

Treball de Fi de Grau

Enginyeria en Tecnologies Industrials

DIGITALITZACIÓ DE PROCESSOS EN L'ÀREA DE SUPPLY CHAIN

MEMÒRIA

Autor: Pedro Julio Delgado Carballo
Director: Lluís Solano Albajes
Convocatòria: Juny 2021



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resum

Aquest treball consisteix en l'anàlisi de la cadena d'aprovisionament d'una empresa mitjana tot identificant els problemes que provoca i la proposta de possibles solucions per a cadascun d'aquests.

Té com a objectiu la digitalització de certa part del procés de aprovisionament de l'empresa. S'ha aconseguit aquest objectiu digitalitzant l'entrada de mercaderia al magatzem mitjançant un prototip que consta d'un programa generador de codis QR (Quick Response) a partir de la informació d'un albarà i una aplicació que té com a funció llegir aquests codis mitjançant la càmera web de l'ordinador, descodificar-los i desar la informació en una base de dades. Aquest programa i aplicació estan programats en llenguatge Python en pràcticament la seva totalitat. La part que connecta l'aplicació i la base de dades esta parcialment escrit en llenguatge SQL (el llenguatge usat per la base de dades).

El programa té com a usuaris objectiu els proveïdors d'una certa empresa model. L'aplicació té com a usuari objectiu la pròpia empresa model, estalviant-se d'aquesta manera els possibles errors humans en la lectura i introducció de les dades mostrades en un albarà convencional a la base de dades corresponent.

El prototip, encara que només digitalitzi una part de la cadena de subministrament, dona peu a futures implementacions per a problemes (mencionats al treball) que es poden 'atacar' amb la mateixa eina tecnològica (els codis QR).

Sumari

SUMARI	3
1. GLOSSARI	6
1.1. MAGATZEM CENTRAL (identificador Centre→5000):	7
1.2. FÀBRICA DE CARN (identificador Centre→4003):	10
1.3. FÀBRICA DE PA (identificador Centre→4004):	10
1.4. LOCALS (identificador Centre→XXYY):	10
2. PREFACI	11
2.1. Origen del projecte	11
2.2. Motivació	11
2.3. Requeriments previs	11
3. INTRODUCCIÓ	13
3.1. Objectius del projecte	13
3.2. Abast del projecte	14
4. ANÀLISI DEL FLUX DE MERCADERIA	15
4.1. Entrada de mercaderia	15
4.1.1. Recepció de mercaderia	15
4.1.1.1. Explicació del procés	15
4.1.1.2. Problemàtica	16
4.1.1.3. Causes	16
4.1.2. Ubicació de la mercaderia a SEC	16
4.1.2.1. Explicació del procés	16
4.1.2.2. Problemàtica	17
4.1.2.3. Causes	17
4.1.3. Ubicació de mercaderia a FRI i PAN	18
4.1.3.1. Explicació del procés	18
4.1.3.2. Problemàtica	18
4.1.3.3. Causes	18
4.2. Reubicació	19
4.2.1. Explicació del procés	19
4.2.2. Problemàtica	20

4.2.3. Causes.....	20
4.3. Sortida de mercaderia.....	20
4.3.1. Preparació de comandes.....	20
4.3.1.1. SEC.....	21
4.3.1.2. FRI i PAN.....	22
4.3.2. Expedició de la mercaderia.....	22
4.3.2.1. Explicació del procés.....	22
4.3.2.2. Problemàtica.....	23
4.3.2.3. Causes.....	23
5. POSSIBLES SOLUCIONS	24
5.1. Recepció de mercaderia.....	24
5.2. Ubicació de la mercaderia a SEC.....	25
5.3. Ubicació de la mercaderia a FRI i PAN.....	26
5.4. Reubicació.....	26
5.5. Preparació de comandes.....	27
5.6. Expedició de la mercaderia.....	28
7. PROTOTIP	29
7.1. Tria de la solució a prototipar.....	29
7.1.1. Què és un codi QR?.....	29
7.2. Entorn de desenvolupament del prototip.....	29
7.2.1. MySQL.....	30
7.2.2. Python.....	30
7.3. Procés de desenvolupament del prototip.....	31
7.3.1. Creació de la base de dades.....	31
7.4. Generador de QR.....	33
7.4.1. Instal·lacions prèvies.....	33
7.4.2. Codi.....	33
7.5. Aplicació lectora de QR.....	34
7.5.1. Codi.....	35
7.5.1.1. Llibreries.....	35
7.5.1.2. Estètica.....	35
7.5.1.3. Funcions.....	36
CONCLUSIONS	42

BIBLIOGRAFIA _____ **44**

1. Glossari

Per poder fer un estudi en paral·lel del procés tant físic com digital per seguidament identificar fonts de problemes i fer propostes de millora (digitalització, auto-verificació...) cal posar-se en context de com funciona la part logística de l'empresa i definir la nomenclatura utilitzada dels diferents centres, magatzems...

Primerament es defineixen dos conceptes en els que es basa tota la cadena d'aprovisionament de l'empresa model:

- **FEFO (“First Expires First Out”)**: aquesta norma obliga al magatzem fer la sortida prioritària de la mercaderia que primer es caduca. Si tenim d'un mateix producte 2 lots amb diferent caducitat, quan arribin comandes dels locals, el primer lot que es farà servir és el de caducitat inferior (encara que aquest hagi entrat posteriorment a l'altre).
- **LIFO (“Last In First Out”)**: aquesta norma obliga a, quan la norma FEFO no sigui suficient per descartar tots els lots disponibles menys un, s'agafa el últim lot que s'ha entrat al magatzem com a primer de sortida.

L'empresa model es dedica a la restauració (una cadena de restaurants) està constituïda per diferents centres que es poden classificar en 3 categories: Magatzem central, Fàbriques i Locals.



1.1. MAGATZEM CENTRAL (identificador Centre→5000):

El magatzem central és un centre que es divideix en diferents centres logístics:

- **Centre Logístic 1(identificador magatzem→CL1):** Aquest centre logístic està ubicat a carrer Cal Mas nº115, juntament amb la fabrica de carn i les oficines de logística. Aquest magatzem esta, alhora, dividit en 3 punts de càrrega:

- o **SEC(identificador→P001):** en aquest punt de càrrega s'emmagatzemen les mercaderies que es poden conservar a temperatura ambient. Les posicions dels articles es classifiquen en com està emmagatzemada la mercaderia ("tipo de almacen") i la ubicació física que té en el magatzem.

Pel que fa al "tipo de almacén", es té:

- **Picking(identificador→PIC):** que estan accessibles al mosso per preparar comandes (altures inferiors a 2).
- **Reserva(identificador→RES):** que no estan accessibles pel mosso (altures majors a 1).
- **Vestuari(identificador→VES):** és un cas apart que fa referència al vestuari on s'emmagatzema la roba dels treballadors de locals i fàbriques.

Pel que fa a la ubicació física, es té:

- **"Platja"(identificador→902):** aquest és el punt de descàrrega de la mercaderia. D'aquí la mercaderia serà transportada a la seva ubicació corresponent.
- **Passadissos(identificador→ GXYYZ):** són 4 passadissos on s'ubiquen les mercaderies a diferents altures:
 - X=nº passadís
 - YY=profunditat

- Z=altura
- **Armaris de l'oficina(identificador→ OFICINA):** dos armaris on es troben mercaderia de consum propi de l'empresa (bolígrafs, llibretes,...).
- **Armari de farmaciola(identificador→BOTIQUIN):** és un armari on es troben els medicaments i els estris quirúrgics per fer els primers auxilis.
- **Armari a l'exterior de l'oficina(identificador→OFI. EXT):** és un armari on es guarden recanvis variats (per a cafeteres, rotllos d'etiquetes,...).
- **Armari aïllat(identificador→ARMARIO):** és un armari on es guarden productes químics.
- **Estanteries de l'altell(identificador→ALT XYZ):** és un altell organitzat en estanteries on es guarden productes variats (taules de terrassa, cadires,...).
 - X=nº d'estanteria.
 - YY=profunditat.
 - Z=altura.
- **Altell(identificador→ALTILLO):** és el mateix altell anterior, però amb un identificador diferent ja que la mercaderia no està ubicada a cap estanteria (simplement posat a terra).
- **Altell de manteniment(identificador→ALT MANT):** és un altell similar a l'anterior, però amb identificador diferent perquè en el passat havia sigut per a la secció de manteniment.
- **REFRIGERAT(identificador→P009):** en aquest punt de càrrega s'emmagatzemen les mercaderies s'han de conservar refrigerades. En aquest cas el "tipo de almacen" és únicament un amb **identificador→FRI**. Pel que fa a les ubicacions es tenen les següents:
 - **Nevera 1(identificador→N01).**



- **Nevera 2(identificador→N02).**
- **En qualsevol ubicació del FRI(identificador→GENERAL):**
aquí s'anoten les mercaderies que des de magatzem no saben on estan ubicades, però saben que està a una de les dos neveres dels dos congeladors.
- **CONGELAT(identificador→P010):** en aquest punt de càrrega s'emmagatzemen les mercaderies s'han de conservar congelades. En aquest cas el "tipo de almacen" és el mateix que per a articles refrigerats amb **identificador→FRI**. Pel que fa a les ubicacions es tenen les següents:
 - **Congelador 1(identificador→C01).**
 - **Congelador 2(identificador→C02).**
 - **En qualsevol ubicació del FRI(identificador→GENERAL):**
aquí s'anoten les mercaderies que des de magatzem no saben on estan ubicades, però saben que està a una de les dos neveres o dels dos congeladors.
- **Centre Logistic 2(identificador magatzem→CL2):** aquest és un magatzem "imaginari" que s'utilitza per comoditat alhora d'anotar les devolucions que ens arriben de les mercaderies que han sortit del CL1. Seguidament es farà una eliminació d'estoc si no es reutilitzable o tornarà a ser transferit al CL1 si és apte per distribució.
- **Centre Logistic 5(identificador magatzem→CL5):** Aquest centre logístic està ubicat juntament amb la fàbrica de pa i l'oficina d'aquesta. En aquest centre logístic es té un únic punt de càrrega:
 - **PA(identificador→P006):** en aquest punt de càrrega s'emmagatzemen les mercaderies produïdes al forn de pa (que s'han de conservar congelades). També, es possible emmagatzemar altres mercaderies congelades que poden provenir del CL1 si és necessari. En aquest cas només tenim un "tipo de almacen" que té **identificador→PAN** i una única ubicació anomenada P1000.

- **Centre Logístic 6(identificador magatzem→CL6):** aquest és un magatzem “imaginari” que s'utilitza per comoditat alhora d'anotar les devolucions que ens arriben de les mercaderies que han sortit del CL5. Seguidament es farà una eliminació d'estoc si no es reutilitzable o tornarà a ser transferit al CL5 si és apte per distribució.

*Cal notar que la ubicació exacta de les mercaderies només la tenim als passadissos i a les estanteries de l'altell. No es té constància si les mercaderies estan, o no, a l'abast del mosso que les ha de carregar en cap de les altres ubicacions físiques. Això prendrà especial importància alhora d'analitzar la reposició de mercaderies al FRI i al PAN.

1.2. FÀBRICA DE CARN (identificador Centre→4003):

La fàbrica de carn és el centre on es fa la producció de tota la part càrnica que fa servir als locals. Aquesta mercaderia serà, un cop produïda, enviada al centre 5000 (ja sigui el CL1 o el CL5) i, finalment seran enviades al local corresponent des d'aquest.

1.3. FÀBRICA DE PA (identificador Centre→4004):

La fàbrica de pa és el centre on es fa la producció de tot el pa que es fa servir als locals. Aquesta mercaderia serà, un cop produïda, enviada al CL5 del centre 5000 i, finalment seran enviades al local corresponent des d'aquest.

1.4. LOCALS (identificador Centre→XXYY):

Finalment, tenim els diferents locals. Cadascun té un identificador diferent:

- XX: és 10 per a locals ubicats a Espanya i 90 per a els locals ubicats a Andorra.
- YY: varia per a cada local.



2. Prefaci

2.1. Origen del projecte

Aquest projecte és fruit de la ineficiència i poca robustesa en la cadena d'aprovisionament de les empreses mitjanes en l'àmbit espanyol que he pogut presenciar durant la meva curta (però intensa) trajectòria professional.

2.2. Motivació

Abans de començar a treballar en l'àmbit d'aprovisionaments, tenia la visió que els magatzems de qualsevol empresa relativament "gran" estava molt més automatitzat del que realment ho estan. Tenia en ment magatzems com els que es poden veure en empreses com AMAZON. Quan vaig començar a treballar, vaig veure que això no era així. En un principi vaig emportar-me una petita decepció, però ràpidament la vaig redreçar cap a una il·lusió per intentar tenir un impacte tangible en l'empresa.

Aquestes millores, a més a més de augmentar la robustesa i eficiència de la cadena d'aprovisionament, facilita la feina dels treballadors implicats.

Tot això m'ha motivat a tirar endavant el projecte amb gran il·lusió.

2.3. Requeriments previs

Per tal de portar a terme aquest projecte, he hagut de informar-me dels fonaments de l'eina de treball SAP. A més he hagut d'aprendre una base d'SQL i aprofundir en el meu coneixement de el llenguatge de programació Python.

3. Introducció

La cadena de subministrament, en els últims anys, ha començat a prendre una importància indiscutible independentment de la indústria de que es parli. Això és, en gran mesura, donat per l'avenç que s'ha produït en el sector per part de certes empreses. Un clar exemple pot ser AMAZON, que ha elevat l'estàndard esperat pel consumidor de manera molt considerable mitjançant magatzems automatitzats, entregues amb terminis d'entrega inimaginablement curts, etc...

Aquesta millora en el sector ha creat una necessitat en moltes empreses de actualitzar el seu *Modus Operandi* per sobreviure. Aquest és el cas per a empreses multinacionals que han hagut (entre d'altres coses) d'agilitzar processos i fins i tot automatitzar-los per complet.

Hi ha, però, empreses que s'han quedat un pas enrere. Empreses on la cadena de subministrament no és el seu principal negoci i, mentre funcioni el negoci, no canvien els seus processos. És el cas de petites i mitjanes empreses que tenen una llarga trajectòria i tenen costum en fer els processos d'una certa manera. Empreses familiars que han crescut en valor de mercat, però els seus processos han canviat només quan ha sigut imprescindible (noves normatives, per exemple).

A l'estudi d'aquestes empreses i la millora dels seus processos sense canviar completament la seva manera de funcionar (amb una inversió assumible per aquesta) és on es centra aquest projecte.

Ens centrarem en una empresa model que es dedica a la restauració. Combina la producció de productes propis i la compra de producte acabat a proveïdor extern.

3.1. Objectius del projecte

L'objectiu principal del projecte és la digitalització parcial del flux de mercaderia dins dels magatzems de l'empresa.

3.2. Abast del projecte

Per realitzar aquest projecte, es farà un anàlisi exhaustiu del flux de mercaderia dels magatzems en qüestió, identificant així els problemes i ineficiències de la cadena d'aprovisionament.

Seguidament, es citarà de quines maneres es poden solucionar els problemes esmentats.

Finalment es focalitzarà en una de les solucions proposades per a un dels processos i farem un prototip de la possible solució per verificar la utilitat d'aquesta. D'aquesta manera es podrà convèncer a l'empresa a fer una inversió en el projecte per poder-lo dur a terme en la cadena d'aprovisionament de l'empresa i que s'interessin en les altres solucions proposades que encara no s'ha fet el prototip.



4. Anàlisi del flux de mercaderia

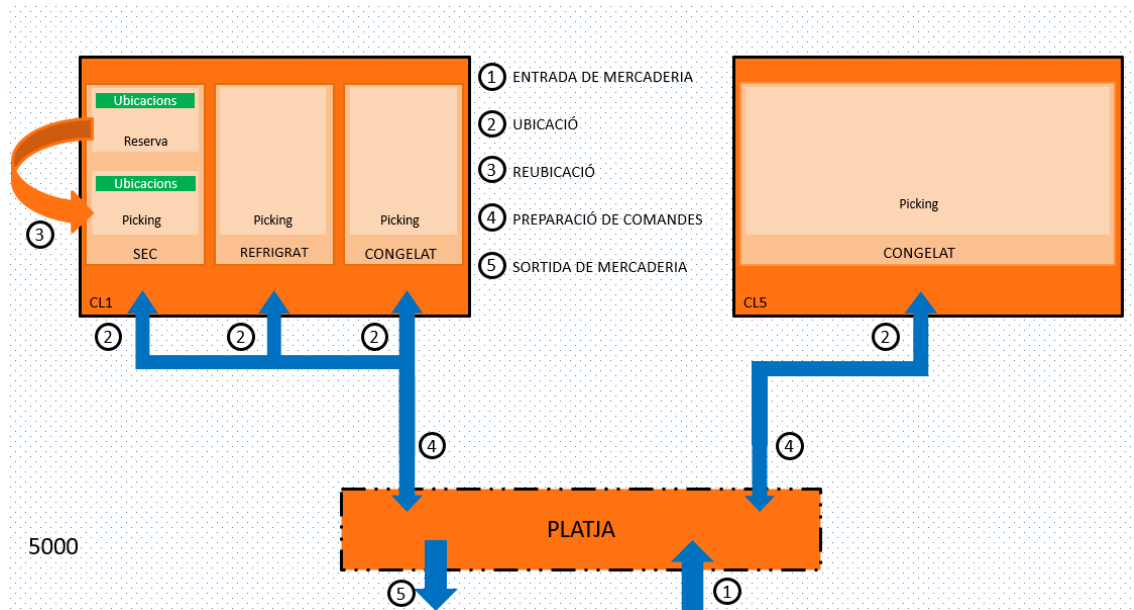


Figura 1: diagrama de blocs de la distribució de processos en els diferents magatzems

4.1. Entrada de mercaderia

4.1.1. Recepció de mercaderia

4.1.1.1. Explicació del procés

El transportista entrega la comanda juntament amb l'albarà i es fa una verificació visual que la mercaderia rebuda i la informació a l'albarà coincideixen. Un cop verificat l'albarà, el mossos omple el llibre de recepcions i el porta a l'administratiu juntament amb l'albarà i, aquest, introdueix al sistema SAP les variables que consten al llibre de recepcions. S'imprimeix una etiqueta on consta la següent informació:

- Ordre de transport (OT).
- Material amb el nom real del material.
- Quantitat d'entrada (amb la unitat corresponent).
- N° de lot
- Comanda de compra.

- N° de proveïdor amb el nom real de proveïdor.
- Data d'entrada.
- Data de caducitat.

*Si per emmagatzemar el lot es necessita més d'un palet, s'imprimirà una etiqueta per cada un d'aquests si esta a SEC. En qualsevol altre punt de càrrega només s'imprimeix una etiqueta i s'haurà de forçar que se'n imprimeixin les altres.

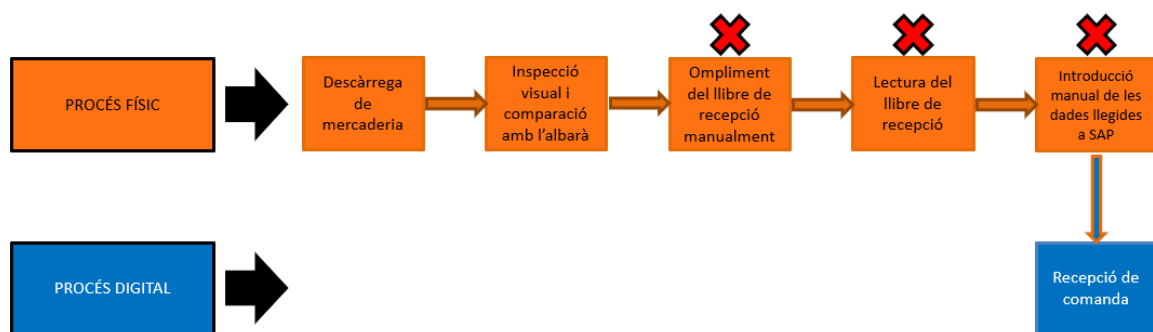


Figura 2: diagrama de blocs de l'entrada de mercaderia

4.1.1.2. Problemàtica

Es poden introduir de forma errònia les dades de la mercaderia que consten a l'albarà. Es veu clarament al diagrama de blocs els passos on pot produir-se un error de lectura, o tipogràfic que pugui produir una pèrdua de traçabilitat de la mercaderia entrada, un desquadrament d'estocs i/o un incompliment del FEFO.

4.1.1.3. Causes

La inspecció visual i l'escriptura a mà pot portar l'entrada errònia de les dades de les mercaderies.

4.1.2. Ubicació de la mercaderia a SEC

4.1.2.1. Explicació del procés

Un cop es té l'etiqueta, un responsable de magatzem (RM) introdueix a SAP l'ordre de transport o el busca (explicat a l'apartat digital) i el sistema li mostra

la localització recomanada pel lot en qüestió. En el cas que el RM vulgui canviar aquesta ubicació, haurà de seguir el procés de reubicació (explicat a l'apartat de reubicació més endavant).

Seguidament els mossos col·loquen el lot en qüestió en la ubicació definida en el pas anterior. No hi ha verificació posterior per saber si el lot ha estat ben col·locat o no.

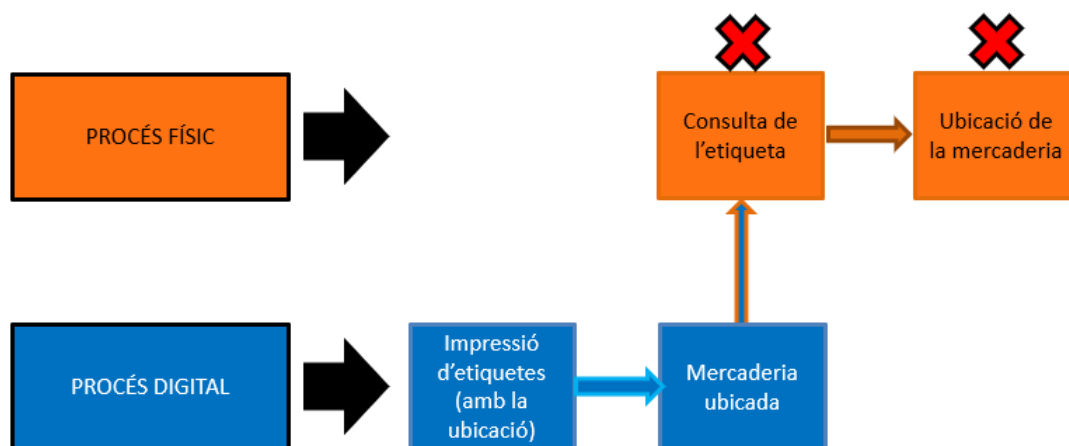


Figura 3: diagrama de blocs de la ubicació de mercaderia a SEC

4.1.2.2. Problemàtica

La mercaderia es col·loca erròniament. A SAP consta que la mercaderia està en una ubicació i realment està en una altra. Això pot produir un incompliment del FEFO i reubicacions d'emergència a mig torn.

4.1.2.3. Causes

La lectura de la localització de la proposta podria ser errònia pel mossos o simplement per rutina, es col·loca el palet entrant amb la metodologia FIFO (sense abans haver aplicat la FEFO).

4.1.3. Ubicació de mercaderia a FRI i PAN

4.1.3.1. Explicació del procés

El procés és idèntic al de la ubicació a SEC, però SAP no dóna una ubicació concreta. Simplement dóna la cambra on s'ha de col·locar (N01,N02,C01,...). En aquestes cambres no hi ha ubicacions concretes i la mercaderia es col·loca com al responsable de magatzem li sembli millor.

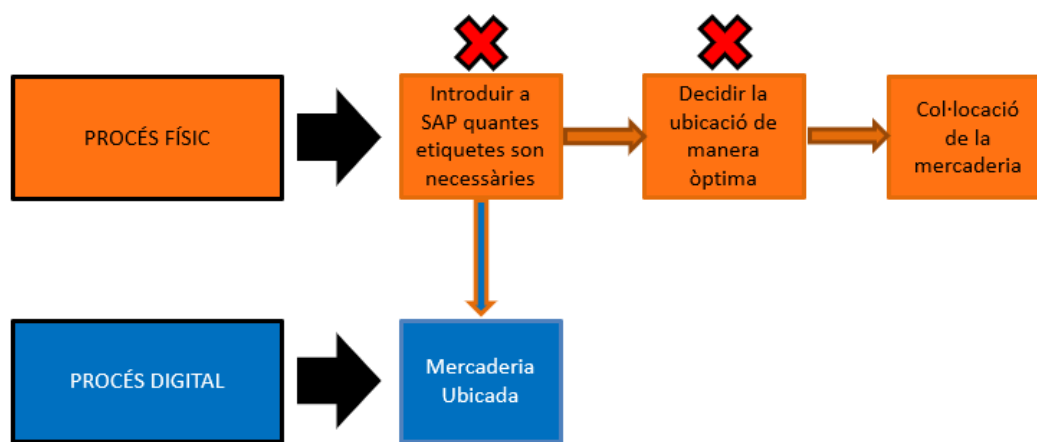


Figura 4: diagrama de blocs de la ubicació de mercaderia a FRI i PAN

4.1.3.2. Problemàtica

No se sap exactament on està la mercaderia. Això fa que no se sàpiga si està col·locada en la ubicació òptima.

4.1.3.3. Causes

No hi ha ubicacions concretes definides a SAP. Tenim l'estoc en bloc tot i tenir prestatges.

4.2. Reubicació

4.2.1. Explicació del procés

Cap al final de la jornada, el RM camina pel magatzem apuntant de manera visual en un full quants palets creu que s'haurien de baixar de RES a PIC (per inspecció visual/intuïció).

Accedeix al sistema i busca quin/s lot/s tenen la caducitat més propera (explicat a l'apartat digital). Aquests seran els lots que tindran prioritat de reubicació de RES a PIC d'acord al procediment FEFO. En cas que el FEFO no descarti el número de lots que volem reubicar, aplicarem el procediment LIFO (Last In First Out).

Un cop triats el/s lot/s a reubicar, el RM entra al SAP i defineix la nova ubicació, material, lot/s a moure...

Finalment es mou la mercaderia en qüestió a la seva nova ubicació.

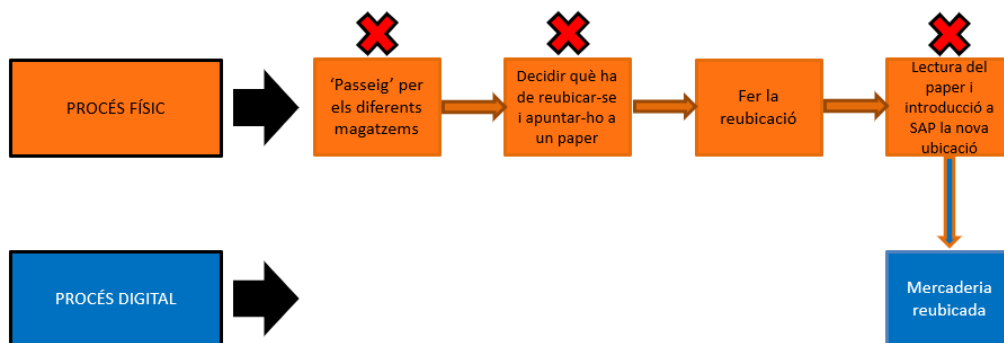


Figura 5: diagrama de blocs de la reubicació de mercaderia

4.2.2. Problemàtica

La mercaderia que hauria de ser reubicada pot ser passada per alta per el RM, provocant així una possible reubicació d'emergència a la meitat del torn.

4.2.3. Causes

Donat que el procés es fa per inspecció visual, el RM en qüestió pot tenir un descuit.

4.3. Sortida de mercaderia

4.3.1. Preparació de comandes

El centre en qüestió envia la comanda. El dia abans d'entrega aquesta comanda es comença a preparar. Els RM hauran 'carregat' manualment la comanda al GOTXI del mosso per tal de que pugui visualitzar la ubicació de cada material que hi ha a la comanda, la quantitat a agafar i el lot del qual l'ha d'agafar. També s'imprimeix la comanda en paper (de manera manual). Si a la comanda es demana PA, s'ha d'enviar la comanda en paper s'ha d'enviar aquella part de la comanda a l'impressora del CL5.

El mosso va al Picking dels materials que li van apareixent de la comanda esmentada i agafa la quantitat especificada i marca manualment al GOTXI que el material ha estat muntat a la gàbia. Un cop muntada tota la comanda en les gàbies necessàries, es fa una segona confirmació al GOTXI després de validar visualment que la comanda (impresa en paper) coincideix amb el material carregat a les gàbies especificat al GOTXI.

Una vegada la comanda esta preparada a les gàbies es porta el paper a l'administratiu per a la seva comptabilització al SAP i seguidament s'imprimeix l'albarà que el transportista agafarà quan mogui la mercaderia.

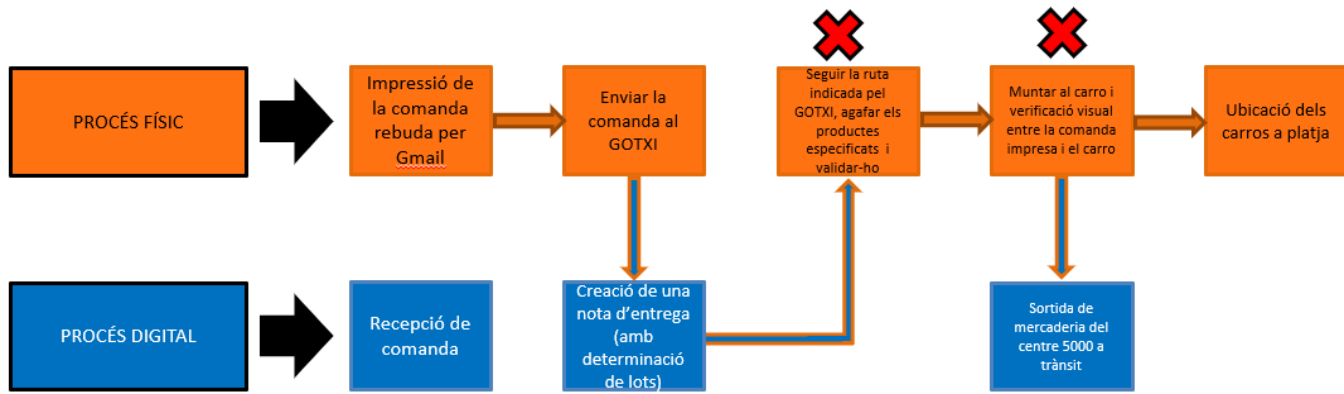


Figura 6: diagrama de blocs de la sortida de mercaderia

4.3.1.1. SEC

A SEC, el GOTXI li demana al mossos els productes en un cert ordre que té en compte el pes dels articles i el seu volum. D'aquesta manera el muntatge de carros es fa de manera més amena i intuïtiva.

La ruta però, és summament ineficient. Ho podem veure clarament a la **Figura #** on es mostra la ruta que fa el mossos al magatzem de SEC.

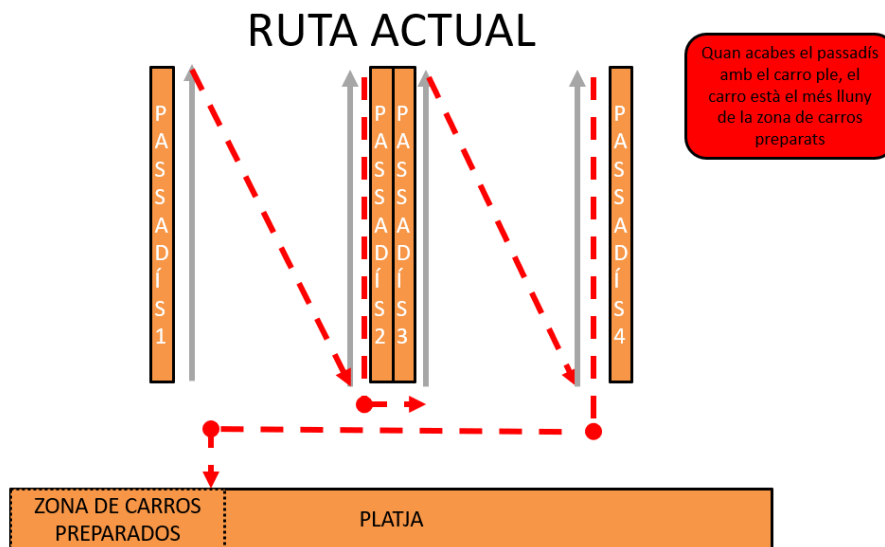


Figura 7: esquema de la ruta presa pels mossos a SEC en la preparació de comandes

4.3.1.2. FRI i PAN

A FRI i PAN no hi ha ubicacions concretes, tal i com hem comentat anteriorment. En aquest cas, quan s'envia una comanda per preparar al GOTXI la mercaderia s'envia per ordre alfanumèric. Això afecta tant a l'ordre de la cambra que es carregada com a cada producte demanat (ordre de referència).

Això fa que els mossos vagin d'una banda a l'altre de cada cambra buscant el material indicat pel GOTXI sense que ni tan sols els hi digui on es troba (ja que SAP no té constància de la ubicació dins de cada cambra).

Això genera rutes encara molt ineficients, ja que els materials no estan ordenats en cada cambra de manera alfanumèrica. És més, molts materials no estan físicament a la cambra que consta a SAP, fent que els mossos hagin de canviar de cambra varies vegades en la preparació d'una comanda.

4.3.2. Expedició de la mercaderia

4.3.2.1. Explicació del procés

Com hem comentat anteriorment, un cop la comanda ha estat preparada, en el sistema consta que la mercaderia està en trànsit. Això fa que tot el procés d'expedició sigui simplement físic.

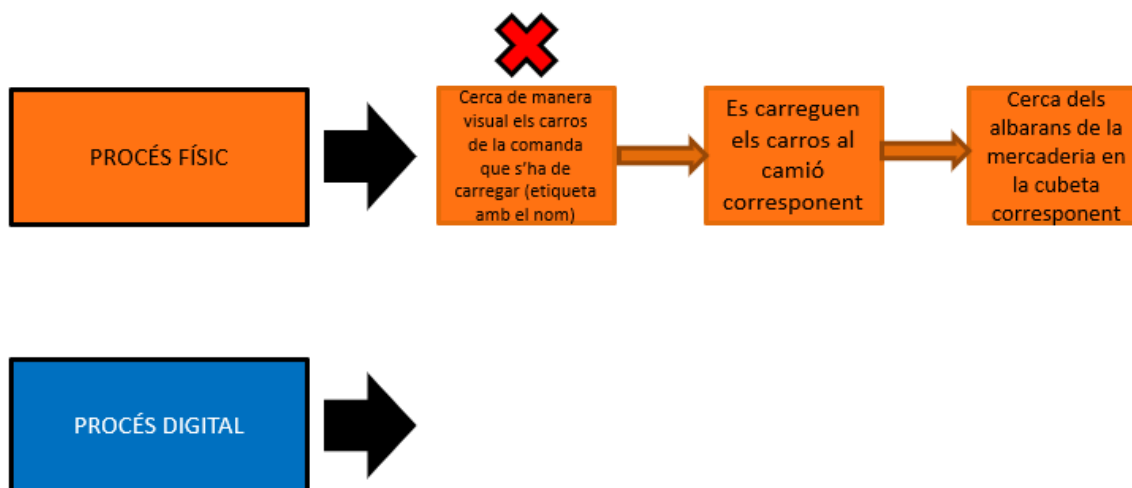


Figura 8: diagrama de blocs de l'expedició de la mercaderia

4.3.2.2. Problemàtica

Els carros queden preparats i s'enganxa una etiqueta amb el nom del local a on han de ser portats. El dia següent, el xofer que ha de fer la ruta, agafa els carros del/s local/s a on descarrega la mercaderia. Aquí no hi ha cap tipus de confirmació que els carros son els correctes i poden haver equivocacions per part del xofer ja sigui agafant els carros d'un altre local o deixant-se algun carro al magatzem.

No ens adonaríem de l'error fins que el local comenci a desmuntar els carros i guardar la mercaderia. En aquest punt el xofer (juntament amb el camió) han marxat del local. Això provoca comandes d'urgència i una possible pèrdua de traçabilitat de mercaderies.

4.3.2.3. Causes

Per rutina els xofers poden llegir per sobre les etiquetes i equivocar-se entre locals que tenen noms similars. També es podria donar la possibilitat de deixar-se un carro que no estigui perfectament visible a l'hora de carregar el camió.

5. Possibles solucions

En aquest apartat es discuteix una possible solució per a cada un dels problemes plantejats anteriorment.

S'ha intentat relacionar les màximes solucions per cridar l'atenció de l'empresa. Si molts problemes requereixen una inversió en un únic bé, l'empresa es podrà veure més interessada que si totes les solucions fossin utilitzant una tecnologia diferent (és més còmode habituar al personal a una nova tecnologia, que a moltes diferents).

5.1. Recepció de mercaderia

Una possible solució a la problemàtica associada a la recepció de la mercaderia al magatzem podria ser eliminar els possibles errors humans creant un sistema de albarans digitals mitjançant codis QR.

D'aquesta manera s'estaria eliminant la interpretació dels diferents responsables al llarg del diagrama de blocs esmentat anteriorment.

Aquesta solució constaria de dues parts i de la col·laboració dels proveïdors externs que té l'empresa.

1. Una aplicació (programa) generador d'albarans digitals en forma de QR on consti tota la informació que conté l'albarà físic. L'usuari d'aquesta aplicació (el proveïdor extern) fa corre l'aplicació i contesta a les preguntes que aquesta li fa. En acabar les preguntes, l'aplicació genera un codi QR que el proveïdor pot incloure al seu albarà.
2. Una aplicació capaç de llegir els codis QR generats per l'aplicació anterior, processar la informació i carregar-la a la base de dades corresponent.

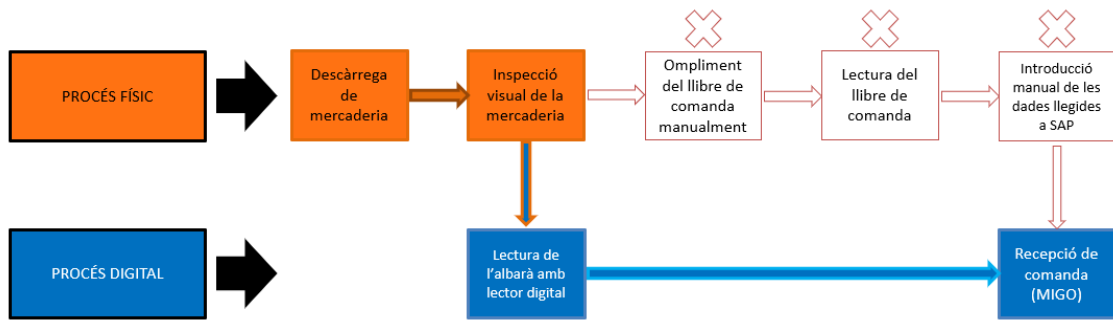


Figura 9: diagrama de blocs de la recepció de mercaderia

5.2. Ubicació de la mercaderia a SEC

Seguint amb la tecnologia QR, per aconseguir una eliminació de l'error humà a l'hora de fer les ubicacions de mercaderia es podrien crear codis QR per a cada una de les ubicacions del magatzem. D'aquesta manera es podria introduir una segona validació que la mercaderia ha estat ben col·locada mitjançant la lectura del codi de la ubicació amb el GOTXI que porten els mossos.

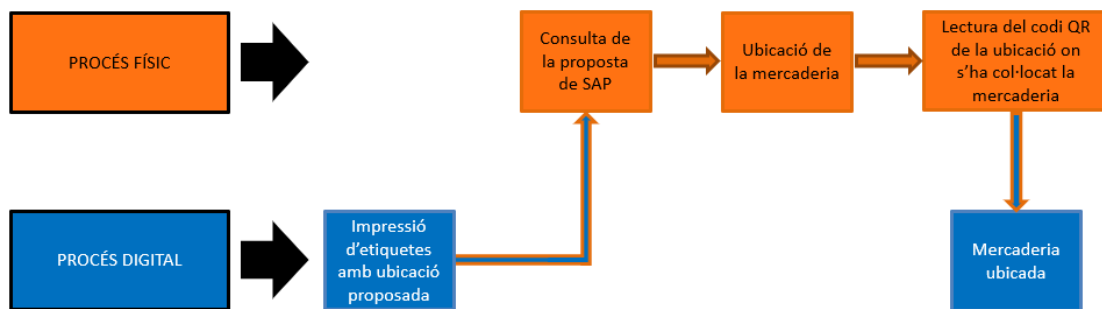


Figura 10: diagrama de blocs de la ubicació de mercaderia a SEC plantejada

5.3. Ubicació de la mercaderia a FRI i PAN

La millora d'aquest procés es pot dividir en dues parts. La primera seria crear les ubicacions de la mateixa manera que hi ha a SEC. Això comportaria la definició de cada ubicació en les estanteries que ja es tenen a les cambres en SAP. La segona part seria fer el mateix procediment de validació de la ubicació mercaderia mitjançant el codi QR amb el GOTXI que es fa a SEC.

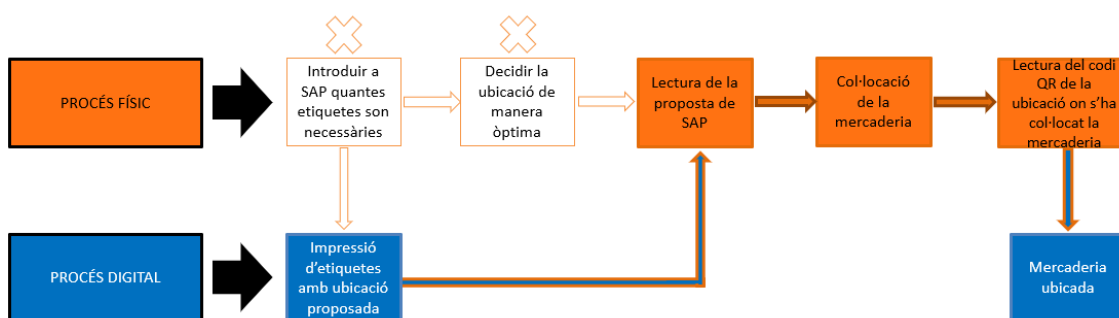


Figura 11: diagrama de blocs de la ubicació de mercaderia a FRI I PAN plantejada

5.4. Reubicació

Per tal d'eliminar els descuits per part dels RM a l'hora de decidir quins lots de mercaderia han de ser reubicats de RES a PIC, es pot fàcilment parametritzar la mercaderia mínima que hi ha d'haver a cada ubicació i de cada material. Això és una eina que ofereix SAP sense cap tipus de cost alternatiu.

Si ja està implementada la solució d'ubicació, també es podria validar la correcta reubicació llegint el codi QR de la ubicació un cop feta mitjançant el GOTXI.

Es veu que, a més a més d'eliminar el possible error humà, també s'està agilitzant el procés (s'estalvia la 'passejada' dels RM al final de la jornada).

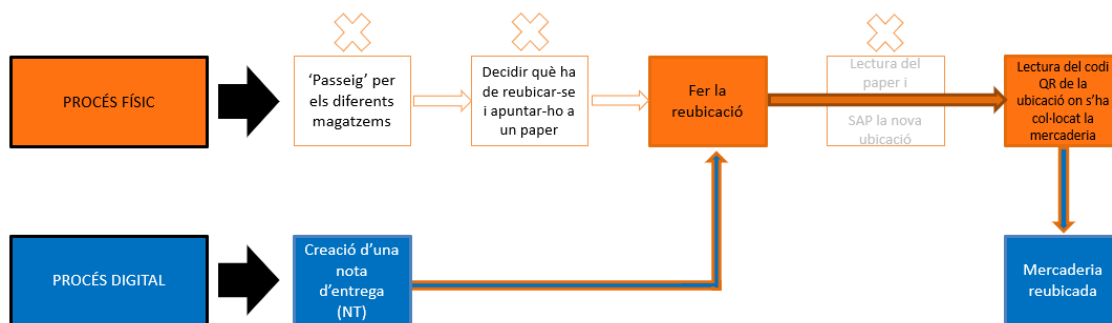


Figura 12: diagrama de blocs de la reubicació de mercaderia plantejada

5.5. Preparació de comandes

Un cop s'ha implementat la solució a FRI i PAN de crear ubicacions concretes igual que a SEC, simplement s'hauria de parametritzar les rutes per ser més òptimes i canviar la mercaderia adientment per tal de que el muntatge de carros sigui intuïtiu (primer les coses pesades, després les que no ho són tant).

Es pot observar a la **Taula #** una ruta molt més òptima per la preparació dels carros a SEC. A FRI i PAN com que no hi ha una ruta concreta es podrà adaptar a conveniència un cop es tinguin les ubicacions.

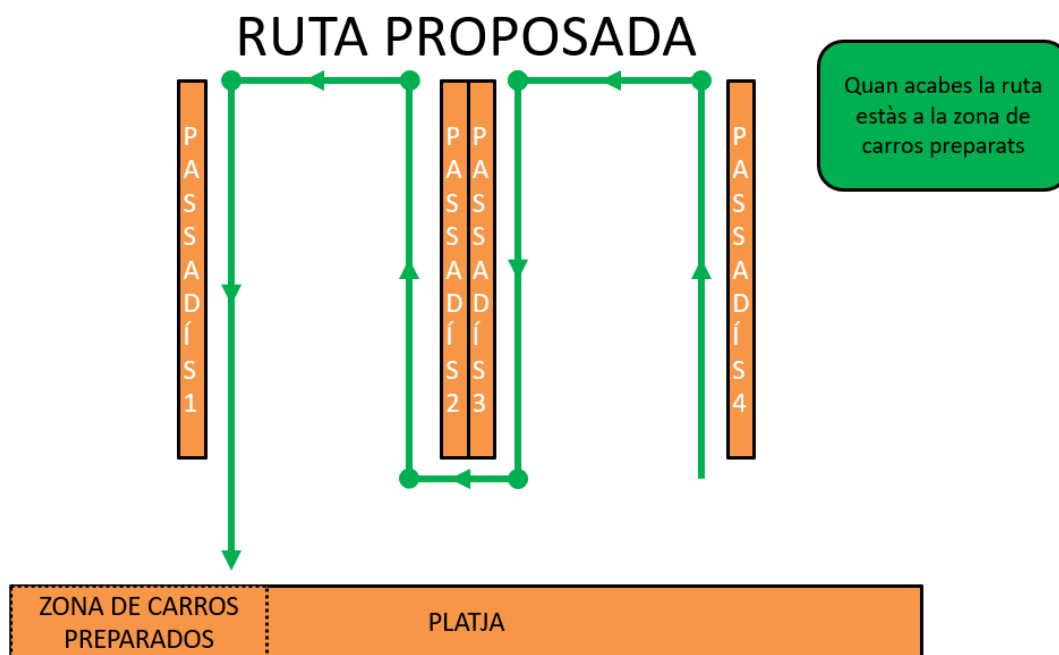


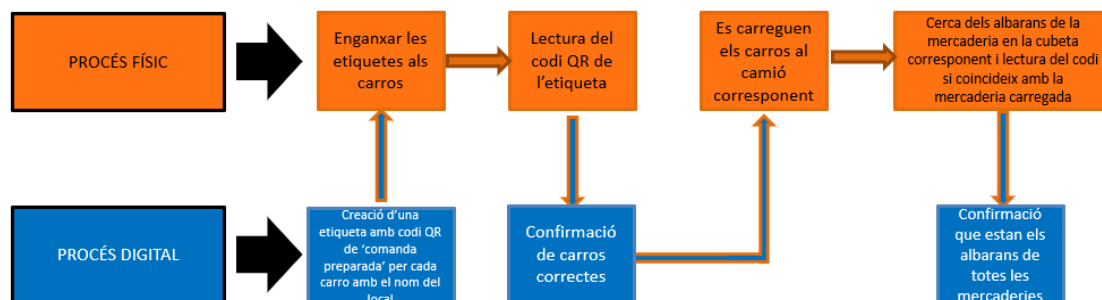
Figura 13: esquema de la ruta dels mossos al magatzem de SEC plantejada

5.6. Expedició de la mercaderia

Finalment s'arriba a la solució per a l'últim procés mencionat anteriorment, l'expedició de la mercaderia.

En aquest cas, s'hauria de crear la ubicació de PLATJA a SAP i així tenir una constància dels carros que estan al magatzem i els que estan realment en trànsit (recordem que a dia d'avui un cop s'acaba de preparar la comanda a SAP consta que està en trànsit).

Una vegada s'ha creat aquesta nova ubicació, també es podria crear un codi QR per a cada carro muntat i enganxant-lo a aquest perquè quan el xofer agafi els carros, hi hagi una validació de que els està agafant del local que li pertoca i no s'està deixant cap d'ells.



6. Figura 14 diagrama de blocs de l'expedició de mercaderia plantejada

7. Prototip

7.1. Tria de la solució a prototipar

Per fer la tria de la solució proposada a prototipar, s'ha buscat la solució que pot ensenyar realment el potencial de la digitalització parcial del magatzem.

Sense cap dificultat s'ha escollit prototipar l'entrada de mercaderia al magatzem. Un cop s'hagi validat la seva utilitat, totes les solucions que inclouen el QR seran molt més fàcilment acceptades per part de l'empresa.

7.1.1. Què és un codi QR?

Un codi QR (Quick Response code) és mòdul per emmagatzemar informació en una matriu de punts. Aquesta informació pot ser de qualsevol tipus (alfanumèrica, binària, numèrica, etc.).

El codi QR té certs avantatges davant dels codis de barres tradicionals (UPCs):

- Poden contenir un quantitat molt superior a un codi de barres tradicional. Normalment, aquests últims poden contenir fins a 25 caràcters diferents, mentre que els codis QR poden arribar a emmagatzemar fins a 640 bits o 4296 caràcters.
- Es poden llegir sense necessitat d'un làser. Simplement amb una càmera en un dispositiu que contingui un descodificador (molt comú en qualsevol smartphone). També es poden llegir des de d'una pantalla amb el codi, cosa que els UPCs no poden fer.

El fet de no necessitar un lector de barres làser fa que sigui significativament més econòmic implementar aquesta solució i fer el seu prototipatge.

7.2. Entorn de desenvolupament del prototip

Com bé s'ha explicat anteriorment, es crearan 2 aplicacions (generador de QR i lector de QR).

Per tal de poder ensenyar el potencial complet de la solució, també es crearà una base de dades en local que vagi registrant les dades dels albarans llegits.

7.2.1. MySQL

Les dades llegides pel lector (les dades que consten a l'albarà) s'hauran de guardar en una base de dades.

Pel desenvolupament d'aquest prototip, s'ha triat utilitzar MySQL per una varietat de motius:

- Flexibilitat i adaptabilitat: MySQL permet crear tant bases de dades simples, fins a bases de dades amb taules relacionades de poden suportar fins a Terabytes d'informació.
- Portable: Suporta Windows, Linux i UNIX.
- Facilitat d'ús: Al ser la meva primera vegada treballant amb una base de dades programada dinàmicament, m'he decantat per una opció relativament simple. El llenguatge utilitzat és SQL, que es ràpid d'aprendre a utilitzar (de manera bàsica).
- La versió utilitzada és gratuïta.

7.2.2. Python

S'ha escollit Python com a llenguatge per a desenvolupar les aplicacions per diferents motius:

- Portabilitat: Python és compatible amb els sistemes operatius més comuns a dia d'avui.
- Facilitat d'ús: Python és un llenguatge intuïtiu i relativament fàcil d'utilitzar. En el meu cas, la facilitat ha estat major donada la meva experiència acadèmica amb aquest llenguatge.
- Gran número de llibreries: aquest és un punt molt important a tenir en compte. Python consta de un gran llistat de llibreries que son molt útils a l'hora de programar. Pel que fa a la utilització de codis QR, Python consta de gran quantitat de llibreries específiques al tema.

7.3. Procés de desenvolupament del prototip

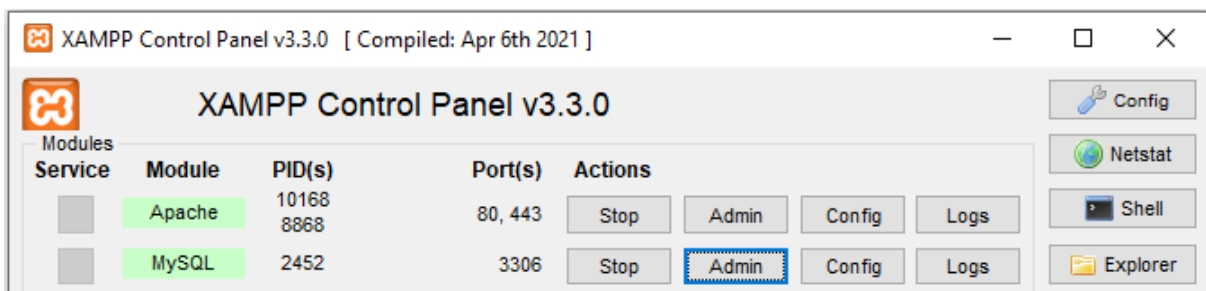
El desenvolupament del prototip constarà de fases:

- Creació de la base de dades.
- Creació del programa generador de QR.
- Creació del programa lector de QR i vinculament amb la base de dades.

7.3.1. Creació de la base de dades

Per tal de crear la base de dades en local s'ha descarregat l'aplicació XAMPP que té 2 components bàsiques.

- Apache: s'utilitza per crear el servidor local.
- MySQL: ens permet fer la base de dades (tal i com he explicat anteriorment).



Un cop tenim el XAMPP instal·lat, es pot passar a crear la base de dades. La base de dades l'he anomenat "recepció de mercaderia".

Un cop creada, s'accedeix a la pestanya "Estructura" on es pot crear una nova taula que anomenarem "productes_t_ambient".

En aquesta taula fixarem el número de columnes a 7 (que són les dades que consten en un albarà bàsic) que seran:

- Data de recepció
- Proveïdor
- Referència del material entregat

- Quantitat entregada
- UMB (Unitats de Mesura Base).
- Lot
- Data de caducitat del producte entregat (si és necessari).

Servidor: 127.0.0.1 » Base de datos: test

Estructura SQL Buscar Generar una consulta Exportar Importar

⚠ No se han encontrado tablas en la base de datos.

Crear tabla

Nombre: Número de columnas:

Continuar

Una vegada està definit el número de columnes, s'han d'anomenar i fixar certes característiques dels elements que s'introduiran a ella. Una vegada definides, ja es té la taula creada.

Servidor: 127.0.0.1 » Base de datos: recepción de mercadería » Tabla: productes_t_ambient

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento

Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<input type="checkbox"/>	1 Data_de_recepció	date			No	Ninguna
<input type="checkbox"/>	2 Proveedor	varchar(255)	utf8mb4_spanish2_ci		No	Ninguna
<input type="checkbox"/>	3 Referència_material	varchar(255)	utf8mb4_spanish2_ci		No	Ninguna
<input type="checkbox"/>	4 Quantitat	int(5)			No	Ninguna
<input type="checkbox"/>	5 UMB	varchar(5)	utf8mb4_spanish2_ci		No	Ninguna
<input type="checkbox"/>	6 Lot	varchar(255)	utf8mb4_spanish2_ci		No	Ninguna
<input type="checkbox"/>	7 Data_de_Caducitat	date			Sí	NULL

El tipus de cada columna ha estat definit segons la informació que se li introduirà:

- Date: s'ha definit aquest tipus de columna per a la data de recepció i la data de caducitat. Aquestes dates s'han d'introduir en el format adient (aaa-mm-dd).

- Varchar: s'ha definit en les columnes on es poden introduir combinacions de lletres i números tal com el nom del Proveïdor, la referència del material, l'UMB i el lot del producte. El "(255)" que podem veure al costat fa referència a la llargada màxima possible del Varchar. En aquest cas, 255 és la llargada màxima permesa pel programa.
- Int: s'ha definit a la columna de quantitat, ja que només volem que s'escriguin números en aquest apartat.

El fet de definir el tipus d'entrades que tindrà la taula ens dóna un cert control d'aquestes que ens serveix com una validació que la informació ha estat ben introduïda. Si no és en aquest format, ens saltarà un error.

7.4. Generador de QR

7.4.1. Instal·lacions prèvies

Per poder generar un codi QR amb Python, primerament hem d'instal·lar dues llibreries imprescindibles:

- 1- "pillow": és una llibreria de processament d'imatges en Python. Aquesta biblioteca ens proporciona el suport necessari per poder utilitzar la llibreria "qrcode". Proporciona una base sòlida per al processament d'imatges bàsiques.
- 2- "qrcode": aquesta llibreria ens facilita la creació de codis QR partint de qualsevol tipus de data (alfanumèrica, binaria,...) d'una manera molt fàcil.

7.4.2. Codi

Cal aclarir que per aquesta eina generadora de QR, s'ha creat una funció executable que, a partir de uns inputs del usuari (el proveïdor en qüestió), genera i guarda un arxiu ".png" al mateix directori on es té el programa de Python.

```
import qrcode
def crea_qr():
    data_recep=input('Data de descarrega (aaaa-mm-dd):')
    proveïdor=input('Nom del proveïdor :')
    ref=input('Referència del material :')
    quantitat=input('Quantitat entregada:')
    unitats=input('Unitat de mesura de la quantitat entregada: ')
    lot=input('lot :')
    data_cad=input('data de caducitat (aaaa-mm-dd) :')
```

Primerament s'importa la llibreria qrcode (la llibreria pillow no es necessària importar-la si ja s'importa la pillow). Seguidament es defineix la funció crea_qr() i també totes les variables que han de constar a l'albarà (les mateixes 7 variables explicades a la base de dades) mitjançant inputs per part del usuari. Aquesta informació es guarda en format 'string' per generar el codi QR.

```
img = qrcode.make(data_recep+' '+proveïdor+' '+ref+' '+quantitat+' '+unitats+' '+lot+' '+data_cad)
f = open("output.png", "wb")
img.save(f)
f.close()
```

En aquesta segona part del codi (que va seguit del codi anterior i dins de la funció crea_qr), es genera la variable "img" mitjançant l'atribut "make" de la llibreria qrcode. És un objecte "Pil Image" (de la llibreria pillow) que és el QR. Seguidament guardem la imatge en un document ".png" que s'anomena output (podria tenir qualsevol altre nom). Finalment tanquem el document.

7.5. Aplicació lectora de QR

L'aplicació consistirà en una finestra en la que es mostrarà la imatge recollida per la càmera (si aquesta està activada mitjançant el botó "INICIAR CAPTURA PER CÀMERA"). Un cop activada la càmera, simplement s'haurà de posar el codi QR davant d'aquesta fins que la informació que conté es vegi a l'espai inferior a la càmera i el botó "TANCAR CÀMERA".

Una vegada apareix el text a la 'caixa de text', es pot prémer el botó "DESAR INFO" que es troba a la part inferior de la finestra. Quan es prem el botó "DESAR INFO" es desarà un arxiu en format ".txt" (es podrà triar el directori on es desi) i automàticament s'afegirà a la base de dades definida anteriorment.

7.5.1. Codi

7.5.1.1. Llibreries

Primerament s'han d'importar les llibreries que seran necessàries per tal de poder començar a programar:

- tkinter: per a l'elaboració de la finestra de la APP amb tots els seus elements ("canvas", botons i la caixa de text).
- pyzbar: per a la lectura dels codis QR.
- numpy: per treballar amb els "arrays" dels "frames" generats per la càmera.
- opencv (cv2): per poder fer us de la càmera web.
- pillow (PIL): per al processament de les imatges (com he explicat anteriorment).

També s'importa la funció conector, que és una funció creada per mi per tal de connectar la APP amb la base de dades definida anteriorment. Estarà explicada més endavant.

```
from tkinter import *
import tkinter.scrolledtext as scrolledtext
from tkinter import messagebox, filedialog
from pyzbar.pyzbar import decode, ZBarSymbol
import cv2
import numpy as np
import threading
from PIL import Image, ImageTk, ImageDraw
from conector_bo import conector
```

Ara ja es pot procedir a inicialitzar la classe (que anomenarem "App") que contindrà els elements i funcions del nostre programa.

Es començarà la APP definint la càmera apagada (=false), posant buides les variables del sistema (self.info, self.codelist) que es faran servir més endavant com a variables globals i definint el nom de l'APP.

7.5.1.2. Estètica

Seguidament es definiran els elements visuals de l'aplicació:

```

class App:
    def __init__(self, font_video=0):
        self.active_camera = False
        self.info = []
        self.codelist = []
        self.appName = 'Lector de QR'
        self.ventana = Tk()
        self.ventana.title(self.appName)
        self.ventana['bg']='black'
        self.font_video=font_video
        self.label=Label(self.ventana, text=self.appName, font=15, bg='orange',
                          fg='white').pack(side=TOP, fill=BOTH)
        self.btnSave = Button(self.ventana, text="DESAR INFO", bg='light blue', command=self.desar)
        self.btnSave.pack(side=BOTTOM)

        self.display=scrolledtext.ScrolledText(self.ventana, width=86, background='grey'
                                                , height=4, padx=10, pady=10, font=('Arial', 10))
        self.display.pack(side=BOTTOM)

        self.canvas=Canvas(self.ventana, bg='black', width=640, height=0)
        self.canvas.pack()
        self.btnCamera = Button(self.ventana, text="INICIAR LECTURA PER CÀMERA", width=30, bg='light blue',
                                activebackground='red', command=self.active_cam)
        self.btnCamera.pack()
        self.ventana.mainloop()

```

Es crea la finestra i se li posa de títol el nom de la l'aplicació definit anteriorment. També es tria el color de fons i el color tant del títol com del fons d'aquest.

Seguidament es crea el botó de guardar (self.btnSave) i es 'carrega a la part inferior de la finestra (pack(side=bottom)). Un cop creat el botó de desar la informació es procedeix a crear la caixa de text (self.display) i també es 'carrega.

Ara es defineix l'objecte canvas on es mostrarà la imatge rebuda per la càmera. S'ha de tenir en compte que l'altura es fixa a 0 inicialment per tal que no es mostri al obrir l'aplicació. Això canviarà un cop es faci clic sobre el botó que ara es definirà.

Per acabar, com s'acaba d'esmentar, es defineix el botó que inicialitzarà la càmera (btnCamera) i farà que es mostri el canvas que tindrà com a nom "INICIALITZAR LECTURA PER CÀMERA".

Per tal de que la finestra es mostri oberta es posa mainloop().

7.5.1.3. Funcions

Una vegada ja definida l'estètica de l'aplicació, es passa a crear les funcions encarregades de llegir els codis QR i ensenyar el resultat a la caixa de text.

Es comença per la funció encarregada llegir el codi QR mitjançant la càmera web del nostre ordinador.

```

def active_cam(self):
    if self.active_camera == False:
        self.active_camera = True
        self.VideoCaptura()
        self.visor()
    else:
        self.active_camera = False
        self.codelist = []
        self.btnCamera.configure(text="INICIAR LECTURA PER CÀMERA")
        self.vid.release()
        self.canvas.delete('all')
        self.canvas.configure(height=0)

```

Aquesta funció simplement activa o desactiva la càmera web en funció de l'estat actual de la càmera (que al principi hem definit com a False). Si la càmera està desactivada (False), executa les funcions VideoCaptura() (per iniciar la càmera) i la funció visor() per mostrar la imatge a l'objecte canvas. Si en canvi està activa (True), finalitza la càmera i borra el contingut del canvas i torna a ensenyar el botó "INICIAR LECTURA PER CÀMERA".

A continuació es descriuran les dues funcions que executa la funció active_cam().

```

def VideoCaptura(self):
    self.vid = cv2.VideoCapture(self.font_video)
    if self.vid.isOpened():
        self.width=self.vid.get(cv2.CAP_PROP_FRAME_WIDTH)
        self.height=self.vid.get(cv2.CAP_PROP_FRAME_HEIGHT)
        self.canvas.configure(width=self.width,height=self.height)
    else:
        messagebox.showwarning("CÀMERA NO DISPONIBLE","El dispositiu està desactivat o no disponible")
        self.display.delete('1.0',END)
        self.active_camera = False

```

La funció VideoCaptura permet assegurar primerament que la càmera estigui encesa en el nostre ordinador, adaptem els valors de altura i grossor del nostre canvas (anteriorment amagat amb altura=0). Si per contra la càmera no està activada, es mostrarà un missatge indicant l'error i tornarà a posar l'estat en l'aplicació de la càmera com a False.

Un cop s'ha adaptat el gruix i l'altura del canvas, es procedeix a mostrar l'entrada de vídeo al canvas.

```

def visor(self):
    ret, frame=self.get_frame()
    if ret:
        self.photo = ImageTk.PhotoImage(image=Image.fromarray(frame))
        self.canvas.create_image(0,0,image=self.photo,anchor=NW)
        self.ventana.after(15,self.visor)

```

A la funció `visor()`, el que es fa és mostrar en el canvas la els frames obtinguts per la càmera amb una periodicitat de 0,0015 segons. Això s'aconsegueix mitjançant el mètode "`create_image()`". Els frames, a la vegada, son creats per la funció "`get_frame()`" que explicarem a continuació.

```
def get_frame(self):
    if self.vid.isOpened():
        verif,frame=self.vid.read()
        if verif:
            self.btnCamera.configure(text="TANCAR CÀMERA")
            self.capta(frame)
            return(verif,cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
        else:
            messagebox.showwarning("CÀMERA NO DISPONIBLE","""La càmera s'esta utilitzant en un altra aplicació. Tanqui-la i torni a provar.""")
            self.active_cam()
            return(verif,None)
    else:
        verif=False
        return(verif,None)
```

Aquesta funció és l'encarregada de proporcionar els frames a la funció `visor()` per mitjà del mètode "`.read()`" que s'emmagatzema a la variable `frame`. En aquesta funció també es verificarà si, encara estant la càmera encesa, no es pot accedir a ella perquè està essent utilitzada per una altra aplicació. En aquest cas es mostrarà un missatge adient. Si podem fer servir la càmera, crearem el botó TANCAR CÀMERA per tornar a l'estat anterior.

Una conseqüència de la funció `get_frame` (si es pot rebre la informació de la càmera) és que es se li aplicarà la funció `capta()`.

```
def capta(self, frm):
    self.info = decode(frm)
    cv2.putText(frm, "Posi el codi de barres davant de la camera", (84, 37), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
    if self.info != []:
        self.display.delete('1.0',END)
        for code in self.info:
            if code not in self.codelist:
                self.codelist.append(code)
                self.display.insert(END, (code[0].decode('utf-8'))+'\n')
        self.draw_rectangle(frm)
```

Aquesta funció el que fa és agafar cadascun dels frames generats i els hi anirà aplicant la funció "`decode()`" per descodificar la informació del codi QR. El resultat d'aquesta operació és desat a la variable "info" (que al principi hem definit com a una llista nul·la) amb la informació del QR. Si no es detecta cap QR la llista es posa nul·la. Si per contra es detecta un QR al frame, el neteja la informació que hi havia prèviament a la variable "info" i es guarda la nova informació. En aquest cas es borra el contingut al quadre de text i s'insereix el nou text descodificat. Finalment es crida a la funció `draw_rectangle()` on es dibuixa un rectangle sobre el codi QR directament al canvas.



```

def draw_rectangle(self, frm):
    codes = decode(frm)
    for code in codes:
        data = code.data.decode('ascii')
        x, y, w, h = code.rect.left, code.rect.top, \
            code.rect.width, code.rect.height
        cv2.rectangle(frm, (x,y), (x+w, y+h), (255, 0, 0), 6)
        cv2.putText(frm, code.type, (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 50, 255), 2)

```

Seguirem per descriure la funció `desar`, on entra en joc la funció `conector` que connecta el que es llegeix a l'aplicació amb la base de dades de MySQL.

```

def desar(self):
    if len(self.display.get('1.0',END))>1:

        disp=self.display.get('1.0',END)
        (a,b,c,d,e,f,g)=disp.split()
        d=int(d)
        f=int(f)
        datos=(a,b,c,d,e,f,g)
        conector(datos)
        messagebox.showinfo("DESAT","INFORMACIÓ DESADA A productes_t_ambient")

```

En aquesta funció primerament es mira si la llargada del text al display (caixa de text) és major a 1 (si no, no es guarda la informació). Si la llargada és major a 1, procedim a fer un Split del string (informació que contenia el QR) aquesta informació la enviem a la funció `conector` que explicarem a continuació.

```

import mysql.connector

def conector(datos):
    conexiionl=mysql.connector.connect(host="localhost",
                                     user="root",
                                     passwd="",
                                     database="recepció de mercaderia")

    cursorsl=conexiionl.cursor()
    sql="insert into productes_t_ambient(Data_de_recepció,Proveïdor,Referència_material,Quantitat,UMB,Lot,Data_de_Caducitat) values (%s,%s,%s,%s,%s,%s,%s)"
    cursorsl.execute(sql, datos)
    conexiionl.commit()
    conexiionl.close()

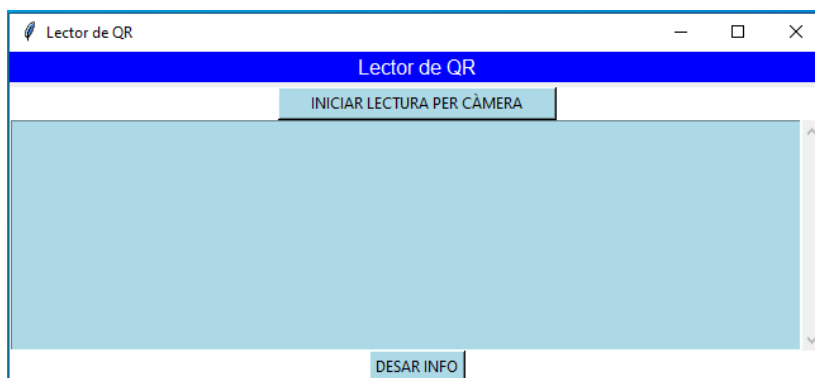
```

Aquí fem la connexió entre el programa de Python i el servidor de MySQL mitjançant el mètode `connect()` de la llibreria `mysql.connector`. Definim el host com a local, el "user" som nosaltres (en aquest cas), sense contrasenya i definim el data base al qual volem connectar-nos. Definim la inserció de dades a la variable `sql` en llenguatge SQL. Seguidament fem servir l'atribut `execute` i el `commit` per tal de formalitzar la càrrega de les dades. Per últim tanquem la connexió.

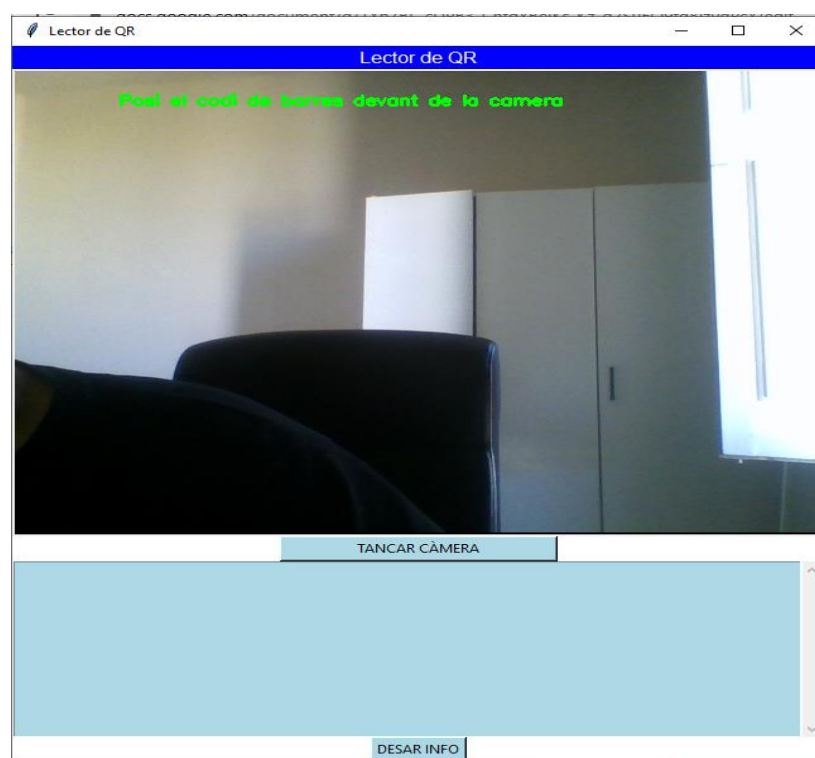
Tornant a la classe App, descrivim les últimes línies de codi. Aquesta última funció serveix per tal de que al finalitzar el programa, tancant la pestanya de la finestra, es finalitzi la càmera si estava en funcionament.

```
def __del__(self):  
    if self.active_camera == True:  
        self.vid.release()
```

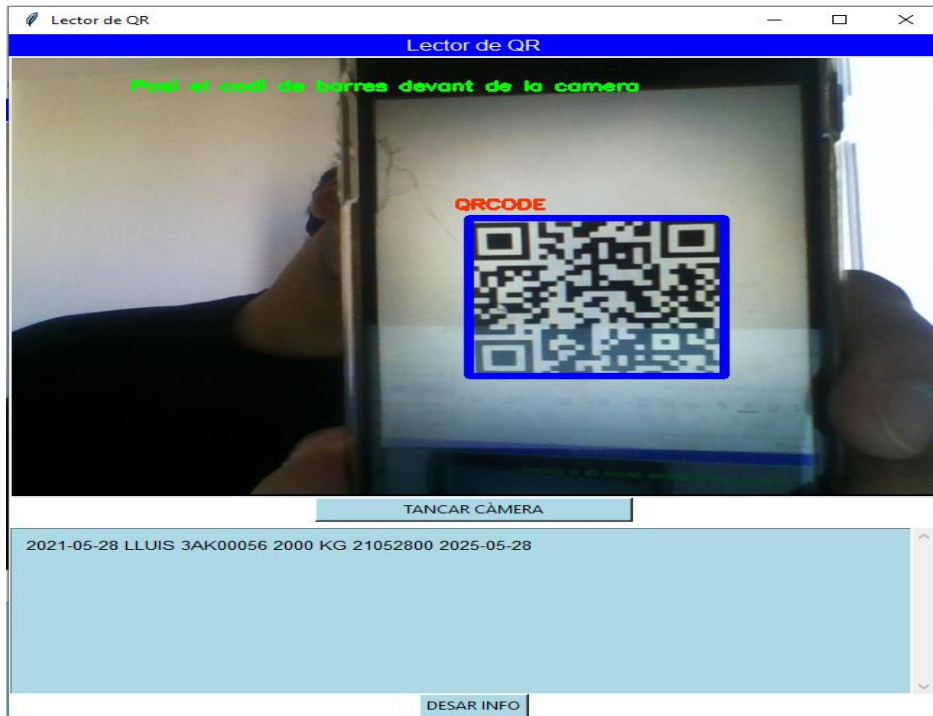
Seguidament es podrà veure l'estètica de la aplicació en els seus diferents estats:



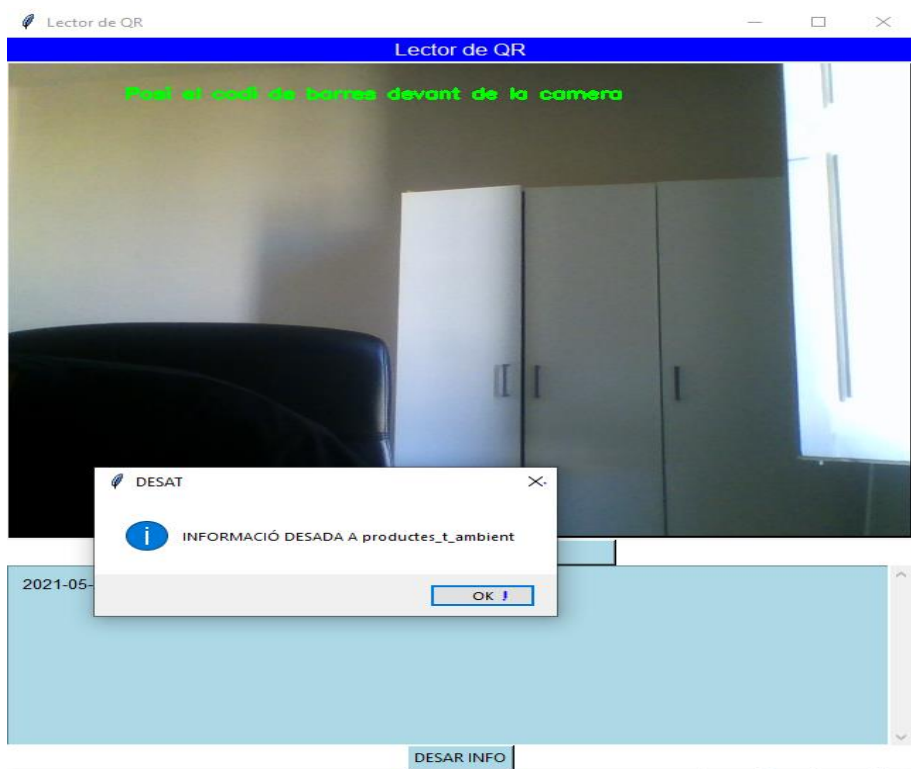
Imatge 1: L'aplicació a l'obrir-se



Imatge 2: L'aplicació al encendre la càmera



Imatge 3: L'aplicació al posar-li un QR davant de la càmera web



Imatge 4: L'aplicació al guardar la informació prèviament llegida

Conclusions

Després de l'anàlisi exhaustiu que he fet tant processos físics com digitals de la cadena d'aprovisionament de la cadena d'aprovisionament de l'empresa model i realitzar el prototip de la digitalització puc dir que he aconseguit l'objectiu proposat al principi del projecte.

Cal remarcar, però, que el prototip no és més que una demostració del que és possible fer amb la tecnologia QR. La aplicació real d'aquesta tecnologia en una empresa haurà de realitzar-se de la mà de professionals en el llenguatge SAP (amb els quals ja estic treballant) per tal de tenir la completa compatibilitat amb el programa.

Sobre aquest aspecte (la implementació a SAP), m'agradaria fer un apunt. En un principi vaig fer un estudi paral·lelament del pas a pas que s'ha de seguir per fer els diferents processos del magatzem (el deixo a l'annex). La meva idea era poder implementar el prototip directament a SAP, però és molt més complicat de fer (per tema de llicències) en una empresa real. Per això, mitjançant aquest prototip amb MySQL, estic tenint reunions amb una consultoria externa per poder implementar el projecte a l'empresa com a tal.

Amb aquest projecte he entès que fer un projecte per a una empresa no és simplement tenir la idea i poder-la implementar, també s'han de tenir en compte les persones afectades pel procés (i la seva adaptabilitat a les noves tecnologies), problemes burocràtics i problemes legals. Al cap i a la fi, he après com gestionar un projecte on es veuen involucrades moltes àrees d'una empresa i, pel que sembla, seguiré endavant amb les diferents solucions plantejades al llarg del projecte.

La carpeta que conté tot el codi ensenyat al treball es pot trobar al següent enllaç:
<https://drive.google.com/drive/folders/1vlyc8l4GzYph7ThwpobQs9e0t6DPFgxl?usp=sharing>

(és necessari tenir correu UPC per tal de poder accedir)



Bibliografia

- [1] TechTerms, “QR Code Definition” 2015. [Online]. Available: https://techterms.com/definition/qr_code.
- [2] Tutoriales Programación ya, “MySQL: Base de datos desde Python”. [Online]. Available: <https://www.tutorialesprogramacionya.com/pythonya/detalleconcepto.php?punto=81&codigo=81&inicio=75>
- [3] W3 Schools, “SQL Tutorial”. [Online]. Available: <https://www.w3schools.com/sql/default.asp>
- [4] W3 Schools, “Python Tutorial”. [Online]. Available: <https://www.w3schools.com/python/default.asp>
- [5] Recursos Python, “Generar código QR”. [Online]. Available: <https://recursospython.com/guias-y-manuales/generar-codigo-qr/>
- [6] W3 Schools, “SQL Working With Dates”. [Online]. Available: https://www.w3schools.com/sql/sql_dates.asp
- [7] El Programador Chapuzas, “Creando aplicación para leer códigos “QR” con la cámara web”. [Online]. Available: <https://programacionpython80889555.wordpress.com/2020/04/28/creando-aplicacion-para-leer-codigos-qr-con-la-camara-web/>
- [8] Python, “Python 3.9.5 documentation” 2021. [Online]. Available: <https://docs.python.org/3/>
- [9] Python, “The Python Package Index (PyPI) is a repository of software for the Python programming language” 2021. [Online]. Available: <https://pypi.org/>