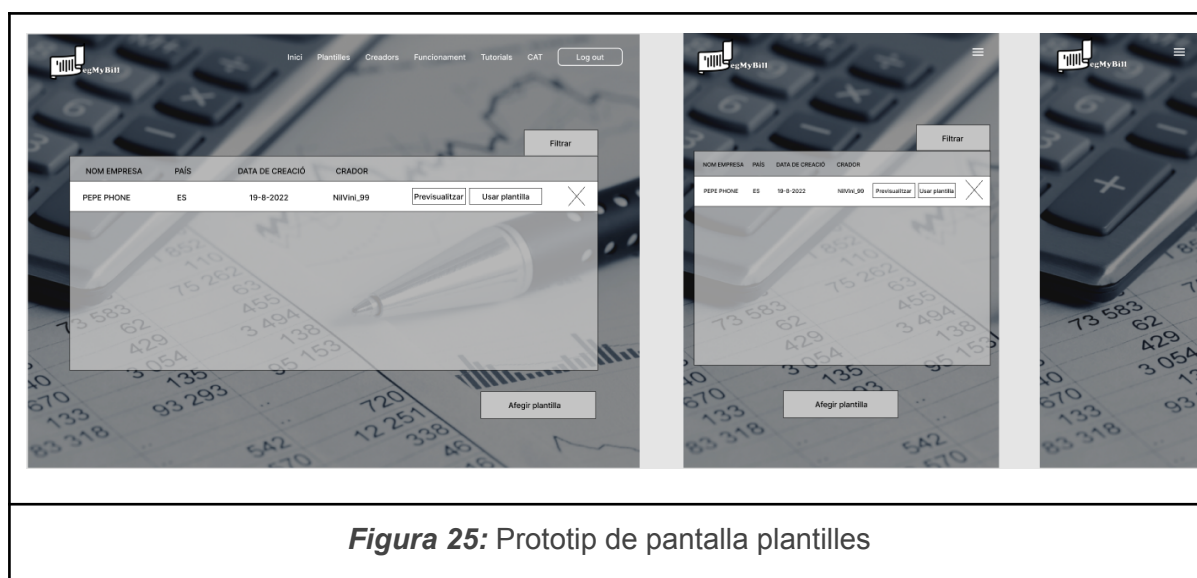


Figura 25: Prototip de pantalla plantilles

A la **Figura 25** és la pantalla a on els usuaris registrats poden veure totes les plantilles ja creades a la web dins d'una taula. En aquesta taula es poden observar els diferents camps més importants de les plantilles, també es pot previsualitzar l'estructura del YAML de la plantilla, utilitzar la plantilla en concret per una o diverses factures (obrint-se un explorador d'arxius per seleccionar les factures a segmentar i un cop fet l'usuari es descarrega el JSON resultant) i per últim només els usuaris que tenen el rol d'administrador poden eliminar les plantilles. Fora de la taula hi ha l'opció d'afegir una nova plantilla dins de la taula.

3.5.2.2 Modal de filtrar plantilles

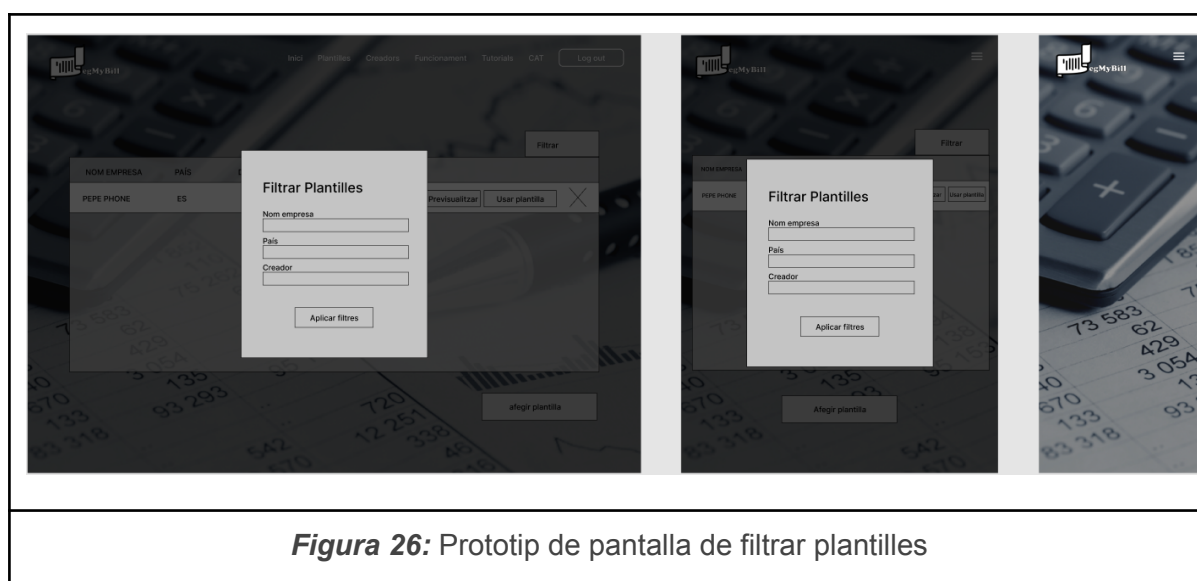
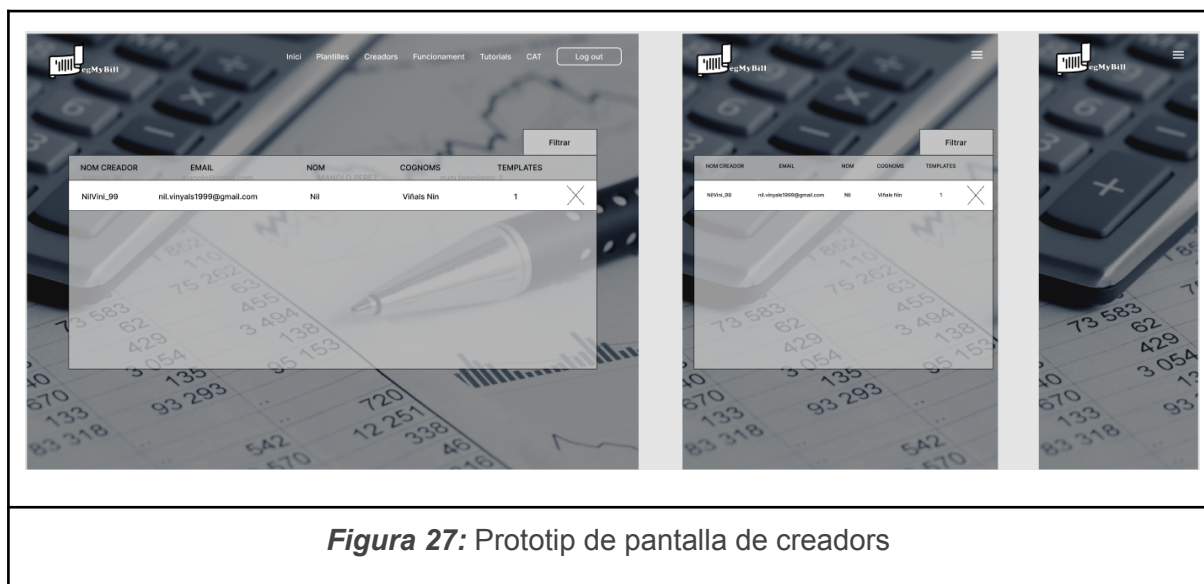


Figura 26: Prototip de pantalla de filtrar plantilles

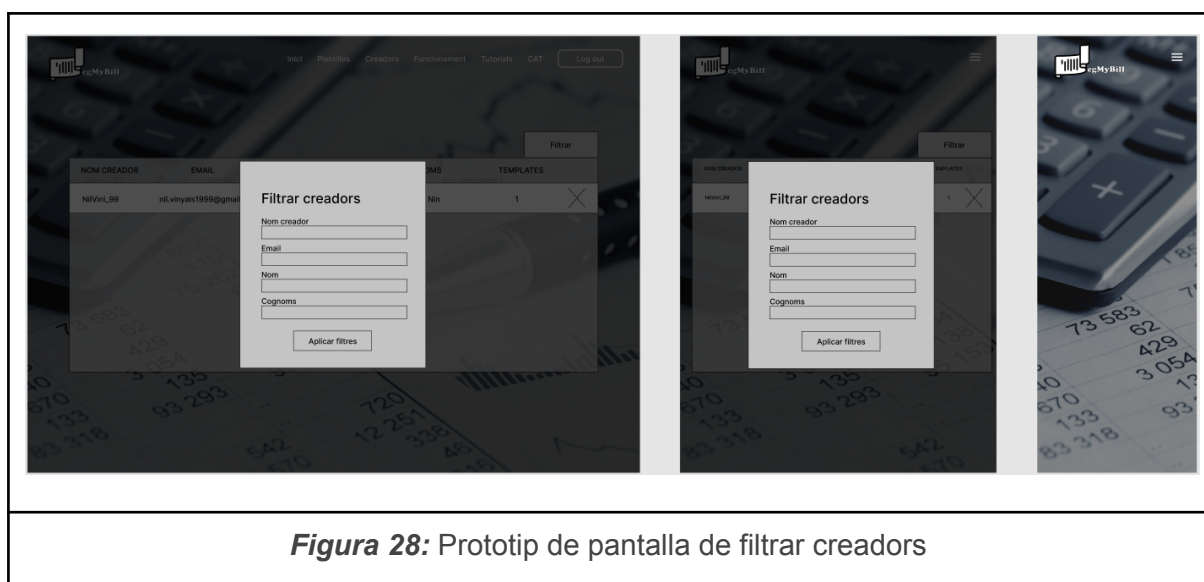
A la **Figura 26** es pot observar el menú de filtratge de resultats de la taula de plantilles.

3.5.2.3 Pàgina de creadors



A la **Figura 27** és la pantalla a on els usuaris amb rol d'administració poden veure tots els usuaris registrats de la web dins d'una taula. En aquesta taula es poden observar els diferents camps més importants dels usuaris i els administradors tenen la possibilitat d'eliminar un o diversos usuaris en concret.

3.5.2.4 Modal de filtrar creadors



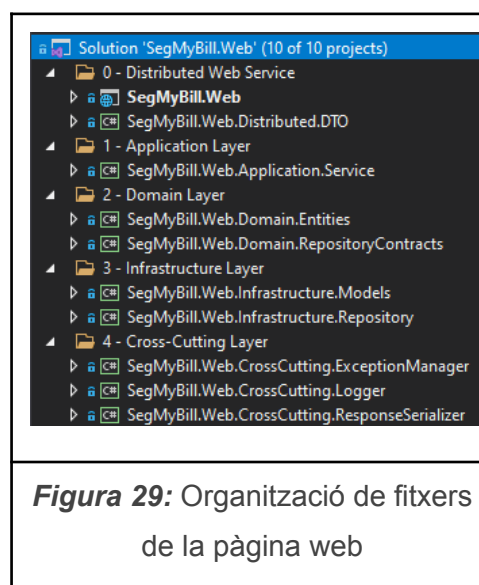
A la **Figura 28** es pot observar el menú de filtratge de resultats de la taula d'usuaris.

4 Implementació

4.1 Esquema organitzatiu dels fitxers

4.1.1 Esquema organitzatiu dels fitxers de la web

Seguint el disseny comentat en apartats anteriors, he aplicat una arquitectura DDD a on es poden trobar carpetes que marquen la capa a on està cada projecte, les solucions que són de la mateixa carpeta tenen els mateixos permisos d'acció entre capes.



Aquesta solució conté dos tipus de projectes, una Web App, que és SegMyBill.Web dins de la capa de Distributed Web Services i les altres són biblioteques de classes que donen els recursos a aquesta última i cada una especialitzada amb un tipus d'acció.

SegMyBill.Web: S'encarrega de la connexió entre el frontend de la pàgina web i el backend, també en ser el projecte d'execució s'encarrega de carregar totes les configuracions del programari i instanciar les seves dependències.

- Controllers: Generen crides possibles d'accedir a través d'un endpoint amb la configuració correcta i són capaços de retornar informació.
- ClientApp: En aquesta carpeta hi ha contingut tota la part del frontend i es pot trobar tota la lògica i configuració d'angular, les pàgines de maquetació i d'estil.

SegMyBill.Web.Distributed.DTO: S'encarrega de definir els objectes que entren i surten dels controladors anomenats DTO i la configuració del seu mapping.

SegMyBill.Web.Application.Service: S'encarrega d'aplicar la capa de lògica de negoci i les seves configuracions d'injecció de dependències i mapping.

SegMyBill.Web.Domain.Entities: Defineix els objectes que s'usen a la capa d'aplicació.

SegMyBill.Web.RepositoryContracts: Defineixen les interfícies a aplicar a la capa d'infraestructura, en concret al projecte de Repository.

SegMyBill.Web.Infrastructure.Models: Defineix els objectes que posteriorment seran serialitzats en la base de dades i també conté tota la configuració respecte a la persistència de dades.

SegMyBill.Web.Infrastructure.Repository: S'encarrega de la lògica per fer les crides a la base de dades.

SegMyBill.Web.CrossCutting.ExceptionManager: S'encarrega de fer el control d'excepcions que poden sorgir sobtadament, gràcies al fet que es passa per injecció de dependències fa que quan succeeix una excepció no caigui el programa.

SegMyBill.Web.CrossCutting.Logger: Conté la configuració del logger fent servir una llibreria externa anomenada Serilog, en la injecció de dependències instanciaré aquesta classe de dins d'aquest projecte com un singleton per tenir una sola instància en tota l'aplicació.

SegMyBill.Web.CrossCutting.ResponseSerializer: Té un petit serialitzador a json de missatges de resposta pels controladors.

4.1.2 Esquema organitzatiu dels fitxers de la Api de segmentació

Aquesta solució es basa d'un script envoltat per una web api feta amb Flask.

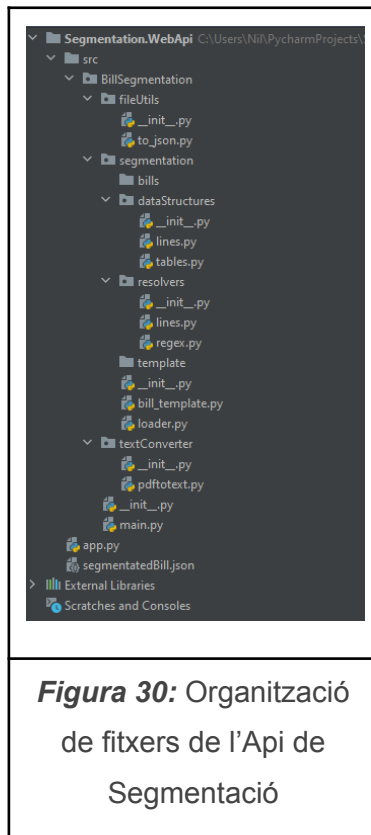


Figura 30: Organització de fitxers de l'Api de Segmentació

app.py: Representa la capa de la web Api a on ens esperem els fitxers que ens han de passar per paràmetres un cop els tenim els guardem dins d'unes carpetes anomenades bills i template, a partir d'allà executem el script passant-li com argument el camí al PDF passat anteriorment.

fileUtils: A dins d'aquesta carpeta hi ha la classe que defineix el mètode per sobreescriure l'output del script a segmentedBill.json

segmentation: Dins d'aquesta carpeta es troba el nucli de la serialització de la plantilla i l'aplicació de patrons segons el que hem escrit a la plantilla dins del text de la factura.

textConverter: A dins d'aquesta carpeta es troba la classe que defineix la funció que ens deixa extreure tot el text d'un PDF i les seves normes pel fet que s'usa una llibreria externa instal·lada en l'entorn de desenvolupament.

4.2 La gestió de la base de dades

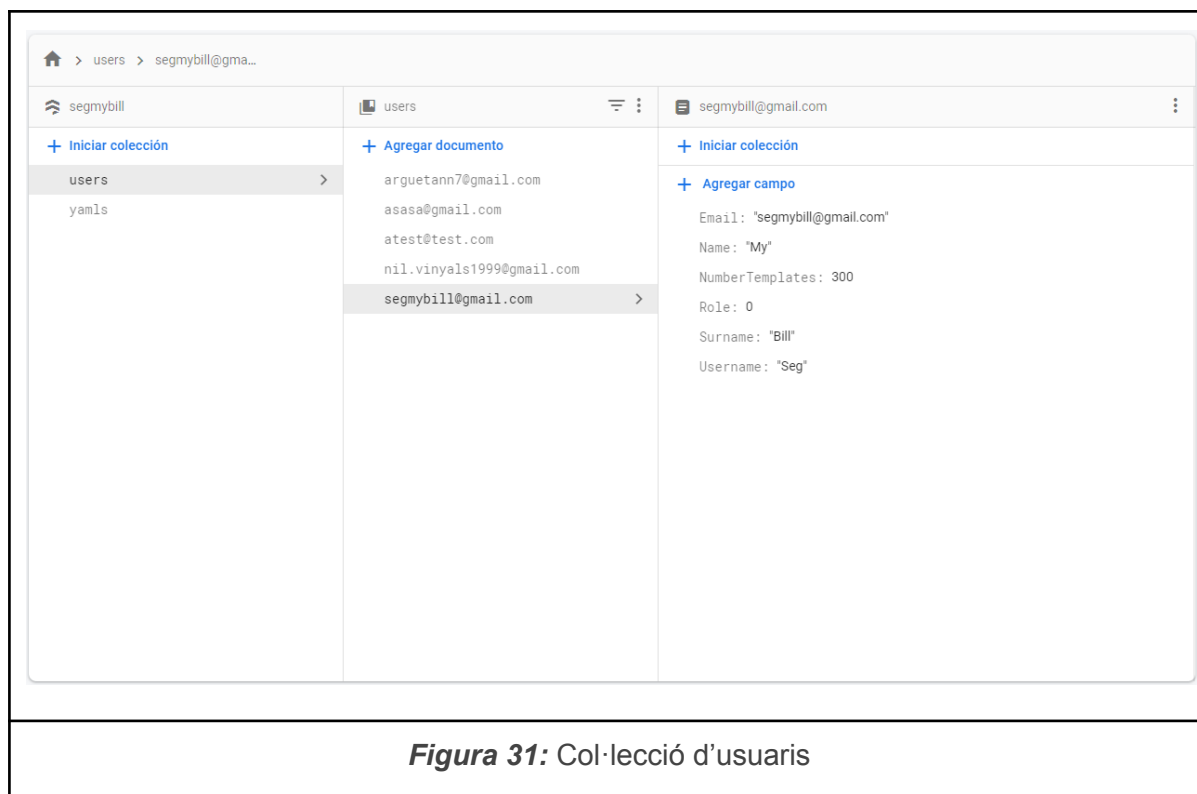
El projecte s'ha aplicat amb Firebase que és una aplicació de base de dades al Cloud especialitzat per aplicacions web i mòbils. Els serveis seleccionats de Firebase han sigut Firebase Authentication, que s'encarrega de la creació d'usuaris i els seus permisos i Firestore Database, que és una base de dades no relacional, de tipus document, a on tenim dues col·leccions a on guardem la informació dels usuaris i les plantilles respectivament a on cada document dintre de la col·lecció significarà o un usuari més o una plantilla i dins d'aquests es troba la informació.

4.2.1 Firestore Database

Les modificacions de la base de dades s'han fet a través d'una llibreria Nuget creada per Google que dóna accés a la seva Api d'una forma més dinàmica, d'instància única i fàcil de configurar.

4.2.1.1 Usuaris

En el backend de la web s'han creat 4 endpoints que permeten crear un usuari, eliminar-lo, obtenir l'usuari amb el correu electrònic i obtenir tots els usuaris. Per poder convertir els objectes a la base de dades, el model d'aquests s'han fet totalment serializables per Firestore.



4.2.1.2 Plantilles

Pel control de les plantilles s'han creat 4 endpoints que permeten crear una plantilla, eliminar-la, obtenir la plantilla segons el seu nom i obtenir totes les plantilles. Per no guardar la totalitat del document dins de la base de dades, el contingut del YAML s'ha codificat en base64 perquè posteriorment pel seu ús es pugui utilitzar sense cap problema, fent estalviar així capacitat a la base de dades. Per poder convertir els objectes a la base de dades, el model d'aquests s'han fet totalment serelizables per Firestore.

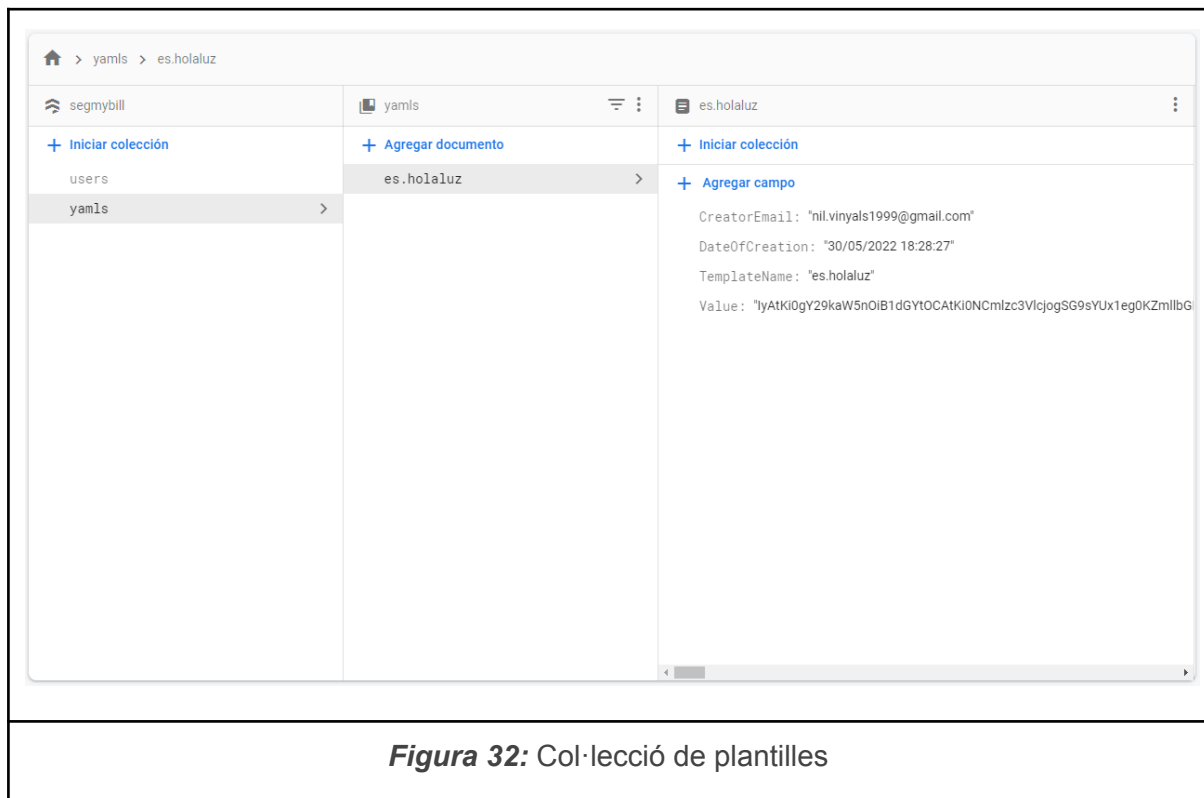


Figura 32: Col·lecció de plantilles

4.2.2 Autenticació

Per protegir les sol·licituds al backend de la web, he utilitzat JWT, aquest token només es pot obtenir a través de registrar-se o iniciar sessió.

La documentació de Firebase Auth no està preparada per C# per extreure el token de la sessió, llavors a través de la llibreria de Firebase a l'hora de crear una nova sessió, creo una sessió nova segons la contrasenya i el correu de l'usuari passat per la sol·licitud i a continuació per extreure el token, es fa una crida a la Api directa de Google i un cop filtrada la resposta extraiem el bearer token que requerim a totes les crides que requereixen inici de sessió. A l'hora d'iniciar sessió es fa la crida directament a la Api de Google

Identificador	Proveedores	Fecha de creación	↓	Fecha de acceso	UID de usuario
asasa@gmail.com		2 jun 2022			zJKwplIEOQQPD6R9JLkkJF0d2m2
atest@test.com		2 jun 2022			puM1Oary49M8ovLpDokaoq7Fwpl1
arguetann7@gmail.com		2 jun 2022			Sqa2375712MJn2Wnavjv7yZX9cw1
segmybill@gmail.com		31 may 2022		1 jun 2022	segmybill@gmail.com

Figura 33: Usuaris amb sessió creada

4.2.2.1. Autorització

A través d'un endpoint que només poden utilitzar tots els tokens autoritzats pel rol d'admin, podem generar noves sessions segons el rol de l'objecte user que li posem com a paràmetre a la crida mentre que sense el token amb el rol d'admin no serà capaç mai de crear una sessió amb rol d'administració.

Aquest desenvolupament s'ha generat a través d'un sistema de Claims, que vindrien a ser uns atributs de sessió a on sempre que fem una nova li assignem segons el rol que se li ofereix sempre que es tingui el permís.

Amb aquesta capa de seguretat els usuaris normals tenen impossibilitat d'accedir a sol·licituds sensibles.

4.3 Interfície

4.3.1 Adaptació a pantalles de diferents mides

La pàgina web s'ha dissenyat com a responsive, que significa que interpreta la grandària de la pantalla del dispositiu i s'ajusta a ella, a partir de la modificació dels seus estils i estructura segons la mida.



4.3.2 Actualització del contingut sense refrescar les pàgines

Els components dinàmics que es poden veure en la part frontend són les taules que es poden trobar a la pàgina d'usuaris i a la de les plantilles a on quan hi ha una modificació només es refresca el component fent que la càrrega que fa la memòria sigui molt més baixa i l'altre component és el de pujar arxius un cop seleccionat el fitxer o bé arrossegat, aquest mostra en la part d'avall del component el fitxer seleccionat encara que el modifiquem.

4.3.3 Permisos d'accés

Els perfils d'usuaris que tenim en el disseny realitzat en ordre de permisos de menor a major són: no registrats, usuaris normals i administradors.

El flux de la web comença a la pàgina d'inici, a on es pot distingir una barra de navegació que indica a quines pàgines pot accedir l'usuari i ens dóna l'opció d'iniciar o tancar sessió segons si està ja registrat.

Un cop registrat ja té capacitat de poder accedir dins de la pantalla de plantilles i utilitzar les seves funcions, però aquesta pantalla segons si tens el rol d'administració et deixa veure més informació per millorar la seguretat, la privacitat i no deixar a la vista informació personal sensible i permet esborrar plantilles de la base de dades.

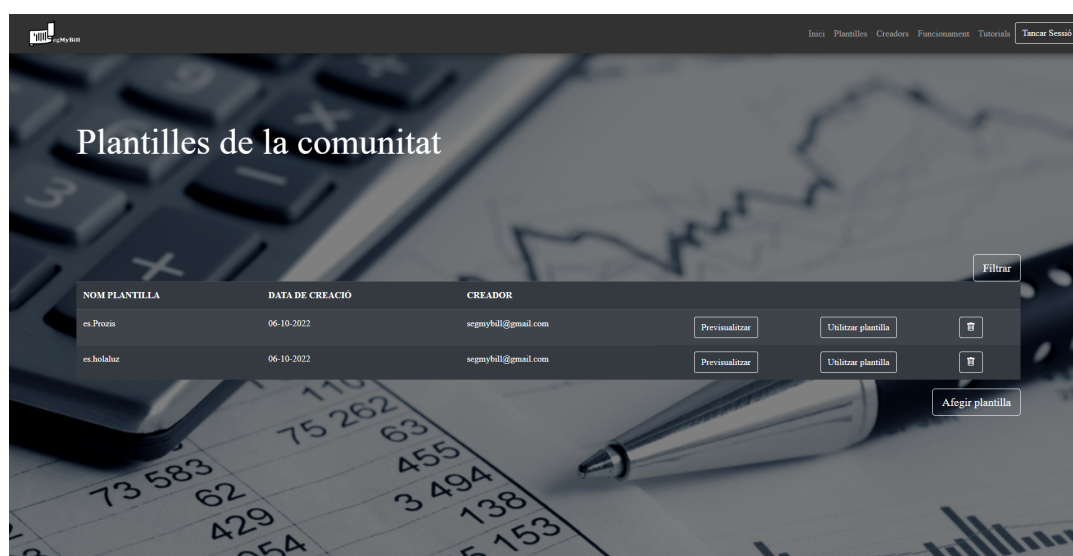
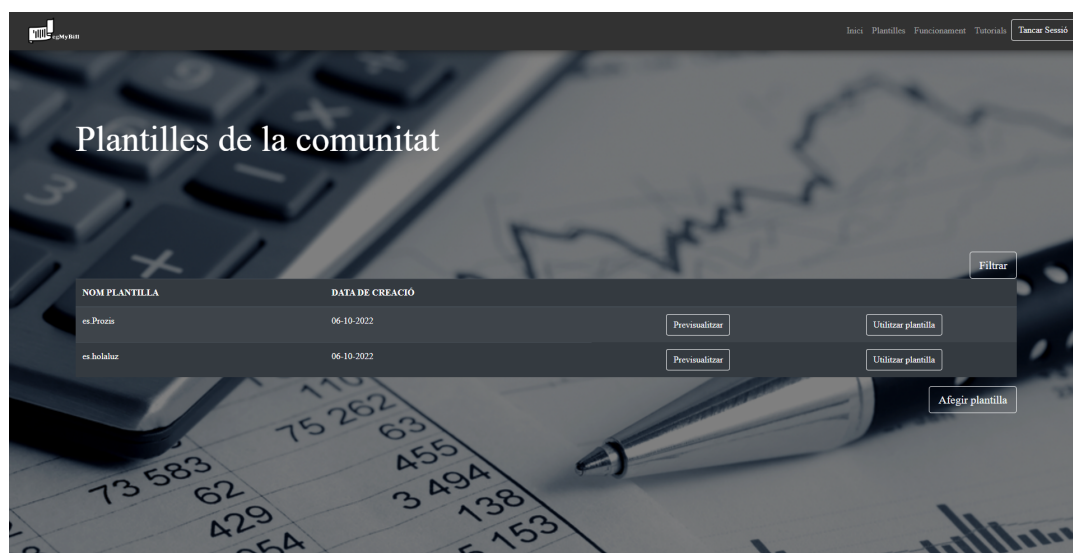


Figura 35: Visió d'un usuari normal i d'un admin de la pàgina de plantilles respectivament

L'usuari administrador també té la capacitat de gestió d'usuaris podent veure tots els que hi ha registrats i la seva informació dins de la pàgina d'usuaris i ser capaç d'eliminar usuaris.

4.3.4 Pantalles principals

La interfície gràfica de la web s'ha intentat adaptar el màxim possible als dissenys inicials i s'ha millorat la càrrega d'informació a la taula i altres aspectes.

4.3.4.1 Fer Login

Un cop que l'usuari està en la pàgina d'inici de sessió si mai s'ha registrat, donant-li al botó amb el text "Crear un compte" fa la redirecció en el formulari per fer el registre d'usuari.

En cas de ja tenir un usuari registrat hem de completar el formulari amb el correu i la contrasenya que li correspon i donant-li a l'enter o al botó amb el text "Entrar", amb aquesta interacció el front procedeix a fer una crida al controlador passant-li per paràmetre la informació passada pel formulari i rep un token i fa la redirecció a la pàgina de plantilles.

A partir que tenim el token ja podem fer totes les altres crides sempre que no estiguem restringits a fer-les dictats pels permisos d'accés.

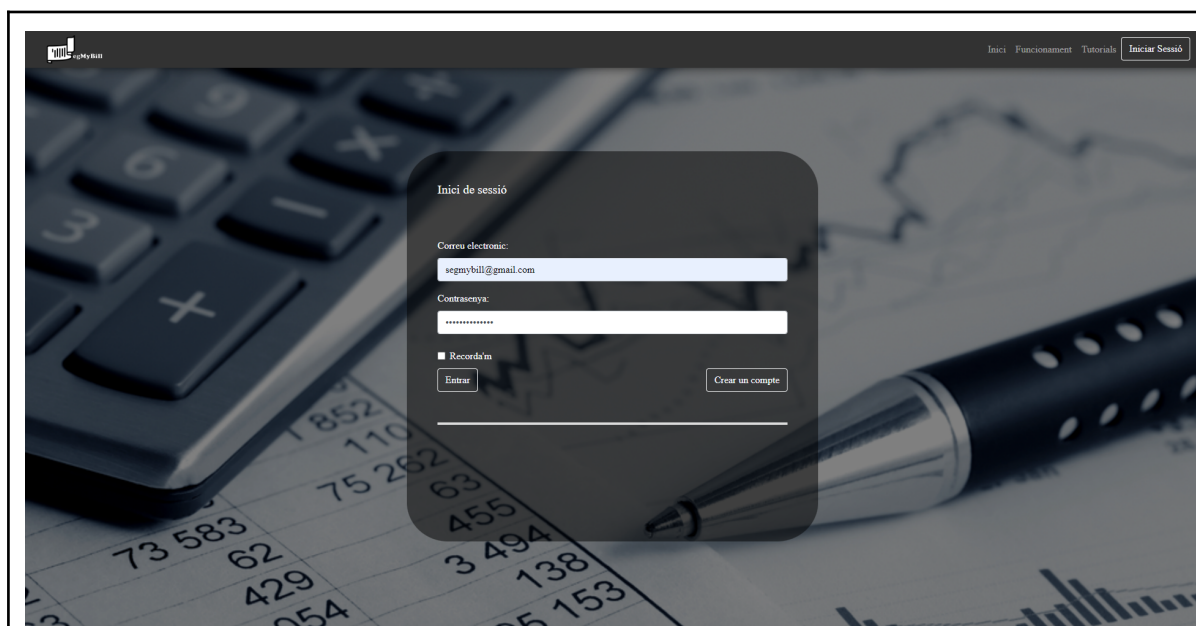


Figura 36: Pantalla d'inici de sessió

4.3.4.2 Fer Sign up

El procediment del signup té la mateixa finalitat que l'anterior, l'únic que aquest és necessari fer-ho el primer cop per registrar l'objecte amb les propietats que li indiquem al formulari i la possible sessió dins del Firebase, retornant així un token que ens permetrà accedir a les pàgines que necessiten autenticació. El funcionament es basa que el front s'encarrega de fer les comprovacions de dins de cada camp necessàries i un cop que es premi el botó amb el text "Crear un compte", el frontend fa una crida al controlador del back, passant-li un objecte usuari deserialitzat en un json dins del cos de la petició i el back s'encarrega de registrar-lo en la base de dades tant la sessió com l'objecte usuari.



Figura 37: Pantalla de registre d'usuari

4.3.4.3 Pantalla Plantilles

Dins de la pantalla de plantilles de la comunitat hi ha moltes funcions a fer, ja que aquesta és la principal de tota la pàgina web.



Figura 38: Pantalla de plantilles

4.3.4.3.1 Afegeir plantilla

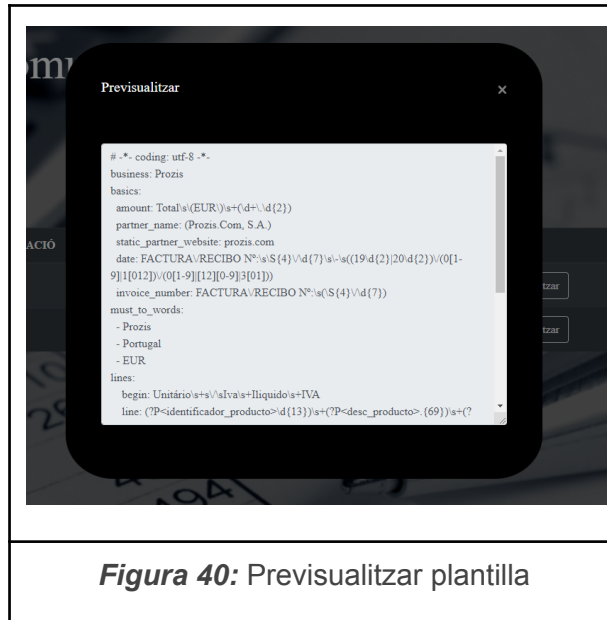
Quan es fa clic al botó afegir plantilla, s'obre una modal que conté al seu interior un component tipus filedrop es pot arrossegar un .yaml o bé fer clic sobre el botó busqueu el fitxer i s'obre un explorador d'arxius per seleccionar un fitxer .yaml, un cop que es faci clic en el botó Acceptar, procedirà a fer una crida en el backend per afegir la plantilla passant-li el document en el contingut de la crida.



Figura 39: Modal afegir plantilla

4.3.4.3.2 Previsualitzar plantilla

Quan es fa clic sobre el botó de previsualitzar, com que el valor de la plantilla el guardo en base64 per mantenir la informació de caràcters especials, s'extreu el valor de la plantilla segons la línia on està i a través de la funció atob de JavaScript es passa de base64 a UTF-8. Un cop es té el text descodificat es carrega una modal i en el seu interior un text-area amb el valor de la plantilla descodificat.



4.3.4.3.3 Eliminar plantilla

En fer clic a un dels botons amb la icona de la brossa, a partir del valor del nom de la plantilla en aquella fila es fa una crida al backend eliminant la plantilla i refrescant la taula.

4.3.4.3.4 Filtrar plantilla

En fer clic sobre el botó "Filtrar" sobre la taula, s'obre una modal amb dos camps per filtrar com es vulgui i quan es faci clic sobre el botó "Aplicar filtros" s'executarà una funció de JavaScript anomenada filter amb la condició de dins dels camps omplerts anteriorment, fent així que només es mostrin les plantilles que compleixin el filtratge.

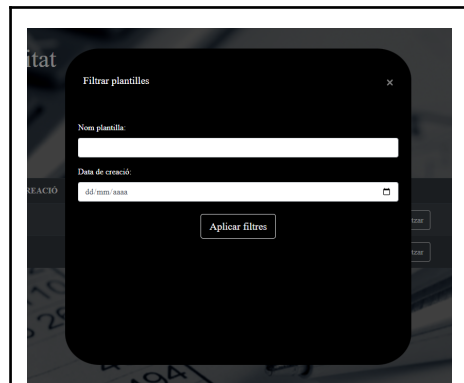


Figura 41: Filtrar plantilles

4.3.4.3.5 Utilitzar plantilla

En fer clic sobre el botó "Utilitzar plantilla" s'obre una modal que permet pujar un fitxer tipus PDF amb una factura, un cop es doni a acceptar, el frontend fa una crida al backend amb la informació de la plantilla de la fila a on hem clicat el botó i la factura pujada, i des del backend es fa la crida a l'api retornant aquesta un JSON amb la segmentació de la factura si tot ha anat correctament i aquest JSON es descarrega automàticament per l'usuari.

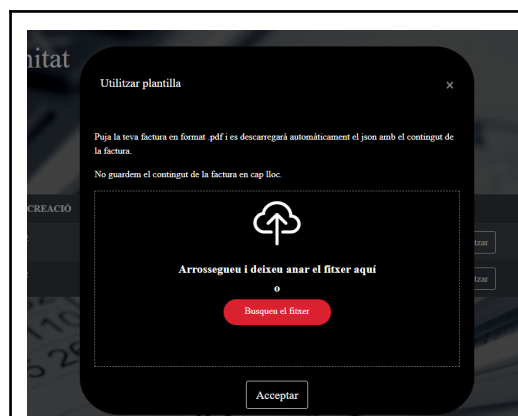
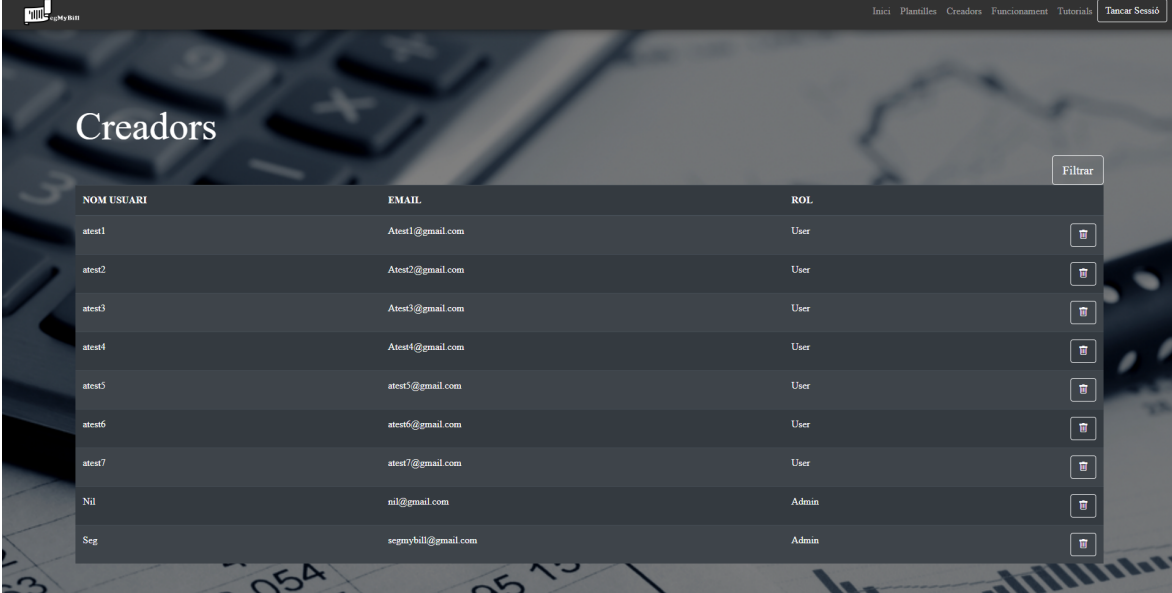


Figura 42: Pantalla d'ús de plantilla

4.3.4.4 Pantalla Users

Aquesta pantalla serveix perquè els administradors tinguin un lloc dins de la web per fer una mica de gestió d'usuaris, fora de les crides ja existents en el back de la web que només es poden utilitzar amb tokens que continguin el rol d'administrador.

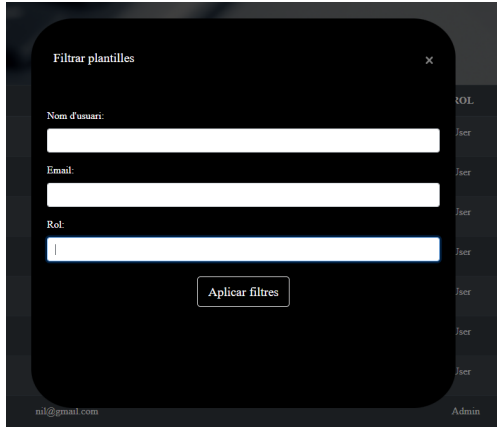


NOM USUARI	EMAIL	ROL
atest1	Atest1@gmail.com	User
atest2	Atest2@gmail.com	User
atest3	Atest3@gmail.com	User
atest4	Atest4@gmail.com	User
atest5	atest5@gmail.com	User
atest6	atest6@gmail.com	User
atest7	atest7@gmail.com	User
Nil	nil@gmail.com	Admin
Seg	segmybill@gmail.com	Admin

Figura 43: Pantalla d'usuaris

4.3.4.4.1 Filtrar usuarios

En fer clic sobre el botó "Filtrar" sobre la taula, s'obre una modal amb tres camps per filtrar com es vulgui i quan es faci clic sobre el botó "Aplicar filtros" s'executarà una funció de JavaScript anomenada filter amb la condició de dins dels camps omplerts anteriorment, fent així que només es mostrin els usuaris que compleixin el filtratge.



Filtrar plantilles

Nom d'usuari:

Email:

Rol:

Aplicar filtros

Figura 44: Filtrar usuarios

4.3.4.4.2 Eliminar usuaris

En fer clic a un dels botons amb la icona de la brossa, a partir del valor del correu de l'usuari en aquella fila es fa una crida al backend eliminant l'usuari i la seva sessió, acabant el procediment refrescant la taula.

4.4 Segmentació de factures

En aquesta part procedirem a anar amb ordre en el procés d'execució en la segmentació de factures.

4.4.1 Web Api de Segmentació

L'objectiu d'aquest de SegMyBill des de l'inici del treball és poder construir una comunitat a on tots els usuaris puguin formar part d'ella i deixar el seu gra de sorra i també donant un servei que pot ser necessari per a moltes persones amb finalitats molt diferents. Llavors un dels propòsits principals era no posar més barreres de les necessàries i fer que la gent personalitza el seu propi ús del servei, degut això s'ha dissenyat una API a on es fa la segmentació directament i és totalment pública.

La web Api s'ha creat amb Flask i conté 2 crides a on una, espera dos fitxers, una plantilla(.yml) i la factura(.pdf) que li correspon a la plantilla i l'altre crida la plantilla està l'objecte Template deserialitzat en un JSON passant el seu valor com està a la base de dades, és a dir codificat a base64.

Un cop que la crida rep les dades guarda els documents dins de les carpetes del script per fer després el seu ús i a continuació executem el script. Posteriorment, es passa com a resposta el JSON que ha estat sobreescrit pel script.

4.4.2 De pdf a text

Pel desenvolupament del script he necessitat diverses llibreries que em podrien facilitar la feina, a causa de la seva quantitat i per accelerar el desenvolupament, ja que dóna més flexibilitat i independència entre projectes vaig crear un entorn d'Anaconda amb totes les llibreries necessàries o per provar el rendiment.

Pel fet que la gran majoria de les factures s'envien en format PDF, una de les llibreries més importants per implementar la segmentació era poppler-utils que és una llibreria de lectura de PDFs, a partir d'aquesta llibreria s'utilitza pdftotext que és una comanda que va distribuïda a molt sistemes operatius base Linux per defecte que s'encarrega de passar el contingut llegit del PDF a un .txt o extreure el seu contingut a una cadena de text.

Un cop extret la informació de la factura en una cadena de caràcters és més fàcil trobar coincidències i els patrons.

Un cop que la plantilla estigui en la carpeta a on l'hem descarregat es farà la serialització de .yaml a un diccionari d'ordre a partir de la llibreria yaml, a on cada camp serà una clau i el contingut, el patró a trobar el valor de la clau en casos directes i en casos més complexos el valor del camp pot ser una llista o bé un altre diccionari.

Per descarregar la càrrega computacional s'ha creat un camp anomenat `must_to_words`, aquest camp sempre ha de formar part de la plantilla i s'ha de posar dins seu paraules que es puguin descobrir sempre en aquella factura en concret, així abans de fer tota la cerca de patrons es mira si dins de la factura hi ha aquelles paraules i si no ho estan, llavors ja no fa res més, indicant que no és una bona plantilla per la factura.

Figura 45: Plantilla serialitzada en un OrderDict

4.4.4 Funcionament del Regex

Com he dit anteriorment Regex és un llenguatge d'expressions regulars a on una sèrie de símbols signifiquen patrons, els símbols més usats són:

Caràcters:

- **Patró:** ABC
- **Text:** abcAaBCABCcABC

Per detectar el patró ha d'estar juntes les lletres ABC amb el mateix ordre.

Rangs:

- **Patró:** [A-C]
- **Text:** abcAaBCaBCcABC

Selecciona totes les lletres que es troben entre A i C, que són A,B i C, aquest patró també és aplicable amb números.

Unió caràcters i rangs:

- **Patró:** 1[0-2]
- **Text:** 1012 31 1 102

L'1 és obligatori a l'esquerra i totes les combinacions estan acceptades del rang amb un 1 a l'esquerra.

Patrons variables:

\d → un dígit	\s → un espai	\w → una lletra	. → qualsevol menys salts de línia
\D → no un dígit	\S → no un espai	\W → no una lletra	

[s\S] → Posar aquestes claus al voltant de patrons variables significa que es compleixin els patrons de l'interior, en el cas de \s\S, selecciona tot el text.

En cas que vulguem fer que ens sumi en el patró un caràcter que no sigui una lletra o número només hem de posar \ i a continuació el caràcter especial, per exemple: \€, \., \@ ...

Modificador de llargada de patró:

- **Patró:** A\s+
- **Text:** Aaaaaa aaA aA a

El signe + després d'un patró significa que no té una llargada establerta, sinó va fins a on pugui arribar complint el patró, en canvi, podem fixar la llargada si li posem {}:

- **Patró:** A\s{3}
- **Text:** Aaaaaa aaA aA a

OR:

El signe | es un condicional de patrons (**patró_1 | patró_2**)

Si es combina tot lo nombrat en aquest apartat podem crear coses com:

- **Patró:** ((([1-9]||[12][0-9])3[01])\V([1-9]|1[012])\V(19\d{2}|20\d{2}))
- **Text:** Dia de facturació: 29/10/1992

Hi ha una pàgina web que dóna tota la informació de com va i pots aprendre molt amb els teus propis exemples anomenada, també ajuda molt en crear patrons per les factures:

<https://regexr.com>

Per aplicar el regex dins del script generalment ha sigut a partir d'una sèrie de funcions de cerca de coincidències de la llibreria de regex anomenada re, passant per paràmetre el patró i el text a analitzar (un tros de la factura passada a string o tota la factura feta string) i això retorna un objecte Match a on estan els valors amb les coincidències.

4.4.5 Estructura de dades dins de les plantilles

Segons la documentació de poppler i de pdftotext, el text conservat a dins de taules es tracta de la mateixa forma que text pla, això genera una sèrie de problemes, ja que no totes les factures apliquen la informació en la mateixa línia, sinó com si estigués continguda dins d'una taula i el valor de l'atribut estigués sota d'aquest.

Per solucionar aquest cas s'ha creat un camp específic anomenat tables, que en fer l'extracció de dades té la seva forma concreta i les seves normes per extreure la informació.

Per utilitzar les taules s'han de posar tres paràmetres, que són:

- Begin: Una referència propera que estigui abans de la dada que es vol extreure.
- Finish: Una referència propera que estigui després de la dada que es vol extreure.
- Content: La variable la que volem extreure i el seu patró.

Begin i finish és aconsellable que sigui text pla no variable, si no s'ha d'apuntar el patró que pot complir la variable.

Exemple:

Fecha de emisión: 23/3/2022	Contrato 2022056081
Figura 46: Exemple de taules	

begin: Fecha de emisión:\s+Contrato

finish: Datos del cliente

content: (?P<date>(?:[1-9][12][0-9][301])\V(?:[1-9][1012])\V(?:19\d{2}|20\d{2})))\s+(?P<contrato>(\d{10}))

El que fa el script és buscar els patrons de lletres que li hem passat com a begin i finish i retalla la part que es troba enmig d'aquests dos punts a la factura, un cop tret separa les línies i per cada línia fa la cerca del patró que hi ha dins del content, un cop es trobi li assignarà un camp amb el nom de la variable que li hem passat (?P<nom_variable>) i el valor que s'ha tret al trobar el patró. Ja que la informació principal de la factura pot estar en

aquest format en cas que la variable contingui en el nom amount o date es fa una conversió de tipus de string a float o datetime depenent del cas.

També s'ha de tenir en compte un altre tipus d'estructura molt comuna dins de les factures que són les línies, tal com he comentat en apartats anteriors.

Com que vull crear plantilles que siguin factibles per totes les factures del mateix tipus, s'ha de tenir en compte que no totes les factures contenen el mateix nombre de files, així que la quantitat de línies és una variable, així que s'ha de gestionar com un cas diferent més.

Les línies contenen 3 paràmetres obligatoris:

- Begin: Una referència propera que estigui abans de la dada que es vol extreure.
- Finish: Una referència propera que estigui després de totes les dades que es volen extreure.
- Line: Si no hi ha first_line significarà que aquesta línia és l'única a seguir el patró.

i també té 3 opcionals per casos en concret:

- first_line: Marcar la línia d'inici a extreure dades.
- last_line: Marcar l'última línia a extreure les dades.
- skip_line: No extreure les seves dades.

Exemple:

Concepto	Base	Precio unitario	Importe
Término fijo	4 días	0,088476 €/día	0,35 €
Término de energía	50,83 kWh	0,177 €/kWh	9,00 €
Impuesto especial sobre hidrocarburos	50,83 kWh	0,00234 €/kWh	0,12 €
Alquiler del contador	4 días	0,020055 €/día	0,08 €
IVA	9,55 €	21%	2,01 €
Total factura			11,56 €

Figura 47: Exemple de línies

begin: Concepto\s+Base\s+Precio unitario

line: (?P<concepto>\w+.*)\s+(?P<base>\S+\s+(\S+|\€))\s+(?P<precio_unitario>\d+(\,|\d+|))+
(\s{1}|\€\s\(\S+|\%))\s+(?P<importe>\d+(\,|\d+))\s\€

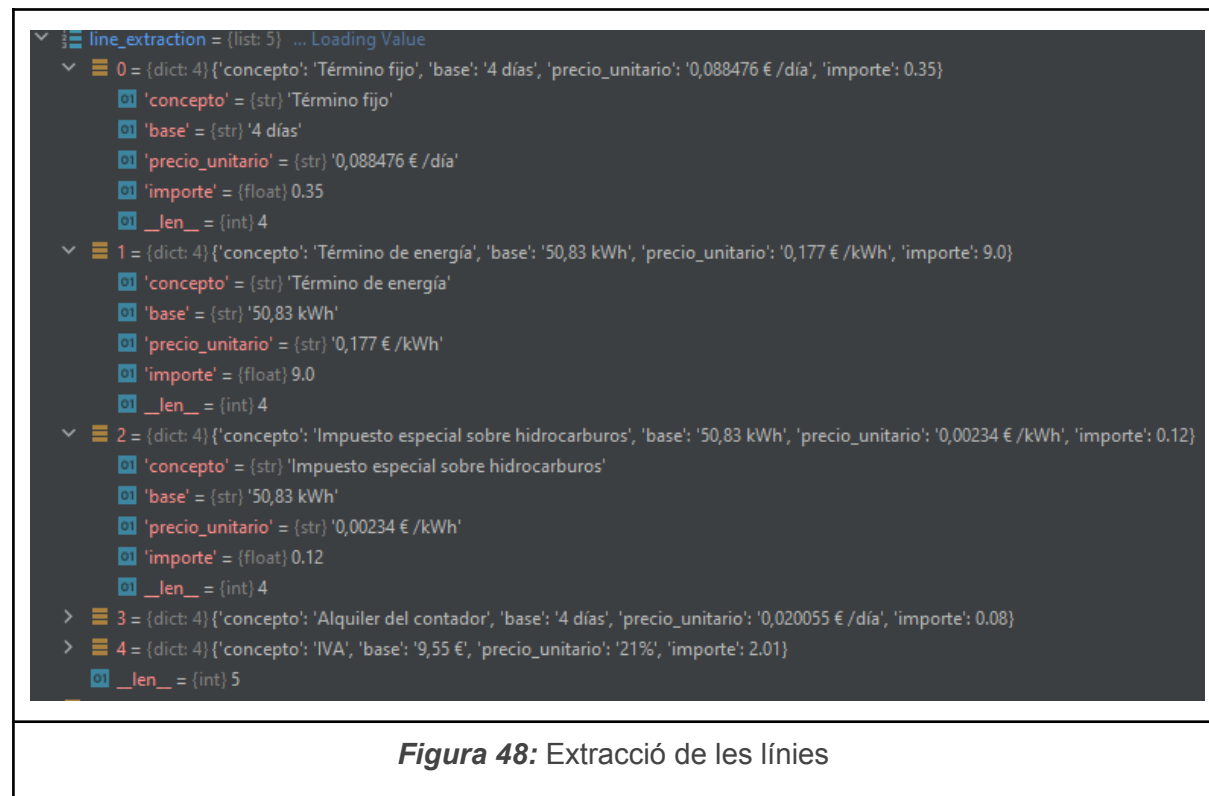
finish: Tus lecturas

types:

importe: float

Amb types tenim la capacitat d'assignar-li a una variable el seu tipus en concret.

Les línies comencen amb el mateix procediment que les taules, és a dir s'agafa el begin i el finish obtenint la part que queda entre els dos punts. Per cada línia que passa intenta buscar el patró en el seu interior, en cas de trobar-lo i que no sigui la línia que s'ha de saltar, guarda tot l'aconseguit dins d'un diccionari i afegeix aquest a una llista.



4.4.6 Extracció a JSON

A partir de les extraccions de les dades de tot el sistema es van afegint dins d'una llista, a on posteriorment a través d'un json dump es passa a format JSON dins d'un fitxer, això és fàcil, ja que el que s'ha extret són col·leccions dins d'altres amb valors primitius al seu interior.

```
{
  "amount": 11.56,
  "business": "HolaLuz",
  "contrato": "2022056081",
  "currency": "EUR",
  "date": "23-03-2022",
  "description": "Bill from HolaLuz",
  "invoice_number": "2022G00280361",
  "lines": [
    {
      "base": "4 días",
      "concepto": "Término fijo",
      "importe": 0.35,
      "precio_unitario": "0,088476 € /día"
    },
    {
      "base": "50,83 kWh",
      "concepto": "Término de energía",
      "importe": 9,
      "precio_unitario": "0,177 € /kWh"
    },
    {
      "base": "50,83 kWh",
      "concepto": "Impuesto especial sobre hidrocarburos",
      "importe": 0.12,
      "precio_unitario": "0,00234 € /kWh"
    },
    {
      "base": "4 días",
      "concepto": "Alquiler del contador",
      "importe": 0.08,
      "precio_unitario": "0,020055 € /día"
    },
    {
      "base": "9,55 €",
      "concepto": "IVA",
      "importe": 2.01,
      "precio_unitario": "21%"
    }
  ],
  "partner_name": "HOLALUZ-CLIDOM, S.A.",
  "partner_website": "holaluz.com"
}
```

Figura 49: Json final

5 Conclusions

Considero que els objectius d'aquest projecte s'han assolit segons la llista de requisits mencionada a la introducció i seguint el disseny i anàlisi premeditats anteriors a la implementació de la forma més fidel possible, a causa d'això diverses implementacions que haurien d'haver sigut molt més costoses han sigut molt més lleugeres i automàtiques.

Gràcies als coneixements adquirits en el grau i a l'experiència en desenvolupament web que he anat adquirint a base que aquest projecte ha avançat i la solució d'errors en el procés ha fet que estigués realment orgullós del resultat definitiu.

També es pot considerar que s'han completat els objectius generals del projecte, ja que la web ofereix el servei de segmentació de factures segons plantilles d'altres usuaris, fent així que es pugui crear una comunitat al voltant de la web sent més fàcil l'obtenció de dades de factures.

Per altra banda, s'ha realitzat una web facilitant a l'usuari les tasques més pesades, el cas de l'inici de sessió, registre d'usuari, observar les plantilles i més. A part de fer una navegació per la web més ràpida i intuïtiva gràcies a la implementació d'Angular que permet crear Single Page Applications, donant als usuaris una experiència molt més fluida de la web.

Un altre aspecte a destacar encara que no formi part de les característiques amb funcionalitat és la seguretat aplicada amb tokens protegint tant com el backend com el frontend de peticions anònimes i també el nivell d'autorització usuari-administrador, que protegeix les crides més sensibles del backend i permet limitar la vista en el frontend dels usuaris normals.

5.1 Futures ampliacions

Una bona ampliació he volgut implementar com un extra, però a partir de la llista de prioritats del projecte s'ha quedat fora, ja que suposava fer una ampliació del script i que no hi hagués cap problema amb la comunicació entre backend i Api, causant així diversos punts de risc. Aquesta característica era passar el nombre de factures que es vulgui aplicar una plantilla en concret i que el JSON resultant contingui una llista de totes les factures amb les seves dades desglossades. Això causaria que hi hagués menys crides a l'Api i fos factible pujar més d'un document des de la pàgina web.

Una altra futura millora seria contenir les solucions dins de contenidors Docker amb totes les dependències necessàries per poder un deploy de la web i l'Api més net possible a una plataforma de serveis al Cloud.

I per últim, una altra millora per la web seria que en crear una plantilla poder-la editar en la mateixa web i amb el text de la factura d'exemple subratllant-se a on anés detectant els patrons detectats, semblant a com es pot veure a la pàgina de regex: <https://regexr.com/>, això faria que hi hagués una taxa de fiabilitat de les factures més gran i que en tenir una resposta immediata de la teva plantilla fa que l'aprenentatge es faci d'una forma òptima.

Glossari

Backend: És la part del software que processa la informació obtenida o sol·licitada pel frontend.

Frontend: És la part del software que interactua amb els usuaris.

Web Api: Api ve de application programming interface, és un conjunt de subrutines, funcions i procediment que ofereixen certa biblioteca per ser utilitzada per un altre software.

Framework: És un entorn de treball que permet desenvolupar software de manera més senzilla, pot incloure llibreries, suport de programes i llenguatge interpretat per ajudar a unir les diferents components d'un projecte.

Angular: És un framework per aplicació de desenvolupament en TypeScript, de codi obert, mantingut per Google, que s'utilitza per crear i mantenir aplicacions web d'una sola pàgina.

TypeScript: És un llenguatge de programació lliure i de codi obert desenvolupat i mantingut per Microsoft. És un superconjunt de JavaScript, que essencialment afegeix tipat estàtic i objectes basats en classes.

Token JWT: És una cadena de caràcters que permet la propagació d'identitat i privilegis, poden ser utilitzats per propagar la identitat dels usuaris com a part del procés d'autenticació entre el proveïdor d'identitat i un proveïdor de serveis.

Deploy: Significa aixecar el servei en un entorn de producció, fent que tothom pugui accedir a ell a través d'internet.

Cloud: És l'ús d'una xarxa de servidors remots connectats a internet per emmagatzemar, administrar i processar dades, servidors, bases de dades, xarxes i software. En lloc de dependre d'un servei físic instal·lat, es té accés a una estructura a on el software i el hardware estan virtualment integrats.

Anaconda environment: És un directori que conté una col·lecció específica de paquets conda que has instal·lat.

Bibliografia

Google. (2010). Angular Cli, A command line interface for Angular. [en línia]. Ús: 10/5/22 <https://cli.angular.io/>

Google. (2010). What is Angular?. [en línia]. Ús: 10/5/22 <https://angular.io/docs>

Google. (2018) Firebase Authentication documentation [en línia] Ús: 2/5/22 <https://firebase.google.com/docs/auth>

Google. (2020) Google Cloud Firestore documentation [en línia] Ús: 30/4/22 <https://cloud.google.com/firestore/docs/how-to>

Google. (2021) Firebase Firestore documentation [en línia] Ús: 30/4/22 <https://firebase.google.com/docs/firestore>

Microsoft. (2021) Dependency injection in ASP.NET Core [en línia] Ús: 22/4/22 <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-5.0>

Microsoft. (2021) Create web APIs with ASP.NET Core [en línia] Ús: 22/4/22 <https://docs.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-5.0>

Microsoft. (2021) Common web application architectures [en línia] Ús: 23/4/22 <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>

Pallets. (2018) Flask User's Guide [en línia] Ús: 19/4/22 <https://flask.palletsprojects.com/en/2.1.x/>

Gskinner (2020) RegExr: Learn, Build, & Test RegEx [en línia] Ús: 15/2/22 <https://regexr.com/>

freedesktop.org(2005) Poppler info and documentation [en línia] Ús: 11/3/22 <https://poppler.freedesktop.org/>

Jason Alan Palmer (2018) Pdftotext git repository [en línia] Ús: 11/3/22 <https://github.com/jalan/pdftotext>

Stephen Watts (2020) The Importance of SOLID Design Principles [en línia] Ús: 29/4/22 <https://www.bmc.com/blogs/solid-design-principles/>

Bartłomiej Iskrzycki (2021) Sending files and additional data using HttpClient in .NET Core [en línia] Ús: 20/5/22 <https://brokul.dev/sending-files-and-additional-data-using-httpclient-in-net-core>

geekflare (2021) Introducció a YAML en Python para principiantes [en línia] Ús: 17/2/22
<https://geekflare.com/es/python-yaml-intro/>

RealPython (2021) OrderedDict vs dict in Python: The Right Tool for the Job [en línia] Ús: 17/2/22
<https://realpython.com/python-ordereddict/>

Jeffrey Friedl (2009) Regular expression operations [en línia] Ús: 15/2/22
<https://docs.python.org/3/library/re.html>