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

Valls Pineda, Jordi; Serrat Gual, Joan, dir. Registre d'activitat en Cloud Provider. 2022. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/264200>

under the terms of the  license

Registre d'activitat en Cloud Provider

Jordi Valls Pineda

Resum– Tota empresa que tracta amb residus ha de tenir una regulació del material d'entrada i sortida per complir amb la llei vigent. En aquest treball es desenvoluparà una aplicació destinada a poder registrar aquests residus d'una manera senzilla i eficaç, poder accedir-hi i visualitzar-los intuïtivament, i poder extreure'n informes adaptats al client a qui va destinat el sistema.

Paraules clau– Registre, ACR, Python, Framework, Django, Google Cloud, Prototipatge, Testing.

Abstract– Every company that deals with waste must have a regulation of the input and output material to comply with current law. In this work, an application will be developed to be able to register this waste in a simple and efficient way, to be able to access and visualize it intuitively, and to be able to extract reports adapted to the client for whom the system is intended.

Keywords– Register, ACR, Python, Framework, Django, Google Cloud, Prototyping, Testing.

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

AQUEST projecte consisteix en el disseny i implementació d'una aplicació de gestió de residus per a una empresa de l'àmbit del reciclatge. En empreses on es reben i es processen residus, ja siguin orgànics, comercials, industrials o sanitaris, s'ha de mantenir un registre dels residus tractats. En el cas dels residus orgànics, per exemple restes vegetals procedents de parcs, jardins o d'altres, s'han de dipositar en una planta de reciclatge especialitzada on es tractin aquests residus i se'n ha de portar un control.

L'aplicació construïda respon a unes necessitats reals: disposar d'un registre centralitzat de tots els residus rebuts per part d'una planta. Com veurem, i degut a certes raons, aquesta problemàtica ens ha dut a desenvolupar una aplicació client-servidor que té una interfície web i que emmagatzema les dades al núvol.

El registre d'entrades dels residus rebuts en una planta de reciclatge és obligatori, ja que de cara a una inspecció de l'Agència Catalana de Residus (ACR) s'ha de disposar d'aquest [1]. També és obligatori introduir anualment el tonatge total de residus tractats [2] a la plataforma de l'ACR, per saber el nivell d'estocatge total a planta. Per tal de poder calcular aquest total, també s'hauran de poder introduir al sistema les sortides de la planta.

Sense voler entrar en detalls per ara, i només a efectes d'exemple, els camps del formulari d'una entrada, entre d'altres, són la data, el nom del productor i del transportista i tipus de residu. Els residus podran ser dels següents tipus: teixit vegetal o residus biodegradables. La varietat de residus a poder entrar variarà segons la planta de reciclatge i els permisos atorgats. En cas que ens ocuparà, i que correspon a desenvolupar la aplicació per a una planta de reciclatge vegetal anomenada Ecotrans Vallès, són els dos mencionats anteriorment.

Els camps del formulari d'una sortida, entre d'altres, són la data, el nom del productor, transportista i destinatari, la quantitat i tipus de material. En aquest cas el tipus de residu de sortida no està regulat, ja que dependrà del tractat que s'hagi fet d'aquest. En el nostre cas en particular, poden ser de tres tipus: pre-triturat de poda, triturat de poda fi

-
- E-mail de contacte: 1494021@uab.cat
 - Menció realitzada: Enginyeria del Software
 - Treball tutoritzat per: Joan Serrat Gual (Computer Vision Center)
 - Curs 2021/22

o triturat de poda cribrat. Aquestes entrades i sortides seran introduïdes directament per l'operari en planta un cop arribi un camió. L'operativa actual consisteix en que, en cas de que no es puguin introduir certes entrades en el mateix moment, es fa servir l'albarà en paper que porta el transportista per introduir les dades posteriorment, al despatx per part de l'administrador. Per tant, la freqüència d'entrada d'aquestes dades dependrà del context, podent ser al moment o al cap d'uns dies.

Actualment aquest registre es realitza manualment, fet que impedeix l'automatització del procés i l'extracció de mètriques relacionades amb la quantitat de tonatge tractat. Així doncs, la motivació principal del projecte és aportar una solució que automatitzi i centralitzi les dades entrades en el sistema, possibilitant el processament de dades i la visualització d'aquestes, cosa que fins ara no és possible ja que s'introdueixen a mà en una full. Encara si només s'hagués de realitzar el registre de les entrades i sortides, es podria utilitzar un simple full de càlcul. Però en haver de calcular mètriques, totals, llistats i fer consultes per després generar informes per a la ACR això no és possible i és necessària una aplicació com la que ens hem proposat desenvolupar.

En aquesta direcció, una de les funcionalitats que hem incorporat és la visualització de les dades més importants en una *dashboard*. També, al moment d'introduir-les, agilitzar el procés mitjançant per exemple camps que s'autocompleten, molt important pel client a qui va destinada l'aplicació per agilitzar el procés d'inserció de dades, com ara obtenir les dades d'un client (una empresa que porti residus a la planta, com un jardiner) només posant les primeres lletres. Per últim, una altra de les necessitats que hem detectat és la possibilitat de fer un seguiment visual i intuïtiu del ritme d'entrades i sortides d'un client, ja sigui mensual o anualment.

2 OBJECTIUS

Com ja hem dit, l'objectiu principal del projecte és implementar una aplicació web per registrar entrades i sortides de residus vegetals en una planta de reciclatge vegetal. També ha de permetre als usuaris visualitzar el tonatge total de residus d'entrada i sortida a planta així com poder filtrar aquesta informació donat un rang de dades concret per poder realitzar els informes pertinents.

Els objectius principals són:

- Poder introduir les dades d'entrada i sortida.
- Poder visualitzar quin és el nivell de *stock* en el moment (comparació entre tones d'entrada envers sortida).
- Cada client (per exemple d'un jardiner o ajuntament) pot tenir més d'una matrícula associada. Poder seleccionar d'entre aquestes.
- Poder modificar les dades dels clients, entrades i sortides en cas de que s'hagi introduït alguna dada erròniament.
- Poder seleccionar quin és el tipus de residu rebut i el tipus de material de sortida.

- Poder exportar les dades.

Per altra banda, podem trobar els següents objectius secundaris, derivats dels principals. Tot i que tenen importància dins la presa de requisits amb el client, tenen un grau de rellevància i prioritat més baix:

- Dissenyar una interfície fàcil d'utilitzar, sent intuïtiva i agradable pels usuaris.
- Implementar l'aplicació per poder ser accedida remotament, pensant des del primer moment en la implementació en el núvol.
- Incloure un dashboard on poder visualitzar les dades filtrades per client mitjançant gràfiques i per tant poder determinar l'evolució d'un client segons el nombre de tonatge dipositat cronològicament.
- Aprofitant que ja es disposa d'usuaris reals, realitzar test d'usabilitat de la interfície i determinar el guany en temps respecte al mètode utilitzat actualment.

Com que les entrades s'han de poder realitzar tant des de planta com des de despatx, és interessant que es puguï accedir a l'aplicació remotament, i per tant s'encararà el desenvolupament del sistema per ser implementat en el núvol. Com a beneficis a aportar per part del sistema, s'inclou el poder visualitzar el nombre d'entrades i sortides per mesos, anys o per clients, aportant informació a temps "real" del nivell de residus existents a planta. També aportarà benefici afegit el poder visualitzar si un client està disminuint el nombre d'entrades o està incrementant el seu volum. Com que s'ha de complir amb la regulació de l'ACR, el sistema també permetrà extreure els informes pertinents del tonatge total rebut a partir d'un rang de dates específic, així com per cada client en concret.

3 ESTAT DE L'ART

Després de realitzar una cerca de programari que respongui als objectius d'aquesta aplicació, s'han trobat diversos resultats en l'àmbit del reciclatge de residus encarats al registre de les entrades i sortides d'aquests. S'han seleccionat dos softwares que compleixen amb els objectius que té el nostre client. Aquests dos sistemes s'anomenen Teixo [3] i Sage X3 [4], ambdós enfocats a la gestió de residus i en la automatització dels tràmits administratius i logístics. Els dos sistemes permeten l'introducció d'entrades i sortides entre d'altres moltes funcionalitats.

Teixo permet gestionar el tractament i la transformació de residus amb informació de processos i de resultats. Genera una memòria anual en què es visualitza la comparativa de la compra i de la venda de residus per processar vinculada a cada client. Disposa de funcionalitats que administren diferents tipus de tractaments segons les característiques dels residus. Controla la recollida i el trasllat de residus i les diferents sortides de productes processats i de deixalles finals.

Per altra banda, *Sage X3* permet gestionar les ofertes i contractes amb tercers, tramitar las incidències en planta i en temps real o establir i modificar els preus dels serveis i importar-los automàticament als albarans o comandes.

Com que són softwares de pagament i no disposen d'una versió de prova, no s'ha pogut accedir a totes les funcionalitats que ofereixen i per tant l'estudi es basa en les funcionalitats que anuncien a les respectives pàgines web.

Podem veure per tant, que tot i que existeixen alternatives al software que volem desenvolupar, la intenció del nostre programari és que sigui el més ajustable, personalitzat i a mida per l'empresa, fet que les altres dues alternatives, al ser més genèriques i incloure més funcionalitats no s'ajusten a les seves necessitats. Aquestes funcionalitats extres encareixen el producte final, i en aquest cas no són necessàries o requerides per la planta de reciclatge on va destinada la nostra aplicació.

4 DISSENY

El desenvolupament del treball va començar per una captura de requeriments, és a dir, saber ben bé que necessitava l'empresa i per tant concretar els objectius anteriors. Per fer efectiva aquesta presa de requisits es va realitzar una entrevista amb el responsable de la planta, encarregat d'introduir totes les dades a mà, per fer la llista de funcionalitats-objectius especificada en l'apartat anterior d'objectius.

Posteriorment es va dissenyar un primer prototip per poder-lo mostrar i validar aquests objectius i requeriments amb l'empresa. En aquest primer disseny de la interfície de l'aplicació s'ha realitzat un prototip no funcional [Figura 1] per visualitzar com es veurien distribuïts els diferents gràfics i funcionalitats.

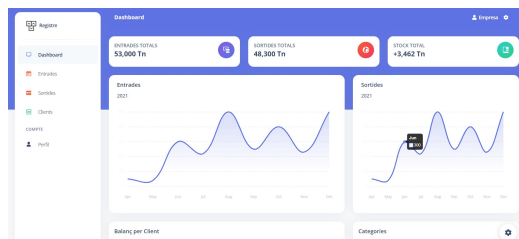


Figura 1: Prototip no funcional

Un cop ensenyat aquest primer prototip a l'empresa, es va decidir que s'havia de construir un prototip funcional per a què l'operari donés un feedback real de la funcionalitat del sistema, ja que només amb captures del prototip era difícil la obtenció de resultats per a poder millorar la interfície i usabilitat. Tot i això, l'empresa va comentar que els gràfics no són prioritaris a la hora de fer servir l'aplicació, sinó més aviat disposar d'una drecera que et redirigeixi directament a la pantalla on poder afegir les entrades o sortides, i poder visualitzar el nivell de *stock* actual tant d'entrades com de sortides.

Amb aquesta primera iteració del prototip no funcional, i el feedback proporcionat de l'empresa, es va realitzar una segona iteració del disseny del prototip, en aquest cas funcional [Figura 2] i utilitzant una eina anomenada Marvel [5], al prototipatge i *testing* d'aplicacions. L'eina permet crear una interfície totalment personalitzable, ja que disposa d'una varietat de formes, botons i icones molt extensa, podent fins i tot importar les imatges o icones que es desitgin.

L'eina ens permet seleccionar diferents seccions de cada pantalla dissenyada, per què un cop seleccionades, et redirigeixin a una altra pantalla. Per tant, es pot crear una funci-

onalitat i usabilitat molt propera a la que tindria l'aplicació final.

Per tant, en aquesta primera iteració del prototip ja es pot interactuar una mica més amb la interfície, podent visualitzar la interfície on s'introdueix una entrada [Apèndix B3], visualitzar l'historial de total d'entrades afegides [Apèndix B4], poder accedir a la pàgina d'edició d'una entrada en concret [Apèndix B5], i finalment filtra-les per clients [Apèndix B6].

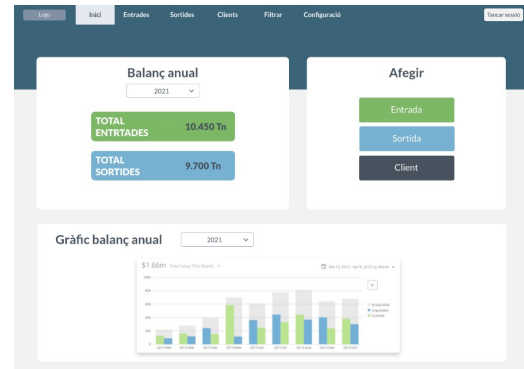


Figura 2: Prototip Marvel

Un cop compartit aquest primer prototip, es va captar el feedback del client i es va crear un segon prototip, també compartit amb el client. Com a millores s'ha introduït la possibilitat de poder eliminar una entrada, poder filtrar per dates en el conjunt d'entrades, ja que en l'anterior prototip s'havia d'especificar un client en concret, i la possibilitat d'editar les taules de tipus de poda, residu i material en el cas de que es volguessin afegir més opcions.

Per últim, una tercera iteració va ser necessària per realitzar l'últim prototip. En aquest cas es va decidir preparar una una pauta [Apèndix A1] amb un seguit d'accions a realitzar per l'operari per monitoritzar si les funcionalitats eren difícils de trobar, i on es podien col·locar per ser accedides de la manera més òptima.

El prototip final es pot trobar al següent enllaç [6] .

5 FUNCIONALITATS DE L'APLICACIÓ

L'aplicació permet introduir les dades corresponents a entrades de residus i sortides de material processat, però també filtrar aquestes dades per data i client. Més concretament, les principals funcionalitats són:

- Mostrar un tauler principal on es disposarà de les dades més rellevants i dreceres a les opcions més utilitzades. El tauler s'ha dissenyat a partir del prototip ensenyat al client, i s'ha decidit que les dades que es visualitzaran seran:
 - Balanz numèric de la quantitat d'entrades i sortides, numèricament i gràficament, podent seleccionar l'any desitjat
 - Dreceres a les operacions d'afegir Entrades, Sortides i Clients.
- Efectuar les quatre operacions bàsiques CRUD (Create, Read, Update i Delete) sobre les entrades i sortides.

- Gestió de clients: es disposarà de les mateixes operacions menys la d'eliminació, ja que comportaria inconsistències en entrades o sortides ja realitzades per aquell client.
- Realitzar una llista de consultes sobre la base de dades que hem decidit que són les més necessàries. A partir d'un rang de dates, es podrà filtrar el conjunt d'entrades o sortides comprès dins del rang, com també per un client en específic. Un cop s'especifica si es vol les entrades totals o només per un client en específic, es retornarà una taula amb les dades corresponents i un gràfic per poder veure visualment els resultats, agrupats per mesos o per dies depenent del rang de dates especificat.

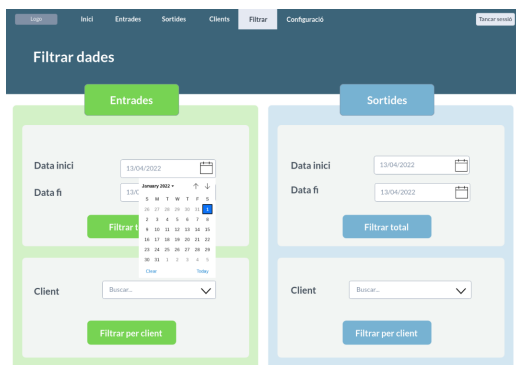


Figura 3: Prototip filtrar dades

- Finalment es disposa d'un apartat per configurar els diferents tipus de residus, poda i materials que es faran servir al moment d'introduir les entrades i sortides. Aquesta funcionalitat està en una secció a part ja que s'hi accedirà en molt poca freqüència.

En aquest últim apartat també es disposarà d'un formulari on poder introduir les matrícules corresponents a cada client.

6 METODOLOGIA

Per desenvolupar, planificar i fer un seguiment de les tasques que s'han de dur a terme en el projecte, s'ha decidit fer servir l'eina *Kanban*. Aquesta eina està basada en una metodologia que permet tenir una visió de les tasques a realitzar al projecte, separant-les en llistes. Per aquest projecte s'han creat tres llistes: llista de tasques, en procés i acabades. Per organitzar les diferents fases del desenvolupament, s'ha creat un taulell general on s'hi han introduït les 6 fases de desenvolupament del projecte [7]:

1. Anàlisi
2. Planificació
3. Disseny
4. Continguts
5. Desenvolupament
6. Testing

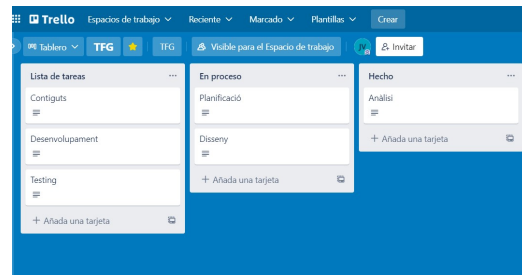


Figura 4: Tauler general Trello

Per cada una de les tasques s'ha creat un taulell per separat, incloent les subtasques que s'han de realitzar en cada una d'elles.

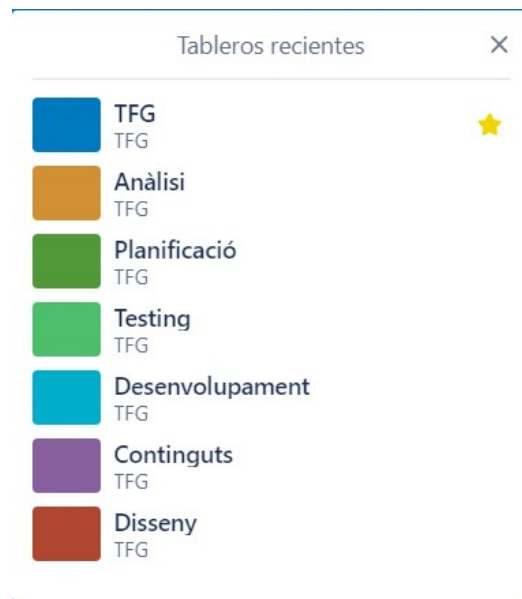


Figura 5: Taulers de tasques en Trello

7 EINES I TECNOLOGIA

Per començar a desenvolupar el projecte vaig pensar en els diferents llenguatges de programació utilitzats durant el grau (PHP, C++, Python...). En algunes assignatures també es van estudiar diversos patrons de disseny, com per exemple el MVC (Model-Vista-Controlador), un patró de disseny utilitzat per a la implementació d'interfícies d'usuari. Aquest patró de desenvolupament de programari divideix l'aplicació en tres parts interconnectades: el model de dades, la interfície usuari i la lògica de control.

Així doncs, es va començar a desenvolupar l'estructura de l'aplicació utilitzant PHP, ja que és l'últim llenguatge de programació utilitzat per a crear una aplicació web i per tant més recent en quant al seu funcionament, i el patró MVC per definir el disseny de l'aplicació.

Com que l'aplicació estarà *hostejada* en un cloud provider, també es va fer un estudi dels principals proveïdors de cloud computing: AWS, Azure i Google Cloud. Es va escollir la darrera opció ja que disposa de característiques i productes que ja coneixia i s'adaptaven a les necessitats del sistema que es vol desenvolupar. Més endavant s'explicaran quins són aquests productes i per què s'han escollit.

En provar una primera versió de l'aplicació, on només s'havia creat la base de dades i alguns vistes per visualitzar dades, em vaig adonar que la tria del llenguatge de programació no m'acabava de fer el pes, ja que al no utilitzar cap framework del llenguatge (com per exemple Laravel o Symfony), havia de crear molt més codi per a l'autenticació d'usuaris, sessions, cookies, etc. A més a més, tot i que no hi va haver problemes per executar l'aplicació en el cloud, el seu temps d'execució i resposta eren molt més elevats que amb altres llenguatges com Go, JavaScript o Python.

Així doncs, vaig començar a mirar informació sobre *frameworks* de PHP per desenvolupar l'aplicació que fossin més eficaços. En aquest punt, vaig meditar si realment volia implementar l'aplicació en PHP. Després d'una recerca de llenguatges que s'adaptessin i permetessin la implementació de les funcionalitats desitjades, es va optar per desenvolupar-la amb Python, ja que és un llenguatge ràpid, eficaç, amb molta documentació pública i amb una gran varietat de llibreries de fàcil implementació.

7.1 Django

Després de realitzar una cerca sobre *frameworks* de Python, vaig trobar bastant interessant Django, ja que segueix el patró MVC, hi ha molta documentació a la seva pàgina oficial i en webs de tercers, tutorials per crear una aplicació de test per aprendre a utilitzar-lo, i sobretot la facilitat, potència i flexibilitat que caracteritza el llenguatge de programació Python.

Però sense dubte, les dues principals característiques que em van fer decantar escollir aquest *frameworks* van ser la incorporació implícita de:

- Una interfície per administrar els usuaris i els grups d'usuaris (incloent una assignació detallada de permisos).
- Correspondència d'objectes relacional: una tècnica de programació per convertir dades de llenguatges de programació orientats a objectes en la seva representació en bases de dades relacionals, a través de la definició de les correspondències entre els diferents sistemes
- Els diferents canvis realitzats a la base de dades es guarden en forma de migracions, les quals s'emmagatzemen a l'aplicació com un control de versions, i per tant en el cas de realitzar un canvi erròniament, es pot tornar a la última migració disponible.

7.2 Google Cloud

Des del primer moment s'ha enfocat el desenvolupament per anar lligat a l'execució en el cloud. S'ha escollit la plataforma de Google com a proveïdor de núvol ja que ofereix una versió de prova per poder utilitzar els seus productes fins a 300€. Altres alternatives com AWS o Azure també ofereixen crèdits gratuïts per començar a desenvolupar un servei al núvol, però actualment dispenso de més experiència en els productes de Google, per tant em vaig decantar cap aquesta plataforma.

Per tant, necessitem diversos productes per poder on executar l'aplicació, la base de dades, i els arxius estàtics:

Cloud Run. És una plataforma que permet executar contenidors que es poden invocar mitjançant sol·licituds o esdeveniments. Cloud Run "no" té servidor (serverless): fa abstracta tota la gestió de la infraestructura, de manera que no cal configurar el servidor (planificació de la capacitat, configuració, gestió, manteniment, etc). S'ha escollit aquest servei ja que permet l'augment automàtic de recursos destinats a l'aplicació depenent del nombre d'usuaris accedint a la plataforma, i també disposa d'un control de versions, per poder migrar gradualment els usuaris a noves versions del software. En aquest producte s'executarà la nostra aplicació desenvolupada amb Django, i per tant on s'executarà tota la lògica i vistes de l'aplicació. Com que el producte es basa en contenidors, no es disposa d'estat ni de memòria no volàtil. Per tant, necessitem algun producte més per la nostra base de dades i arxius no temporals.

Cloud SQL. És un servei de bases de dades totalment gestionat que ajuda a configurar, mantenir, gestionar i administrar les bases de dades relacionals. S'ha escollit aquest servei ja que ofereix nativament la possibilitats de fer backups periòdics, manteniment, actualitzacions de seguretat automàtiques i monitoratge, entre d'altres.

Cloud Storage. És un servei per emmagatzemar els objectes a Google Cloud. Un objecte és una peça de dades immutable que consisteix en un fitxer de qualsevol format. Els objectes s'emmagatzemen en contenidors anomenats buckets. En aquest producte s'emmagatzemaran els arxius estàtics (fitxers CSS i JS), i imatges si se'n utilitzen per a l'aplicació.

Cloud Build. És un sistema creat per a l'automatització de l'execució de les aplicacions i sistemes implementats al cloud. En aquest projecte l'utilitzarem per implementar un CI/CD pipeline, una pràctica que combina la integració i desplegament continu. Aquest producte ens ajudarà al desenvolupament i integració des de que es desenvolupen noves funcionalitats de l'aplicació fins a que es puja a l'aplicació en producció al cloud, integrant aquest procés. Aquest procés s'explicarà més extensament a la secció CI/CD.

8 DESENVOLUPAMENT

Per començar a desenvolupar l'aplicació, es va començar la cerca de documentació del llenguatge Django integrat a Google Cloud, on es fés ús de bases de dades relacionals. Es va trobar aquest tutorial [8] on s'integra una aplicació de mostra de la pàgina oficial de Django i s'executa al cloud.

Es va fer la configuració pertinent al projecte cloud, activant els productes involucrats i creant els recursos necessaris, en aquest cas un contenidor a Cloud Run, una instància de Cloud SQL i un bucket a Cloud Storage.

Un cop comprovat que tots els productes funcionaven adequadament i sense errors, es va procedir a executar la mateixa aplicació en local, ja que per desenvolupar la nostra aplicació necessitem poder depurar-la.

Un cop comprovat que la configuració és possible i funciona correctament, es va procedir a la creació de l'aplicació.

8.1 Estructura de l'aplicació

Django funciona en diferents directoris, on cada un d'ells consta del patró MVC (model-vista-controlador), que específicament en aquest framework s'anomena MTV (model-template-view). El patró MVC és un patró d'arquitectura de programari que separa la presentació de dades (frontend) de la lògica (backend). Ha existit com a concepte durant un temps i ha experimentat un creixement exponencial en l'ús. També s'ha descrit com una de les millors pràctiques de crear aplicacions client-servidor [9], i podem veure que tots els principals *frameworks* per a web es construeixen al voltant del concepte MVC.

Per començar el desenvolupament de l'aplicació, es creen 6 directoris corresponents a cada una de les diferents seccions del sistema: inici, entrades, sortides, clients, filtrar i configuració.

Cada directori, anomenat *app* per Django, conté els mateixos fitxers:

- **models.py:** fitxer on es declaren les diferents taules que es crearan a la base de dades (ex. la taula d'entrades amb els seus respectius camps).
- **views.py:** fitxer on s'implementarà la lògica de l'aplicació, i per tant el punt en comú entre el backend i el frontend (en aquest cas entre model i template).
- **templates:** directori on aniran tots els fitxers html.
- **urls.py:** fitxer on es declaren les rutes de la secció corresponent (ex. la ruta entrada/crear o sortida/editar/1)

Aquests són els principals fitxers, però més endavant se'n crearan de nous, com per exemple *forms.py* per implementar els formularis, o *tables.py* per a la visualització de les dades en taules.

També es disposa d'un directori comú on es crearan vistes com la de la barra de navegació o el peu de pàgina, els quals seran comuns per a totes les vistes.

8.2 Administració

Com s'ha comentat anteriorment, una de les grans avantatges d'aquest *frameworks* és que porta integrat un sistema d'administració per defecte, el qual podem personalitzar al nostre gust, i on podem crear nous usuaris per l'aplicació.

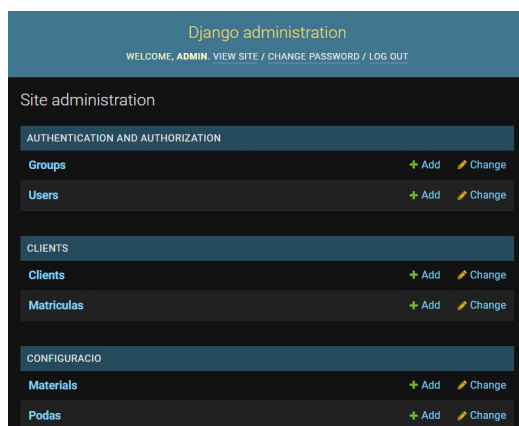


Figura 6: Panell d'administració

Aquest panell d'administració ens ha permès la introducció i modificació de les primeres dades visualitzades a l'aplicació, així com l'assignació dels primers usuaris amb diferents permissions dins l'aplicació (administrador i usuari).

El sistema ens permet assignar permisos granularment a cada usuari, així que en un futur si només volem donar a un usuari permisos per visualitzar i no per editar o eliminar entrades, tindrem la opció d'implementar-ho.

8.3 Migracions

Django funciona amb migracions, per tant cada vegada que s'efectua un canvi en algún fitxer *models.py*, ja sigui per crear una nova taula a la base de dades, o per editar-ne una d'existent, s'ha de crear una nova migració per actualitzar la base de dades.

Les migracions són la manera de Django de propagar els canvis fets als models (afegir un camp, suprimir un model, etc.) a l'esquema de la base de dades. Estan dissenyades per ser generades automàticament, a excepció de si s'està intentant realitzar un canvi que posa en risc la integritat de les dades ja existents a la base de dades.

Per realitzar la migració, és tant senzill com executar dues comandes: una anomenada *makemigraton*, on es defineix la migració, i l'altre *migrate*, on es realitza la migració a la base de dades.

Els principals beneficis d'aquestes migracions són:

- Control de versió: Com que les migracions s'emmagatzemen al control de versions, en el cas d'aplicar-ne una erròniament es pot tornar a l'anterior versió.
- Mapatge d'objectes relacional: la pròpia migració ens convertirà els canvis realitzats a les taules editades automàticament en llenguatge SQL.

8.4 CI/CD

Anteriorment hem mencionat que per implementar una continua integració i desenvolupament de l'aplicació s'ha fet servir el producte Cloud Build.

La Integració Contínua (Continuous Integration – CI) permet integrar contínuament codi en un únic repositori compartit i de fàcil accés. L'entrega contínua (Continuous Delivery – CD) permet descarregar el codi emmagatzemat al repositori i desplegar-lo contínuament en producció. El procés de CI / CD és una forma ràpida i eficaç per portar un producte al mercat abans que la competència, a més de llançar noves funcions i corregir errors per mantenir l'aplicació al dia segons els requeriments dels clients actuals.

En el nostre cas, l'aplicació està pujada a Github, amb un repositori per l'aplicació en fase de desenvolupament, i per tant executada en local, i un altre repositori amb l'aplicació en fase de producció, executada en el cloud.

Per pujar l'aplicació en el cloud es pot fer de dues maneres:

- Executar manualment comandes per desplegar l'aplicació en els diferents serveis del cloud.
- Utilitzar la pràctica CI/CD, que consisteix en crear un fitxer *cloudbuild.yaml* amb les comandes que executem manualment i crear un Trigger (activador), el qual

inicia automàticament una compilació sempre que hi hagi canvis en codi font, en aquest cas en el nostre repositori de Github. Es pot configurar el Trigger per compilar el codi amb qualsevol canvi al repositori d'origen o només canvis que coincideixin amb determinats criteris.

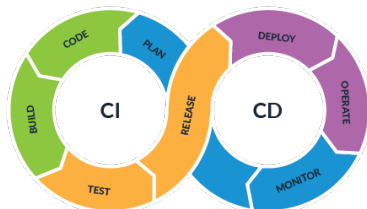


Figura 7: CI/CD

9 RESULTATS

En aquesta última fase del projecte s'ha realitzat testing de l'aplicació final amb el client, utilitzant una eina de user testing anomenada Loop11 [10].

Aquesta eina ens permet preparar tasques que el client ha d'anar realitzant. Un cop acabada cada tasca, el client ha de contestar una pregunta per donar feedback de les funcionalitats que ha trobat a faltar. L'eina porta integrada la funcionalitat de gravació de pantalla del client durant tota la sessió per posteriorment analitzar quines funcionalitats han estat més difícils de trobar.

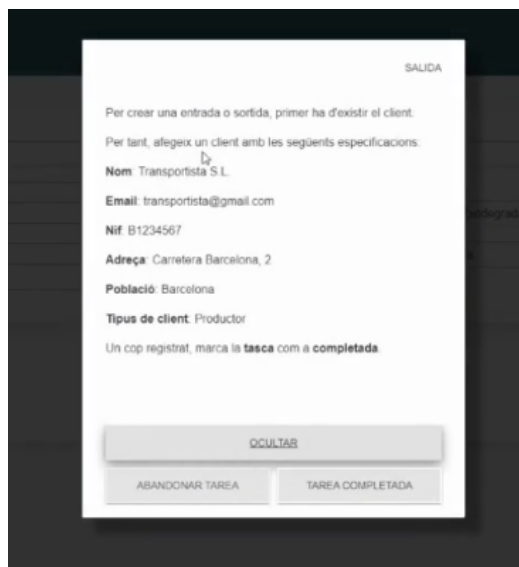


Figura 8: Tasca en Loop11

Per realitzar el testing, s'han preparat 11 tasques amb les seves respectives 11 preguntes per donar feedback de la tasca corresponent. S'han realitzat tasques com la d'inici de sessió, creació i modificació de dades introduïdes, filtrar aquestes dades per dates i clients i exportar-les en els diferents formats disponibles. Llistat complet de les tasques realitzades a l'annex en la Subsecció A.2.

Com a feedback aportat pel client, les funcionalitats que ha trobat a faltar i que per tant han de ser modificades en l'aplicació han sigut:

- A la pàgina d'inici de sessió, tenir l'opció de visualitzar la contrasenya.
- Tenir un accés directe a la creació de nou tipus de client, material, poda i residu des de la mateixa pàgina de creació de la entitat corresponent.
- A la pàgina d'edició, canviar el botó 'Editar' per 'Guardar', ja que és més intuïtiu.
- També a la pàgina d'edició, canviar el botó 'Eliminar' per 'Eliminar entrada', ja que ajuda a saber què s'està eliminant.
- Es van presentar dos formats de dates per visualitzar l'historial d'entrades i sortides: "15 Maig 2022" i "15/06/2022". El client ha escollit el format 'dd/mm/yyyy'.
- Per filtrar les entrades i sortides no els hi ha quedat clar quina diferència hi ha entre filtrar per client o sense, per tant es fusionarà en un mateix formulari i es posarà una nota explicant el funcionament d'aquest.

Un cop realitzat l'estudi, l'extracció del feedback mencionat anteriorment, i la implementació dels canvis suggerits, s'ha realitzat un nou estudi per contabilitzar el guany en temps que aporta l'aplicació en comparació amb el model utilitzat fins l'actualitat, el qual era a mà.

S'han escollit 10 entrades reals de 10 clients diferents, i s'ha comparat el temps en que s'introdueixen a mà i a l'aplicació.

Registrar una entrada a mà es tarda una mitjana de 41 segons, mentre que fent servir l'aplicació es tarden de mitjana uns 17 segons. Veiem doncs que la diferència és de 24 segons, i si ho multipliquem per 10 entrades veiem que el guany de temps és de 4 minuts.

Nº	A mà	Aplicació
1	39,2	18,1
2	36,5	16,9
3	42,7	15,6
4	39,7	16,1
5	41,2	17,2
6	45,2	16,5
7	39,9	18,3
8	44,2	15,9
9	39,3	16,8
10	42,6	17,1
	41,1	16,9

Taula 1: Comparació de temps emprat en realitzar 10 entrades, en segons

Aquest guany a priori no és molt notable, però on realment el client està agilitzant el procés és en el filtrat de dades, ja que anualment han de crear un informe del tonatge entrat per cada client, i un informe del total d'entrades i sortides processat. Aquest informe consta de dues dades que s'havien de buscar i calcular manualment: el NIF del client, el qual el buscaven manualment a la última factura realitzada al client, i sumar el total de tonatge, el qual era molt propens a un error humà en el càlcul.

Amb la nostra plataforma s'agilitzen i simplifiquen l'obtenció d'aquestes dues dades necessàries pels informes ja que es generen i s'obtenen el l'ordre de segons, al contrari que quan es feia a mà, que podia comportar hores de feina.

Pel que fa a la interfície i usabilitat de l'aplicació, s'ha aconseguit crear un sistema quasi idèntic al prototip desenvolupat i acordat amb el client durant la fase de d'anàlisi i disseny. Aquest fet que ha facilitat molt aquesta última fase de testing amb el client ja que moltes funcionalitats han sigut les esperades. Per altra banda, també s'han trobat matisos que calia canviar en la interfície, tal i com he indicat anteriorment [Apèndix C].

10 CONCLUSIONS

Un cop acabades totes les fases del projecte, acabant amb el testing de l'aplicació amb el client, podem concloure que l'aplicació s'ha adequat als objectius plantejats inicialment pel client.

Com a objectius principals assolits, podem incloure el balanç numèric de la quantitat d'entrades i sortides, molt important per complir amb la regulació de l'ACR, així com poder extreure informes a mida del tonatge total d'un client en concret a partir d'un rang de dades, molt important també pel client ja que fins ara només es podia fer a mà.

Aquesta funcionalitat és molt important pel client ja que aporta un guany de temps molt important, ja que com ha estat comentat anteriorment el client realitzava aquests informes a mà, podent tardar varies hores (des de 30-45 minuts a 2 hores) depenent del client. Amb l'aplicació aquest informe es genera en segons.

Podem veure que tot i realitzar un prototipatge extens, havent realitzat fins a tres iteracions del prototip per captar tots els requeriments del client, encara han sorgit noves modificacions al fer el testing final amb l'aplicació en una fase quasi final, on s'esperava que totes les funcionalitats implementades ja eren les esperades per l'empresa.

Crec que aquestes dues fases són molt importants, ja que si no es realitzen poden acabar amb un producte que no s'adapta als requeriments del client, havent suposat una càrrega de treball no necessària i entregant un producte que no és funcional, i encara més important, no ajustat al client.

En aquest cas però, el client ha quedat molt satisfet de poder participar en totes les fases de desenvolupament de l'aplicació, ja que ha pogut decidir les diverses funcionalitats, com accedir a aquestes i en quin format poder visualitzar-les. Ha sigut un software fet a mida per l'empresa, la qual ha pogut participar activament en la presa de decisions i presenciar l'evolució del producte final.

Tot i haver acabat el projecte assolint tots els objectius plantejats inicialment, ja s'han parlat de noves funcionalitats a poder incloure dins l'aplicació. En un futur s'implementarà una secció de facturació, ja que el client crea les factures a partir del tonatge entrat per cada client.

Com que aquesta informació ja està present a la plataforma, es podrà crear una nova taula on poder especificar les tarifes i preus actuals, i crear una factura per a cada client.

AGRAÏMENTS

M'agradaria donar les gràcies al tutor del projecte Joan Serrat, per tots els consells que m'ha donat durant el desenvolupament d'aquest projecte, així com el contacte directe i eficaç que s'ha mantingut de principi a fi.

També m'agradaria donar un especial agraïment a l'empresa on s'ha destinat aquesta aplicació ja que ha tingut molta paciència en totes les iteracions fetes de la presa de requisits, prototipatge i testing de l'aplicació, ja que han sigut molt útils per tirar endavant amb el projecte.

REFERÈNCIES

- [1] <https://residus.gencat.cat/ca/ambits-dactuacio/tipus-de-residu/residu-industrials/gestors/obligacions-i-responsabilitats/>
- [2] <https://residus.gencat.cat/ca/ambits-dactuacio/tipus-de-residu/residu-industrials/gestors/darig/>
- [3] <https://teimas.com/es/teixo>
- [4] <https://www.onegolive.com/software-gestion-residuos/>
- [5] <https://marvelapp.com/>
- [6] <https://marvelapp.com/prototype/i331767>
- [7] <https://grupovisualab.com/las-7-fases-de-desarrollo-de-un-sitio-web-optimo/>
- [8] <https://cloud.google.com/python/django/run>
- [9] <https://www.geeksforgeeks.org/benefit-of-using-mvc/>
- [10] <https://www.loop11.com/>

APÈNDIX

A.1 Accions a realitzar dins del prototip

1. Entrada

- Afegir una entrada
 - Data: 13 d'Abril de 2022
 - Productor: Ajuntament 2
 - Transportista: Transportista 1
 - Matricula: 0000 XXX
 - Quantitat: 2,7
 - Tipus residu: Restes de teixit vegetal (020103)
 - Tipus poda: N/A (no especificar, deixar en blanc)
 - Registrar l'entrada
- Modificar una entrada
- Eliminar una entrada

2. Sortida

- Afegir una sortida
- Modificar una sortida
- Eliminar una sortida

3. Clients

- Afegir un client
- Modificar un client

4. Filtrar

- Filtrar sense especificar client
 - Data inici: 01/01/2022
 - Data fi: 13/04/2022
- Filtrar especificant client
 - Data inici: 01/01/2022
 - Data fi: 13/04/2022
 - Client: Jardiner XX

A.2 Tasques realitzades durant la sessió d'usuari testing

Introducció: Gràcies per realitzar aquest estudi. Se't demanarà completar varies tasques per provar la usabilitat de l'aplicació per adaptar-la completament al teu gust. Cada tasca se't presentarà automàticament en pantalla. Un cop començada la tasca, apareixerà una finestra rodona amb el nom "MOSTRAR" al costat inferior esquerra. Quan hagi finalitzat la tasca, obre la finestra i marcal-la com a Tasca Completada. Comencem!

Nota: els números que no estan presents en la següent llista són les preguntes que es fan acabada cada tasca per recollir el feedback del client.

1. Iniciar Sessió:

Per iniciar sessió, introdueix les següents credencials:
Usuari: xxxxx Contrasenya: yyyyyy

Un cop iniciada la sessió, marca la tasca com a completada.

2. Afegir client:

Per crear una entrada o sortida, primer ha d'existir el client. Per tant, afegeix un client amb les següents especificacions:

Nom: Transportista S.L.
Email: transportista@gmail.com
Nif: B1234567
Adreça: Carretera Barcelona, 2
Població: Barcelona
Tipus de client: Productor
Un cop registrat, marca la tasca com a completada.

4. Crear nou tipus de client:

Ens hem adonat de que el client no és de tipus Productor, sinó Transportista. Per tant, haurem d'afegir un nou Tipus de Client: Transportista. Un cop acabada la tasca, marca-la com a completada.

6. Editar Client:

Perfecte, ara que hem creat un nou tipus de client, hem d'editar el client creat anteriorment per canviar-lo al tipus Transportista.

8. Afegir entrada:

Ara que ja tenim el client creat correctament, afegirem una nova entrada.

Data: 2/06/2022
Productor: Client1
Transportista: Transportista S.L.
Matrícula: 1111AAAA

Quantitat: 3,5

Tipus residu: Residus biodegradables (2002001)

Tipus poda: Poda neta

Un cop feta la tasca marca-la com a completada.

10. Afegir entrada +:

A continuació afegirem una altra entrada, però fixa't que en aquest cas hauràs d'escollir camps que encara no existeixen. ...

Data: 22/05/2022

Productor: Ajuntament

Transportista: Transportista S.L.

Matrícula: 1111AAAA

Quantitat: 4

Tipus residu: Restes de teixit vegetal (020103)

Tipus poda: Poda bruta

Un cop hagi afegit l'entrada, marca la tasca com a completada.

12. Editar entrada:

Ens hem adonat que l'última entrada afegida està incorrecte, ja que la quantitat no és 4, sinó 5,5. Edita l'entrada i marca la tasca com a completada.

14. Afegir sortida:

Ara que ja sabem com afegir i editar entrades, passem a les sortides. Crea una sortida amb la següent informació:

Data: 2/06/2022

Productor: Client2

Transportista: Transportista S.L.

Destinació: Client1

Quantitat: 10

Tipus material: Triturat de poda fi

Un cop acabada la tasca marca-la com a completada.

17. Eliminar sortida:

La sortida del dia "13/06/2022" va ser afegida accidentalment, així que volem eliminar-la. Un cop acabada la tasca marca-la com a completada. D'ara en endavant ja no es posarà l'avís de completar la tasca un cop acabada, però continua fent-ho com fins ara.

19. Filtrar entrades:

Volem filtrar el total d'entrades introduïdes el mes de Maig. Posteriorment volem descarregar-les en format PDF.

21. Filtrar sortides:

Volem filtrar el total de sortides introduïdes des de l'1 de Gener de 2022 a avui, del client anomenat Client1. Posteriorment volem descarregar-les en format CSV.

B.3 Prototip Marvel - Historial d'entrades

Historial d'entrades

+ Registrar nova entrada

Exportar

Data	Productor	NIF	Transportista	Matricula	Tipus residu	Quantitat (Tn)	Tipus poda	Acció
13/04/2022	Ajuntament 2	ZZZZZ	Transportista1	0000XXX	Residus de toixot vegetal (020103)	2,7	Poda bruta	Editar
01/03/2022	Jardiner3	YYYYY	Jardiner3	8888RRR	Residus de toixot vegetal (020103)	2,4	Poda neta	Editar
01/03/2022	Jardiner2	XXXXXX	Jardiner2	1111EEE	Residus biodegradables (200201)	7,8	Soca/palmera	Editar
01/03/2022	Jardiner2	XXXXXX	Jardiner2	1111EEE	Residus biodegradables (200201)	0,8	Poda neta	Editar
01/03/2022	Jardiner1	PPPPP	Jardiner1	80000XX	Residus de toixot vegetal (020103)	2,75	Neteja vïaria	Editar

Mostrar 5 resultats

1 2 3 ... 20 21 22

B.4 Prototip Marvel - Editar entrada

Editar Entrada

Data: 14/04/2022

Productor: Ajuntament 2

Transportista: Transportista1

Matricula: 0000XXX

Quantitat (Tn): 2,7

Tipus residu: Residus de toixot vegetal (020103)

Tipus poda: Poda bruta

Eliminar Guardar

B.5 Prototip Marvel - Filtrar entradas

Filtrar dades

Entrades

Data inici: 13/04/2022

Data fi: 13/04/2022

Client: Buscar...

Filtrar total

Filtrar per client

Sortides

Data inici: 13/04/2022

Data fi: 13/04/2022

Client: Buscar...

Filtrar total

Filtrar per client

C.6 Aplicació final - Historial d'entrades

Historial d'entrades

+ Registrar nova entrada

Exportar

Data	Productor	Transportista	Matricula	Quantitat (Tn)	Tipus residu	Tipus poda	Acció
15 Maig 2022	Client1	Client2	1111AAAA	4,1	Residus biodegradables (200201)	Poda neta	Editar
20 Abr. 2022	Client2	Client1	1111AAAA	3,0	Residus biodegradables (200201)	Poda neta	Editar
12 Maig 2022	Client1	Client1	1111AAAA	2,0	Residus biodegradables (200201)	Poda neta	Editar
21 Maig 2022	Client1	Client1	1111AAAA	4,5	Residus biodegradables (200201)	Poda neta	Editar
02 Juny 2022	Client1	Transportista S.L.	1111AAAA	3,5	Residus biodegradables (200201)	Poda neta	Editar

C.7 Aplicació final - Editar entrada

Editar entrada

Data: 15/05/2022

Productor: Client1

Transportista: Client2

Matricula: 1111AAAA

Quantitat: 4,1

Tipus residu: Residus biodegradables (200201)

Tipus poda: Poda neta

Eliminar Editar

C.8 Aplicació final - Filtrar entradas

Filtrar dades

Entrades

Data inici: dd/mm/aaaa

Data fi: dd/mm/aaaa

Client: Seleccionar un client

Filtrar total

Filtrar per client

Sortides

Data inici: dd/mm/aaaa

Data fi: dd/mm/aaaa

Client: Seleccionar un client

Filtrar total

Filtrar per client