PROJECTE FI DE CARRERA

TÍTOL: **Mòdul Terminal Punt de Venda per a Tryton**

AUTOR: Sergi Castro González

TITULACIÓ: Enginyeria Tècnica d'Informàtica de gestió

DIRECTOR: Jordi Esteve Cousiné

DEPARTAMENT: Llenguatge i sistemes informàtics (LSI)

DATA: Febrer 2013

Aquest Projecte té en compte aspectes mediambientals: □ Sí □ No

PROJECTE FI DE CARRERA

RESUM (màxim 50 línies)

Les aplicacions de planificació de recursos empresarials (en anglès Enterprise Resource Planning), també conegudes per ERPs són sistemes d'informació que s'integren a molts dels negocis actuals per gestionar els seus processos. Tryton és una aplicació ERP d'aquest tipus que permet gestionar diversos aspectes com les vendes, les compres, la facturació la comptabilitat, la gestió d'estoc, la fabricació,...

Aquest projecte consisteix en desenvolupar una aplicació per afegir les funcionalitats d'un terminal punt de venda a un ERP existent com és Tryton. L'aplicació desenvolupada s'encarrega principalment de crear i gestionar vendes al públic o minoristes en un establiment comercial. Permet gestionar el control d'estoc dels productes, el control de caixa, actualitzar la comptabilitat relacionada a les vendes i la impressió de tiquets de venda.

L'aplicació permet obtenir informació detallada sobre les vendes realitzades per saber en quin estat es troben, si ha estat abonada total o parcialment, si s'ha creat la factura i saber quins i quants pagaments s'han realitzat.

Una altre funcionalitat que ofereix l'aplicació és la possibilitat de poder crear operacions per introduir o extreure diners de la caixa, com per exemple retirar diners per ingressar al banc.

Tant les operacions com els pagaments d'una venda s'anoten en un registre des d'on s'actualitza automàticament la comptabilitat de l'empresa així com també consultar els assentaments comptables generats.

Aquest projecte afegeix noves funcionalitats a Tryton i optimitza el processos relacionats a les vendes al detall realitzades en un establiment.

Paraules clau (màxim 10):

TPV	Venda	ERP	Tryton
Registre de Caixa	Caixa	Terminal punt de venda	Moviments de caixa

Índex

1. Introducció 7
1.1. Presentació 7
1.2. Objectius 8
1.3. Motivació 8
1.4. Estat de l'art 9
1.4.1. Arquitectura Tryton 12
1.4.2. Servidor Tryton 13
1.4.3. Client Tryton 15
1.4.4. Llicència 15
1.4.5. Tecnologies utilitzades 16
2. Descripció de l'aplicació 18
2.1. Introducció 18
2.2. Caixa 18
2.3. Registre de caixa 19
2.3.1. Moviments de caixa 19
2.4. Venda 19
2.3.1. Línies de venda 19
2.4.2. Pagament/abonament de la venda 20
2.5. Operacions 20
3. Planificació inicial 21
3.1. Definició de tasques 21
3.1.1. Definició del projecte 21
3.1.2. Desenvolupament del projecte 21
4. Especificació 24
4.1. Glossari de conceptes 24
4.2. Requeriments del sistema 25
4.2.1. Caixa 25
4.2.2. Registre de caixa 25
4.2.3. Moviments de caixa 26
4.2.4. Vendes 27
4.2.5. Línies de venda 28
4.2.6. Pagament/devolució 29

4.2.7. Operacions 30
4.3. Model conceptual 30
4.5. Actors 32
5. Disseny 36
5.1. Casos d'ús i models de col·laboració 36
5.1.1. CU 1 - Crear caixa 36
5.1.2. CU 2 – Crear registre de caixa 38
5.1.3. CU 3 – Obrir registre de caixa 40
5.1.4. CU 4 – Crear moviments de caixa 41
5.1.5. CU 5 – Tancar registre de caixa 45
5.1.6. CU 6 – Confirmar registre de caixa 47
5.1.7. CU 7 – Nova venda 51
5.1.8. CU 8 – Pagar/abonar venda 54
5.1.9. CU 9 - Processar venda 57
5.1.10. CU 10 – Nova operació 59
5.1.11. CU 11 – Introduir operació 61
5.1.12. CU 12 – Configurar TPV 62
5.2. Model de components 65
5.2.1. Model de component 1 65
5.2.3. Mètodes dels components 1 apareguts. 66
5.2.4. Model de components 2 67
5.2.5. Mètodes dels components 2 apareguts 68
5.3. Disseny modular 69
5.3.1. Caixa 72
5.3.2. Registre de caixa 73
5.3.3. Moviments de caixa 74
5.3.4. Venda 75
5.3.5. Línies de venda 76
5.3.6. Pagament/abonament de la venda 76
5.3.7. Operacions 77
5.4. Disseny de la base de dades 78
5.4.1. Taules de la base de dades 79
5.5. Interfícies gràfiques de l'usuari 83
6. Planificació final 91

6.1. Anàlisis dels costos 92
6.2. Comparació entre la planificació inicial i la planificació final 94
7. Possibles millores 96
8. Conclusions 97
9. Bibliografia 98
9.1. Bibliografia complementària 99

Índex de figures

Figura 1: Arquitectura Tryton 12
Figura 2: Arquitectura d'un mòdul Tryton 14
Figura 3: Planificació inicial 22
Figura 4: Model conceptual 1 30
Figura 5: Model conceptual 2 31
Figura 6: DC actors 1 32
Figura 7: DC actors 2 33
Figura 8: CU 1 – new cash 36
Figura 9: CU 1 – end cash 37
Figura 10: CU 1 – crear caixa 37
Figura 11: CU 2 – new_cash_reg 38
Figura 12: CU 2 – end_cash_reg 39
Figura 13: CU 2 – Crear registre de caixa 39
Figura 14: CU 3 – open cash reg 40
Figura 15: CU 3 – open cash reg 41
Figura 16: CU 4 – New cash mov 42
Figura 17: CU 4 – New cash mov sale 43
Figura 18: CU 4 – New cash mov sale amb operació 44
Figura 19: CU 4 – End cash mov 44
Figura 20: CU 4 – New cash mov 45
Figura 21: CU 5 – close_reg_cash 46
Figura 22: CU 5 – close_reg_cash 46
Figura 23: CU 6 – confirm_reg_cash 48
Figura 24: CU 6 – add_move ( ) 49
Figura 25: CU 6 – actualitza_caixa ( ) 49
Figura 26: CU 6 – end_confirm ( ) 50

Figura 27: CU 6 – confirm_reg_cash 51
Figura 28: CU 7 – new_sale 52
Figura 29: CU 7 – add_line 53
Figura 30: CU 7 – end_sale 53
Figura 31: CU 7 – new_sale 53
Figura 32: CU 8 – new_pay 55
Figura 33: CU 8 – pay_sale 55
Figura 34: CU 8 – end_pay 56
Figura 35: CU 8 – new_pay 56
Figura 36: CU 9 – process sale 57
Figura 37: CU 9 – end process 58
Figura 38: CU 9 – process sale 58
Figura 39: CU 10 – new operation 59
Figura 40: CU 10 – end operation 60
Figura 41: CU 10 – new operation 60
Figura 42: CU 11 – introduir operació 61
Figura 43: CU 11 – introduir operació 62
Figura 44: CU 12 – pos configuration 63
Figura 45: CU 12 – afegir informació 63
Figura 46: CU 12 – fi configuració 64
Figura 47: CU 11 – pos configuració 64
Figura 48: Model de components del mòdul account_cash 65
Figura 49: Model de components del mòdul pos 67
Figura 50: Diagrama de la estructura del mòdul pos 69
Figura 51: Diagrama dels mòduls 69
Figura 52: Relació entre les caixes i els registres de caixa 70
Figura 53: Diagrama d'estats del registre de caixa 74
Figura 54: Diagrama d'estats del registre de caixa 76
Figura 55: Estructura de dades del mòdul account_cash 78
Figura 56: Estructura de dades del mòdul pos 78
Figura 57: Llista de caixa 83
Figura 58: Formulari de caixa 83
Figura 59: Llista de registres de caixa 84
Figura 60: Formulari del registre de caixa 84
Figura 61: Apunts comptables generats pel registre de caixa 85

Figura 62: Formulari de la configuració del mòdul pos 85
Figura 63: Llista d'operacions 86
Figura 64: Formulari d'operacions 86
Figura 65: Llista de vendes 87
Figura 66: Formulari de venda 87
Figura 67: Pestanya de pagaments del formulari de venda 88
Figura 68: Pestanya de factures del formulari venda 88
Figura 69: Assistent per pagar una venda 89
Figura 70: Assistent per introduir una operació 89
Figura 71: Planificació final 91
Figura 72: Resum tasques per perfil 93
Figura 73: Comparació entre la planificació inicial i la planificació final 94

I. Projecte

1. Introducció

1.1. Presentació

Les aplicacions de planificació de recursos empresarials (en anglès *Enterprise Resource Planning*), també conegudes per ERPs (1), són sistemes de informació gerencials que integren i utilitzen molts dels negocis per gestionar els seus processos. Els sistemes ERP són aplicacions que permeten gestionar diversos aspectes d'una empresa, com les vendes, compres, logística, facturació, comptabilitat,...

Es caracteritzen per ser modulars i configurables; una empresa és un conjunt de departaments que estan interrelacionats per la informació que comparteixen i que es genera a partir dels seus processos. Un avantatge dels ERPs, és que les seves funcionalitats estan dividides en mòduls, els quals permeten instal·lar-se d'acord amb els requeriments de l'empresa. És configurable gràcies a que es poden desenvolupar altres mòduls i adaptar l'aplicació a les necessitats de cada empresa.

Tryton és una aplicació d'aquest tipus que, entre altres funcions, permet realitzar el procés de venda d'un producte a un client i automatitzar els processos relacionats com la facturació, moviments d'estoc, facturació i comptabilitat. Per tant, és un ERP sobre el que es pot realitzar el desenvolupament d'un terminal punt de venda que agilitzi les vendes al públic.

TPV (2) és el acrònim de terminal punt de venda (en anglès *point of sale* o *POS*). Fa referència al dispositiu i tecnologies que ajuden en les tasques de gestió d'un establiment comercial de venda al públic. Permeten la gestió d'inventaris, el control de caixa, la impressió de tiquets i altres funcions de tràmit comercial.

El mòdul TPV per a Tryton desenvolupat en aquest projecte permet crear i gestionar vendes minoristes (al públic)i gestionar tots els processos relacionats. Tryton ja té un mòdul, anomenat *sale*, que permet crear i gestionar vendes i els seus processos (vendes comercials o majoristes), però existeixen algunes diferències entre el mòdul TPV desenvolupat i el mòdul *sale* existent. Una de les diferències entre les vendes dels dos mòduls és que el flux d'informació de les vendes *sale* és major que el flux de les vendes TPV, aquest últim ha de ser més simplificat. En les vendes *sale*, per exemple, s'ha de preparar una comanda amb els productes que un client desitja comprar mentre que en les vendes TPV no cal preparar la comanda perquè el client ja la fa a l'instant (ja ha recollit els productes dels prestatges de la botiga).

Una altra diferència entre els dos tipus de venda és la manera en que es generen els albarans (moviments d'estoc). En el mòdul desenvolupat en aquest projecte l'albarà es crea automàticament quan s'abona totalment la venda mentre que en les vendes *sale* es genera abans de pagar la venda. Aquest fet és perquè en aquestes últimes prèviament s'ha de preparar al magatzem la comanda per enviar al client. El procés de creació d'una factura de venda minorista també és més simplificada al mòdul TPV perquè la factura es genera quan finalitza el cicle de vida de la venda i a més, es genera en un estat més avançat que en les vendes majoristes. L'última diferència entre els dos mòduls és que els pagaments de les vendes TPV són registrats per tenir constància de cada pagament i poder manipular posteriorment aquesta informació.

1.2. Objectius

Els objectius que desitjo assolir amb aquest projecte es poden dividir en:

- Objectius generals:
    - Desenvolupar un mòdul per a Tryton que permeti gestionar vendes minoristes.
    - Integració del mòdul en l'aplicació.
- Objectius específics:
    - Gestió de vendes d'un establiment comercial amb un terminal punt de venda (TPV).
    - Automatització de l'estoc dels productes del magatzem.
    - Coneixement del flux d'una venda.
- Objectius personals:
    - Aprenentatge del llenguatge de programació Python.
    - Aprenentatge del sistema de programació de l'ERP Tryton.
    - Ús de l'aplicació Tryton.
    - Ús de XML.
    - Utilització i participació en fòrums de la comunitat Tryton.

1.3. Motivació

L'elecció d'aquest projecte està motiva per diversos factors:

- La manca d'un mòdul TPV en l'aplicació Tryton.
- La possibilitat de participar en comunitats sobre aquesta aplicació.
- L'interès que em genera les aplicacions ERP i el desenvolupaments de nous mòduls per aquestes aplicacions.
- Formar-me en tecnologies i sistemes que s'utilitzen en molts àmbits.

1.4. Estat de l'art

Tryton (3) és una aplicació d'alt nivell i propòsit general de tres capes sota la llicència GPL-3 (4). És el nucli base d'una solució empresarial completa que ofereix modularitat, estabilitat i seguretat.

Tryton permet:

- Gestionar i planificar recursos de l'empresa.
- Automatitzar processos.
- Augmentar l'eficiència dels processos eliminant la duplicitat d'informació i de processos innecessaris.
- Compartir informació entre departaments en temps real.

Les característiques de Tryton són:

- Forta modularitat

  Tryton disposa de mòduls per gestionar diversos aspectes del negoci. En l'actualitat les àrees cobertes amb els principals mòduls són:

  - Comptabilitat:

    Està pensat per treballar conjuntament amb els sistemes de vendes i compres de Tryton per evitar duplicar processos en la introducció de dades i per disposar de la informació comptable en temps real. Comporta grans avantatges:

    - Redueix el treball del nostre departament comptable/financer gracies a la reducció de duplicitat i errors.

    - Permet prendre decisions estratègiques en funció dels balanços comptables reals.

  - Facturació:

    Treballa conjuntament amb els sistema de comptabilitat i d'inventari de Tryton. Permet crear una factura a partir d'una venda/compra generar els assentaments comptables adients de forma automatitzada. Genera el document de la factura, permet la possibilitat d'enviar-la per correu

electrònic i d'imprimir, entre altres. També permet conciliar apunts de la factura amb els seus pagaments.

- Gestió de vendes:

  Aquest mòdul realitza el procés de venda d'un producte a un client. Realitza automàticament la creació d'albarans per l'equipament al client, és automatitzat i transparent per a l'usuari així com també el procés de creació de la factura. Per aquest motiu, la gestió de vendes treballa conjuntament amb el mòdul comptable i amb el mòdul de la logística.

- Gestió de compres:

  Semblant al mòdul de gestió de vendes, també treballa juntament amb el mòdul comptable i el mòdul logístic però permet automatitzar processos diferents. Mentre que la gestió de venda crea albarans de sortida del magatzem, aquest mòdul gestiona els albarans dels productes que arriben al magatzem. Permet crear regles per automatitzar les comandes a proveïdors.

- Comptabilitat analítica:

  Amb aquesta funcionalitat es pot mesurar els costos de cada pas de venda i/o producció d'un producte i comparar el resultat amb els costos finals d'aquest producte. D'aquesta manera facilita la comparació entre els dos resultats ajudant a la gestió empresarial.

- Gestió d'estocs (logística):

  Permet automatitzar tant els processos relacionats amb les existències de mercaderies com els moviments que es realitzen dels productes (agrupats en forma d'albarà). Està relacionat amb el mòdul de compres i vendes de productes entre altres.

- Administració d'usuaris:

  Els usuaris s'assignen a grups d'usuaris. Segons el grup o grups a que pertany un usuari tindrà més o menys permisos per gestionar diferents àmbits de l'aplicació.

- Flux (workflow)

  Es poden definir flux (workflow) per estructurar els processos que es van realitzant. Automatitza les seqüències d'accions, activitats o tasques utilitzades per la execució del procés, incloent el seguiment de l'estat de cada una de les seves etapes i l'aportació de les eines necessàries per gestionar-ho.

- Motor d'informes

  Eina que permet que amb poques línies de codi es puguin crear informes de qualsevol objecte de l'aplicació. Això permet tenir un dinamisme enorme sobre la creació d'informes permetent dissenyar-los segons la necessitat.

- Serveis web

  Tryton té integrades diverses funcions relacionades amb els serveis web. Es poden enviar correus electrònics o connectar amb Google Maps.

- Internacionalització

  Facilita la traducció a qualsevol idioma mitjançant un mecanisme que permet adaptar Tryton a un idioma sense la necessitat de modificar el codi programat. L'idioma base per desenvolupar en Tryton és l'anglès.

- Comunitat

  Hi ha grans comunitats de Tryton en molts països (per exemple Mèxic, Bèlgica, França, Alemanya, Catalunya,...) que donen suport sobre qualsevol aspecte de l'aplicació.

- Gestió de clients, proveïdors, empleats i companyies

  - Tercers, són els clients i/o proveïdors que té l'empresa i que realitzen compres o vendes.

  - Companyia, es poden definir una o més empreses i gestionar-les de forma independent (muti companyia).

  - Empleats, són les persones que treballen per tercers o en una companyia.

1.4.1. Arquitectura Tryton

Tryton es basa en una arquitectura de tres capes (model – vista – controlador, MVC): capa de presentació, capa de negoci i capa de dades. L'objectiu és la separació de la lògica de negoci de la lògica de disseny; un exemple bàsic consisteix en separar la capa de dades (base de dades) de la capa de presentació a l'usuari (aplicació).

- **Capa de presentació:** També denominada capa d'usuari. És la interfície gràfica de l'aplicació. Comunica i captura la informació de l'usuari en un procés mínim. Ha de tenir les característiques de ser entendible i fàcil d'usar per l'usuari. Aquesta capa es comunica únicament amb la capa de negoci.

- **Capa de negoci:** És la part on resideix la implementació de les funcionalitats i processos de Tryton juntament amb els arxius de disseny de la capa de presentació. Aquesta capa es comunica amb la capa de presentació per rebre les sol·licituds i presentar els resultats, i amb la capa de dades, per sol·licitar al gestor de base de dades emmagatzemar o recuperar dades.

- **Capa de dades:** Correspon a la capa on resideixen les dades. Esta formada per un gestor de base de dades. Es comunica amb la capa de negoci perquè reben sol·licituds i emmagatzematge o recuperació d'informació.

A continuació es presenta un gràfic que mostra com és l'arquitectura de Tryton:

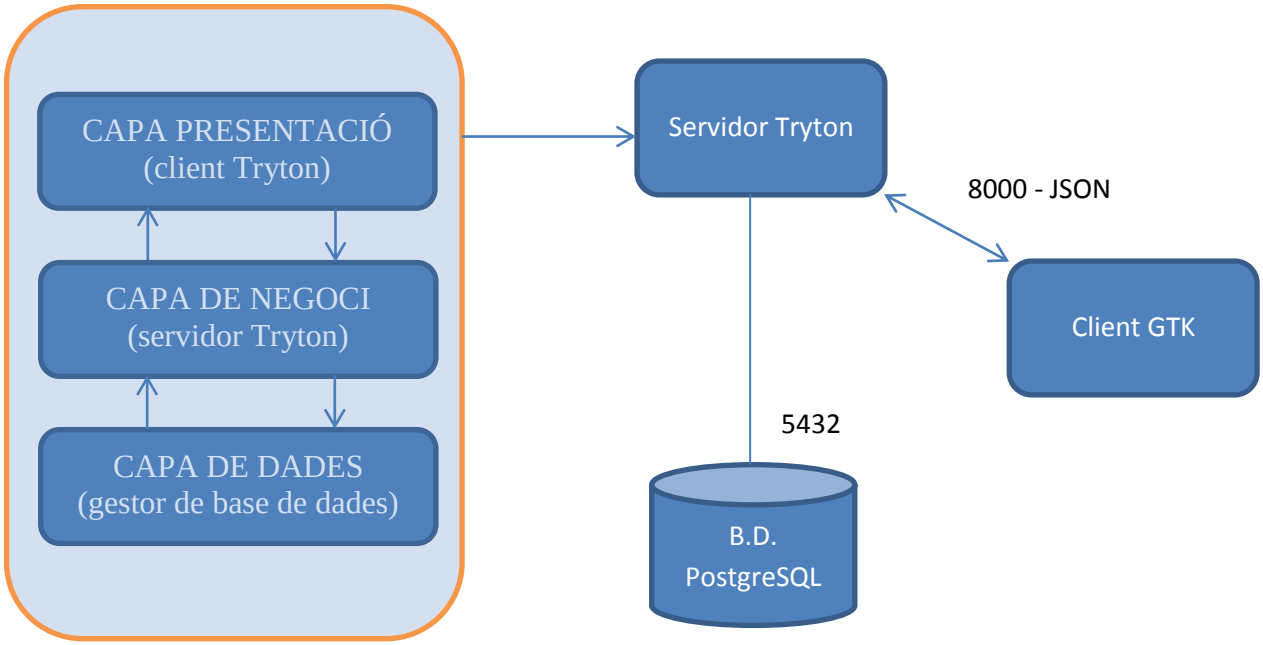


Figura 1: Arquitectura Tryton

1.4.2. Servidor Tryton

També anomenat *core (nucli)*, correspon a la capa de negoci i conté tot el codi que s'executa en l'aplicació. Es comunica constantment amb el gestor de base de dades per extreure dades i enviar-li al client per mostrar-les als formularis de l'aplicació. El servidor inclou una llista de mòduls que es poden instal·lar i proporciona unes eines (*framework*) per desenvolupar-ne de nous.

Les funcionalitats que té el servidor són:

- **Creació de BDD:** Es crea la base de dades fàcilment des d'una interfície del client Tryton.
- **Motor de fluxos (workflows):** Gestiona la creació i utilització de workflows a Tryton.
- **Motor d'informes:** Mecanisme que permet crear informes d'una manera molt dinàmica permetent aprofitar la potència d'un processador de textos com LibreOffice.
- **Multi-idioma:** Suporta traduccions a varis idiomes, entre ells el català i el castellà.

Els tipus d'arxiu que conté un mòdul de Tryton són:

- **Vistes:** Són arxius que defineixen les interfícies de l'aplicació (llistes, formularis, gràfics, calendaris). Mostren la informació emmagatzemada a la base de dades perquè cada vista definida apunta a la taula corresponent de la base de dades. També defineixen la traçabilitat entre formularis, es a dir, la navegació entre formularis.

  Es poden definir expressions per avaluar un conjunt de registres i mostrar a l'usuari només els que compleixin les condicions establertes. Per exemple, que només mostri les vendes que es troben en un estat concret.

- **Models i capa de negoci:** S'encarreguen de crear la taula corresponent a la base de dades i de comunicar-s'hi amb aquesta. També es defineixen els atributs de la classe, poden ser de diferents tipus: enter, text, numèric, relacionats (un a molts, molts a molts, ...).

  De la mateixa manera que a les vistes, es poden definir expressions per avaluar un conjunt de registres i mostrar només els que compleixin les condicions establertes.

- **Arxiu configuració Tryton:** Arxiu que conté les dependències a altres mòduls que cal satisfer per poder executar les seves funcionalitats. També conté els nom dels arxius de les vistes. Amb aquesta informació, quan el servidor instal·la un mòdul per primer cop, el primer que fa és instal·lar totes les dependències definides.

- **Arxiu configuració mòdul:** Conté tota la informació referent a traduccions a altres idiomes, qui ha creat el mòdul, la versió del mòdul i la versió de l'aplicació entre altres.

- **Arxiu inicialització del mòdul:** Llistat dels noms de totes les classes, de tots els assistents i tots els informes del mòdul. Permet que, quan es posa en funcionament el servidor, registri totes les classes del mòdul per poder utilitzar-les durant l'execució de l'aplicació.

- **Informes:** Arxius de tipus *.odt* que conté el disseny del(s) informe(s) del mòdul (OpenOffice, LibreOffice).

A continuació es presenta un gràfic que mostra l'arquitectura que té un mòdul al servidor. S'ha de tenir en compte que el servidor té molts mòduls i que cadascun té com a mínim aquests arxius.

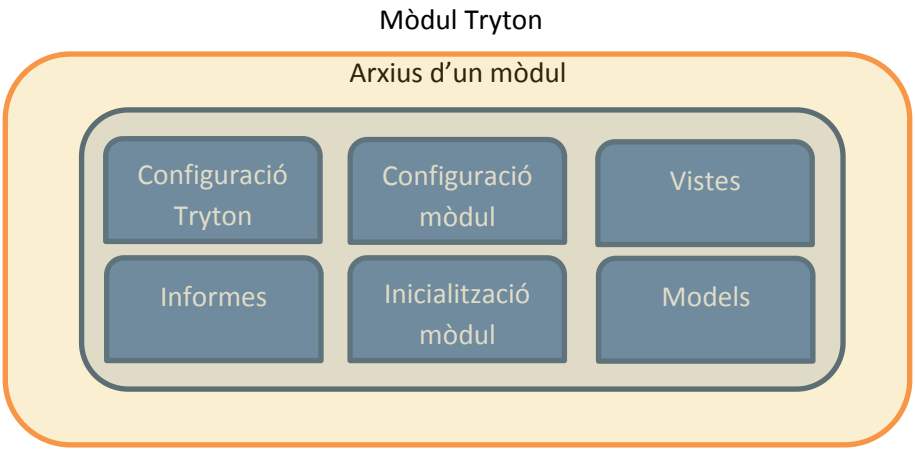


Figura 2: Arquitectura d'un mòdul Tryton

La seguretat de la comunicació de Tryton i el servidor basa en:

- **SSL:** *Secure SocketsLayer*, protocol criptogràfic que proporciona comunicacions segures per internet. SSL proporciona l'autentificació i privacitat de la informació entre extrems sobre Internet mitjançant l'ús de criptografia. Habitualment només el servidor és autentificat (es garantitza la seva identitat), mentre que el client es manté sense autentificar.

- **Autentificació:** S'encarrega de comprovar si l'usuari que es connecta a la base de dades és vàlid.

- **Regles d'accés:** Regles que defineixen els permisos que disposen els usuaris sobre Tryton.

1.4.3. Client Tryton

Correspon a la capa de presentació; és la interfície gràfica de l'aplicació. S'encarrega de mostrar i captar les dades que consulta o introdueix l'usuari. El client Tryton pot comunicar-se amb el servidor remotament. Per tant, el client pot estar instal·lat en un ordinador diferent a l'ordinador en el que està instal·lat el servidor. Conté els arxius sobre la versió i els arxius amb les traduccions a altres idiomes de l'aplicació, entre altres. El client és lleuger i també multi-plataforma; té versions d'Ubuntu, Windows i Mac Os encara que per desenvolupar un nou mòdul el sistema operatiu òptim és Ubuntu. La comunicació en xarxa amb el servidor es realitza a través dels següents protocols:

- **XML – RPC:** (5)És un protocol que crida a un procediment remot que utilitza XML per codificar les dades i el protocol HTTP per la transmissió de missatges.

- **JSON – RPC:** (6)D'igual manera que el protocol anterior, aquest crida a un procediment remot, però en aquest cas s' utilitza JSON per realitzar la codificació dels missatges.

1.4.4. Llicència

Aquest projecte l'he pogut desenvolupar gràcies a la llicència que té Tryton. És una aplicació que es distribueix sota la llicència GPL-3.

Aquesta llicència permet tenir llibertat d'usar l'aplicació, llibertat d'estudiar el funcionament de l'aplicació i adaptar-la a les nostres necessitats, llibertat de distribuir còpies de l'aplicació, llibertat de millorar l'aplicació i de fer públiques les millores.

La llicència GPL-3 no obliga a publicar el treball que he realitzat però, en cas que ho publiqui, si que he de posar en disposició dels usuaris el codi de l'aplicació desenvolupada i alliberar-la amb la mateixa llicència GPL-3.

1.4.5. Tecnologies utilitzades

El sistema operatiu utilitzat durant el projecte és Ubuntu 12.04. És un sistema operatiu lliure que permet disposar gran quantitat de software lliure. La versió de Tryton on s'ha realitzat el desenvolupament del projecte és la versió 2.6, és l'última versió estable publicada de l'aplicació i disponible a la pàgina web oficial.

A continuació explico les tecnologies que s'utilitzen per desenvolupar nous mòduls a Tryton:

- **Python:** (7) Conegut llenguatge de programació que té la filosofia de fer un codi net i llegible. És un llenguatge de programació interpretat i orientat a objectes. És multi plataforma, per tant es utilitzar aquest mòdul de Tryton, per exemple, a Windows.
- **XML:** (8) És un llenguatge de marques derivat del llenguatge SGML. Permet definir la gramàtica dels llenguatges específics per estructurar documents grans. A més XML dona suport a base de dades. Permet intercanviar informació estructurada entre diferents plataformes de manera segura, fiable i fàcil. Es molt senzill definir un arxiu amb estructura XML (només cal un editor de text pla).
- **Framework Tryton:** Un framework és una eina que facilita el desenvolupament de software, permetent no perdre temps en detalls dels llenguatges de programació com, per exemple, programar la vista d'una finestra sense generar cap error.

  El framework de Tryton permet,

  - Dissenyar les interfícies gràfiques a través d'arxius XML i arxius de Python. En aquests últims es defineixen els camps que tindrà la interfície gràfica i el tipus de dada (numèric, text, etc) que es guardarà a la base de dades.
  - Realitzar transaccions amb la base de dades per recuperar informació i operar amb aquesta.
  - Relacionar registres entre sí provinents de diferents taules de la base de dades.
  - Recuperar i emmagatzemar les dades de les interfícies gràfiques de l'aplicació.
  - Creació de taules i dels seus camps a la base de dades.
  - Creació d'informes i workflows.
  - Creació de assistents (en anglès *wizards*).

- **PostgreSQL:** (9) Motor de base de dades relacional. Té tolerància amb concurrència alta perquè inclou un sistema que, en cas de que un usuari accedeixi a modificar una dada d'una taula mentre un altre usuari està modificant algun valor d'aquesta mateixa taula no es bloquegi la base de dades. Permet herència de taules, integritat transaccíonal, suporta el llenguatge SQL i es pot integrar en entorns de treball amb altres gestors de base de dades.

- **Libre Office Writer:** (10) Processador de text de distribució lliure que permet dissenyar informes per utilitzar a Tryton. Es poden aprofitar les seves característiques per dissenyar un informe amb qualsevol aparença. Amb poques línies de codi es pot recuperar informació dels registres amb els que s'està realitzant alguna operació a Tryton en aquell mateix moment per tal de generar un informe amb les seves dades.

2. Descripció de l'aplicació

2.1. Introducció

El mòdul TPV sobre Tryton es basa en l'automatització del procés de venda minorista des del moment de la seva creació en el sistema fins la finalització d'aquest procés. Es precisa desenvolupar un mòdul sobre un entorn ja fet com és Tryton.

Les funcionalitats que el mòdul ha d'oferir són:

- Crear i gestionar vendes de manera ràpida i senzilla.
- Crear i gestionar altres tipus d'operacions relaciones amb la entrada i sortida de diners.
- Control i gestió de la caixa.
- Creació automàtica d'albarans.
- Creació automàtica de factures.
- Actualització de l'estoc dels productes.
- Actualització de la comptabilitat de l'empresa.

La idea principal va ser desenvolupar un mòdul encarregat de enregistrar vendes i realitzar els processos derivats automàticament. Per aconseguir afegir aquesta funcionalitat al mòdul es necessita un component que enregistri tots els moviments de diners (rep el nom de registre de caixa), un altre component que guardi el saldo de diners, (rep el nom de caixa), les vendes que es creen al sistema i que recullen tota la informació important, per últim, operacions amb diners d'origen diferent a les vendes com, per exemple, el pagament del servei de neteja.

2.2. Caixa

Es necessita un element que permeti guardar el saldo total acumulat a través de tots els moviments de caixa que han sigut registrats per un registre de caixa. Es necessita que el saldo emmagatzemat a la caixa s'utilitzi per calcular el saldo inicial del registre de caixa quan s'inicialitza.

Cada cop que el registre de caixa finalitzi el seu propòsit, es necessita que s'actualitzi el saldo de la caixa. Així, el pròxim cop que s'inicialitzi un registre de caixa amb una caixa que ja s'ha utilitzat anteriorment, el seu saldo d'obertura serà igual al saldo guardat per la caixa.

2.3. Registre de caixa

El registre de caixa és un element que s'utilitza per enregistrar operacions de diners en un període de temps donat. Es necessita que el registre de caixa disposi de diferents estats per poder processar la informació disponible. Per aquest motiu es necessita definir un workflow sobre el registre de caixa on cada pas del workflow està associat a un estat diferent.

El registre de caixa haurà de calcular el saldo de tancament (a través del saldo de cada moviment de caixa) per fer que l'usuari introdueixi un saldo manualment. Es necessita comprovar que el saldo de tancament calculat i el saldo introduït per l'usuari quadri.

Es necessita que el registre de caixa actualitzi la comptabilitat de l'empresa creant al sistema un assentament comptable amb dos apunts comptables per cada moviment de caixa.

2.3.1. Moviments de caixa

Com he comentat anteriorment, el registre de caixa s'ocupa d'enregistrar moviments de diners a la caixa. Per aquest motiu, el registre de caixa contindrà un conjunt de moviments de caixa. Es necessita que cada moviment de caixa contingui informació detallada sobre cada operació de diners enregistrada.

La informació més important que es necessita incloure en cada moviment de caixa és la data de realització, un saldo i un compte comptable.

2.4. Venda

Una venda és el procés per el qual el venedor satisfà les necessitats d'un comprador a canvi d'una determinada quantitat de diners. Per automatitzar aquest procés es necessari definir un workflow associat a la venda. L'objectiu és automatitzar tots els processos necessaris per la realització d'una venda des del moment de la creació fins la finalització d'aquesta. La venda ha de poder contenir un conjunt de línies de venda.

També es necessita que per cada venda minorista es generi la seva factura corresponent així com el seu tiquet de venda.

2.3.1. Línies de venda

Es necessita que una línia de venda es composi, coma mínim, per un producte, el número d'unitats del producte, impostos relacionats, preu unitari i cost total. Amb aquesta informació es podrà calcular el cost total de la venda. Es necessari que les línies de venda continguin la informació necessària per poder actualitzar l'estoc dels productes.

2.4.2. Pagament/abonament de la venda

Es necessita un mecanisme per poder pagar/abonar una venda minorista. Un pagament/abonament haurà de ser anotat per un registre de caixa.

Per saber si una venda s'ha pagat/abonat es necessita que cada cop que es realitza un pagament s'actualitza un camp de la venda que indica l'import que encara manca per pagar. També es necessita que, al finalitzar el pagament/abonament d'una venda minorista s'actualitzi l'estoc dels productes.

2.5. Operacions

Es necessita un element per realitzar operacions esporàdiques de diners i que les enregistri un registre de caixa.

Les operacions seran de dos tipus i han de poder crear moviments de caixa tant positius com negatius.

3. Planificació inicial

3.1. Definició de tasques

3.1.1. Definició del projecte

Es tractaria de parlar amb el destinatari de l'aplicació, en el meu cas el tutor del projecte i fixar els objectius del projecte i que és el que s'espera d'aquest. Quan s'ha definit l'objectiu del projecte es realitza la presa de requeriments.

Els requeriments són funcions que ha de tenir l'aplicació per poder complir els objectius definits anteriorment. Per acabar de definir el projecte es fa una previsió dels temps per la realització de cada tasca.

3.1.2. Desenvolupament del projecte

- **Anàlisis:** Gairebé totes les tasques d'anàlisis es realitzen alhora degut a que una petita modificació en els casos d'ús afecta directament al model conceptual, i a la inversa també.
    - **Model conceptual:** Es defineix un diagrama dels conceptes més importants del problema que es desitja resoldre i la manera en que s'associen.
    - **Casos d'ús:** Es tracta de definir la funcionalitat que ha de resoldre cada ús de l'aplicació.
    - **Diagrames de seqüència:** Defineix la seqüència d'accions referents als casos d'ús definits anteriorment.
- **Disseny:** Igual que en l'anàlisi, en la fase de disseny totes les tasques es realitzen simultàniament. Això simplifica cada tasca perquè un petit canvi en un diagrama de components afecta directament al model de components.
    - **Diagrames de components:** Es creen els diagrames que mostren com es realitza cada tasca definida als casos d'ús.
    - **Model de components:** Model que mostra totes les visibilitats entre components d'un mateix cas d'ús.
    - **Disseny de la base de dades:** Disseny de la base de dades amb les taules més importants, incloent les claus principals i les claus foranies.
- **Implementació:** Està formada per tota la implementació del mòdul.
    - **Disseny de la interfície:** Es dissenya l'aparença de la interfície gràfica del mòdul. S'ha de respectar el conveni de programació de Tryton.

- **Codificació:** Es tracta d'implementar totes les funcionalitats que ha de tenir el mòdul per complir l'objectiu definit.

- **Control de qualitat:** Correspon a tota la part que s'encarrega de testejar les funcionalitats del mòdul per assegurar el perfecte funcionament d'aquest.

  - **Comprovació d'errors:** S'inicià durant la fase de codificació ja que és molt important anar solucionant els problemes. Si es deixen per al final del projecte s'hauria de reestructurar l'aplicació per poder corregir els errors.

  - **Procés d'usuari:** Proves realitzades per altres usuaris; proven les funcionalitats de l'aplicació per comprovar el seu correcte funcionament i, si cal, modificar algun aspecte visual per ajustar-ho més a les seves necessitats.

- **Documentació:**

  - **Documentació de l'aplicació:** Comentaris en el codi de la implementació, explicant el funcionament de les funcions. Hi ha parts incloses en la memòria explicant decisions preses durant la fase d'anàlisi i disseny.

  - **Manual de l'administrador:** Documentació referent a l'ús de l'aplicació enfocat als usuaris finals.

  - **Redacció de la memòria:** Redacció de la memòria del projecte final de carrera. Inclou parts de l'anàlisi i disseny del mòdul

A continuació es mostra el diagrama de Gantt corresponent a les tasques anteriors:

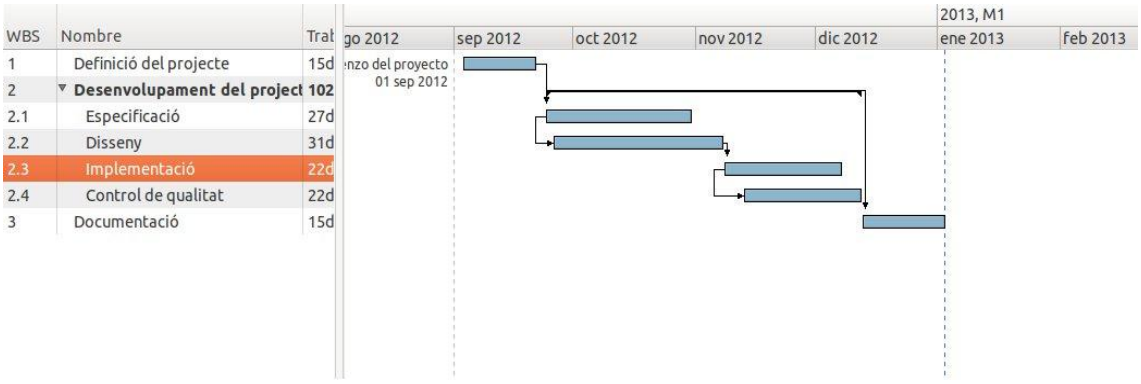


Figura 3: Planificació inicial

La planificació dóna un total de 102 dies. Començant el dia 1 de setembre s'acabaria el dia 11 de desembre.

II. Especificació

4. Especificació

4.1. Glossari de conceptes

Usuari: Persona que utilitza el sistema o aplicació.

Producte: Objecte exposat per a la venda al públic que s'intercanvia per diners. Poden ser objectes materials o també serveis. Un servei són tasques que l'empresa ofereix als clients a canvi de diners.

Client: Entitat o persona que pot ser coneguda o anònima i que adquireix productes a canvi d'una quantitat de diners.

Venda: Intercanvi que fa un client d'un conjunt de productes per una quantitat de diners.

Línia de venda: Formada per un producte, les unitats del producte, el preu de venda del producte i el sub-total (número d'unitats * preu del producte).

Caixa: Lloc físic o virtual on es guarden els diners. Per exemple: Caixa efectiu, targeta de crèdit, ...

Registre de caixa: Ús d'una caixa durant un interval de temps en una data donada. Enregistra moviments de caixa. Té un saldo inicial i saldo de tancament. El saldo inicial són els diners amb els que s'inicialitza la caixa i el saldo de tancament és la suma del saldo inicial + operació d'entrada – operacions de sortida

Moviment de caixa: És un moviment de diners que s'ha realitzat durant la utilització de l'aplicació i s'anota en un registre de caixa.

Moviment comptable: Assentament comptable associat a un moviment de caixa. Conté dos apunts o línies comptables.

Pagament/devolució: Entrada/sortida de diners que enregistra el registre de caixa com a conseqüència del pagament o devolució total/parcial d'una venda.

Operació IN/OUT: Entrada i sortida especials de diners que enregistra el registre de caixa. Per exemple pagament personal neteja.

4.2. Requeriments del sistema

4.2.1. Caixa

A continuació es mostren els requeriments corresponents de la caixa (CSH):

Codi	Descripció
CSH – 1	Per crear una caixa al sistema són necessàries les següents dades, el nom de la caixa, un diari comptable, l'empresa i la moneda que utilitza la companyia.
CSH – 2	El saldo per defecte és zero fins que no s'actualitzi.
CSH – 3	El saldo de la caixa només pot ser modificat a través del registre de caixa.
CSH – 4	El diari comptable de la caixa ha de tenir associat un compte comptable pel debe i un altre compte per l'haver
CSH – 5	Totes les dades que s'han d'introduir per crear una caixa són requerides.
CSH – 6	Només els usuaris amb els privilegis adients poden crear caixes.
CSH – 7	Es poden crear tantes caixes com calgui.
CSH – 8	Les caixes poden ser utilitzades per tots els usuaris.

4.2.2. Registre de caixa

A continuació es mostren els requeriments corresponents al registre de caixa (RG):

Codi	Descripció
RG – 1	Per crear un registre de caixa al sistema són necessàries les següents dades, nom, l'usuari que el crea, el període que utilitza i la caixa amb la que s'associa el registre de caixa.
RG – 2	El registre de caixa té quatre estats, esborrany, obert, tancat i confirmat.
RG – 3	Mentre el registre de caixa es trobi en estat esborrany no es poden registrar moviments de caixa.
RG – 4	Mentre el registre de caixa es trobi en estat esborrany es poden modificar les dades introduïdes quan s'ha creat al sistema.
RG – 5	El registre de caixa, si està en estat esborrany, podrà adoptar l'estat obert.
RG – 6	Si el registre de la caixa està en estat obert podrà enregistrar moviments de caixa.
RG – 7	Quan s'obre el registre de caixa, el saldo d'obertura és igual al saldo guardat en la caixa.
RG – 8	El registre de caixa tindrà constància de la data en que s'obre.
RG – 9	Quan s'obre el registre de caixa no es podrà modificar cap informació.
RG – 10	Si l'usuari no ha introduït cap nom pel registre quan hi estava en estat esborrany, quan s'obre el registre es generarà un nom automàticament.
RG – 11	Si l'usuari ha introduït un nom per al registre quan esta en estat draft, quan s'obre el registre, ho farà amb el nom introduït.

RG – 12	L'usuari podrà introduir moviments de caixa manualment.
RG – 13	El registre de caixa, si està en estat obert pot tornar a l'estat esborrany.
RG – 14	El registre de caixa, si està en estat open pot adoptar l'estat tancat.
RG – 15	Quan es tanca el registre de caixa el sistema calcula el saldo final corresponent a la suma del valor de tots els moviments de caixa.
RG – 16	El registre de caixa ha de tenir constància de la data de tancament.
RG – 17	Per poder adoptar l'estat confirmed l'usuari haurà d'introduir en el registre de caixa el saldo de tancament.
RG – 18	Si el registre de caixa està en estat close i l'usuari introdueix un saldo de tancament erroni el sistema mostrarà un missatge d'error.
RG – 19	El registre de caixa si està en estat tancat pot tornar a l'estat obert.
RG – 20	El registre de caixa si està en estat tancat pot adoptar l'estat confirmat.
RG – 21	En estat confirmat, no es pot modificar cap informació del registre de caixa.
RG – 22	En estat confirmat, el registre de caixa no pot registrar moviments de caixa.
RG – 23	Si el saldo de tancament introduït per l'usuari és igual al saldo final generat pel registre de caixa es donaran d'alta al sistema un assentament comptable amb dos apunts comptables per cada moviment de caixa. El registre de caixa adoptarà l'estat *confirmat.*.
RG – 24	Si el saldo de tancament introduït per l'usuari és igual al saldo final generat pel registre de caixa i aquest no té cap moviment de caixa registrat, el registre de caixa adopta l'estat confirmed sense crear cap assentament comptable.
RG – 25	Quan es confirmen els moviments de caixa registrats pel registre de caixa, si ja s'ha generat l'assentament comptable amb dos apunts comptables sobre algun moviment de caixa, no es tornen a generar.
RG – 26	Quan es confirmen els moviments de caixa que té anotats el registre de caixa, s'actualitzarà el saldo inicial de la caixa utilitzada.
RG – 27	Només es generarà un assentament comptable amb dos apunts comptables d'aquells moviments de caixa registrats al registre de caixa i que es trobin a l'estat esborrany.
RG – 28	Si el registre de caixa ja ha confirmat els moviments de caixa, aquests adoptaran l'estat *confirmat*.
RG – 29	Si el registre de caixa té l'estat confirmat, l'usuari pot tornar a obrir-lo.

4.2.3. Moviments de caixa

A continuació es mostren els requeriments corresponents als moviments de caixa (MV):

Codi	Descripció
MV – 1	Per crear un moviment de caixa al sistema cal que el registre de caixa estigui obert.
MV – 2	Per crear un moviment de caixa al sistema cal introduir tota la informació necessària.
MV – 3	De tota la informació només es requereix l'introducció d'un import i un compte comptable.
MV – 4	Els moviments de caixa tenen dos estats, esborrany i confirmat.

MV – 5	Els moviments amb estat esborrany són moviments de caixa que encara no han estat comptabilitzats.
MV – 6	Els moviments que tenen l'estat confirmed són els moviments de caixa que ja han estat comptabilitzats.
MV – 7	Els moviments de caixa podran ser generats per pagaments/abonaments, operacions d'entrada i sortida o manualment des del registre de caixa.
MV – 8	Els moviments de caixa poder ser amb imports negatius.
MV – 9	Els moviments de caixa amb saldo negatiu també generarà un assentament comptable amb dos apunts comptables.
MV – 10	Els moviments de caixa passen de l'estat esborrany a l'estat confirmat quan han generat l'assentament comptable amb dos apunts comptables.

4.2.4. Vendes

A continuació es mostren els requeriments corresponents a vendes (SAL):

Codi	Descripció
SAL – 1	Per crear al sistema una venda minorista cal especificar l'usuari que la crea, la moneda en que es realitza la venda, el tercer que genera la venda i la data en que es realitza.
SAL – 2	La venda també disposa d'un camp on es mostra el cost total sense impostos, un altre camp on es mostra el cost total de l'impost aplicat i un altre que mostra el cost total amb impostos inclosos.
SAL – 3	També té un camp que mostra el total a pagar i, un altre que mostra el la quantitat que s'ha pagat.
SAL – 4	La venda minorista pot adoptar tres estats, esborrany, confirmada, processada.
SAL – 5	La venda es crea al sistema amb l'estat esborrany per defecte.
SAL – 6	Si s'ha configurat, el client de la venda minorista es mostrarà per defecte.
SAL – 7	La venda minorista té constància de la data de creació en el sistema.
SAL – 8	Mentre la venda estigui en l'estat esborrany es poden modificar algunes dades.
SAL – 9	Mentre la venda minorista es troba en l'estat esborrany es poden afegir línies de venda.
SAL – 10	Es podran afegir tantes línies de venda com calgui.
SAL – 11	Poden haver línies de venda iguals.
SAL – 12	Cada cop que s'introdueix una nova línia de venda s'actualitzaran tots els camps que mostren els costos així com també el total a pagar.
SAL – 13	Si la venda té l'estat esborrany es pot iniciar el procés de pagament.
SAL – 14	Si el total a pagar és igual a zero la venda minorista passarà de l'estat esborrany a l'estat confirmada.
SAL – 15	Mentre la venda minorista tingui l'estat confirmada no es podrà realitzar cap acció.
SAL – 16	Si la venda minorista té l'estat confirmada podrà adoptar l'estat

	processada.
SAL – 17	Per a que la venda minorista adopti l'estat processada s'ha de crear la factura corresponent.
SAL – 18	Una venda minorista només pot crear una factura.
SAL – 19	Quan s'ha creat la factura associada a la venda minorista, aquesta adopta l'estat processed.
SAL – 20	Quan la venda minorista té l'estat processada no es pot modificar cap dada.
SAL – 21	El procés de la venda minorista finalitza quan es processa.
SAL – 22	Si existeix un conjunt de vendes minoristes amb el mateix tercer associat i totes aquestes vendes minoristes es troben en l'estat confirmed, l'usuari pot crear una sola factura.
SAL – 23	Si es genera una sola factura d'un conjunt de vendes minoristes, aquestes adoptaran l'estat processada.
SAL – 24	Si un conjunt de vendes minoristes tenen diferents tercers i s'intenta generar una sola factura el sistema mostrarà un missatge.

4.2.5. Línies de venda

A continuació es mostren els requeriments corresponents a vendes (LN):

Codi	Descripció
LN – 1	Per crear una línia de venda al sistema cal introduir, un producte, la unitat de medita, la descripció, el preu de cada unitat, el número d'unitats i els impostos (si el producte té).
LN – 2	Quan s'inclou un producte en una línia de venda minorista, automàticament s'actualitza els costos de la venda minorista i tota la informació inclosa en la línia de venda.
LN – 3	Només pot haver un producte en una línia de venda.
LN – 4	Si es modifica el camp quantitat de la línia de venda, es re-calculen tots els camps relacionats amb el cost de la venda.
LN – 5	La introducció d'un producte es obligatòria.
LN – 6	Les línies de venda només es poden crear al sistema si la venda minorista associada té l'estat draft.

4.2.6. Pagament/devolució

A continuació es mostren els requeriments corresponents als pagaments/devolució (PG):

Codi	Descripció
PG – 1	El pagament només es pot fer des d'una venda minorista en estat esborrany.
PG – 2	Si s'intenta realitzar un pagament des d'una venda minorista en estat esborrany i aquesta no té cap línia de venda, el sistema mostrarà un missatge d'advertència.
PG – 3	Per realitzar un pagament o abonament, ha d'existir al sistema al menys un registre de caixa que tingui l'estat obert.
PG – 4	Una venda minorista es pot abonar mitjançant un pagament total o parcial.
PG – 5	Quan s'inicià el pagament d'una venda minorista, l'usuari pot seleccionar en quin registre de caixa vol que es registri el pagament.
PG – 6	L'usuari pot fer diferents pagaments i anotar-los en diferents registres de caixa.
PG – 7	Es tindrà constància de la data en que s'ha realitzat cada pagament/abonament.
PG – 8	Quan es realitza una pagament s'actualitzen el total a pagar i el total pagat de la venda minorista.
PG – 9	Si es realitza un pagament superior a la quantitat a pagar, l'import que es mostra per pagar és negatiu. Correspon al retorn del canvi.
PG – 10	Tots els pagaments que es realitzin s'anotaran a un registre de caixa
PG – 11	Un pagament negatiu també s'anota al registre de caixa.
PG – 12	El pagament d'una venda minorista serà complert quan el total a pagar sigui igual a zero.
PG – 13	Quan el total a pagat és igual a zero es crea al sistema un nou albarà associat a la venda minorista.
PG – 14	En el moment que el total a pagar és igual a zero, la venda minorista adopta l'estat confirmada.
PG – 15	Es pot interrompre el procés de pagament d'una venda minorista.
PG – 16	Es pot reanudar el procés de pagament d'una venda minorista en el mateix punt on es va interrompre.

4.2.7. Operacions

A continuació es mostren els requeriments corresponents a les operacions (OP):

Codi	Descripció
OP – 1	Per crear una operació al sistema cal introduir el nom, el compte comptable que l'usuari desitgi associar a la caixa i el tipus d'operació que és (entrada o sortida).
OP – 2	Per registrar els moviments de caixa que generen les operacions d'entrada i sortida es requereix al menys un registre de caixa amb estat open.
OP – 3	Les operacions d'entrada generen un moviment de caixa positiu en un registre de caixa.
OP – 4	Les operacions de sortida generen un moviment de caixa negatiu en un registre de caixa.
OP – 5	A l'hora de realitzar un operació d'entrada i sortida es pot seleccionar qualsevol registre de caixa que tingui l'estat obert.
OP – 6	S'anota la data en què es realitza l'operació d'entrada o sortida.
OP – 7	Els moviments de caixa es generen associant el compte comptable associat a cada operació.

4.3. Model conceptual

Per fer més entendible les associacions entre conceptes existents he dividit el model conceptual en dos parts diferents. A continuació es mostra el model conceptual 1. Representa el flux per registrar moviments de caixa i d'actualització la comptabilitat de l'empresa.

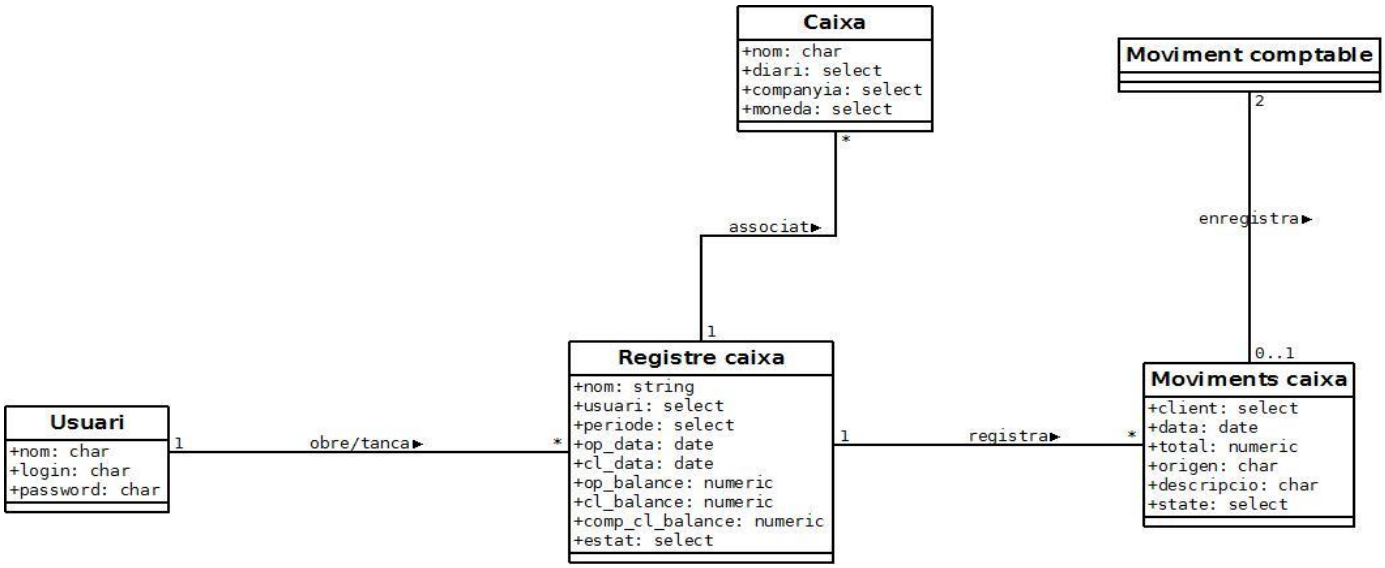


Figura 4: Model conceptual 1

A continuació es mostra el model de conceptual 2 relacionat amb el flux de les vendes TPV i el seu pagament. Hi ha conceptes que no s'han especificat perquè ja es estan desenvolupats a l'aplicació Tryton.

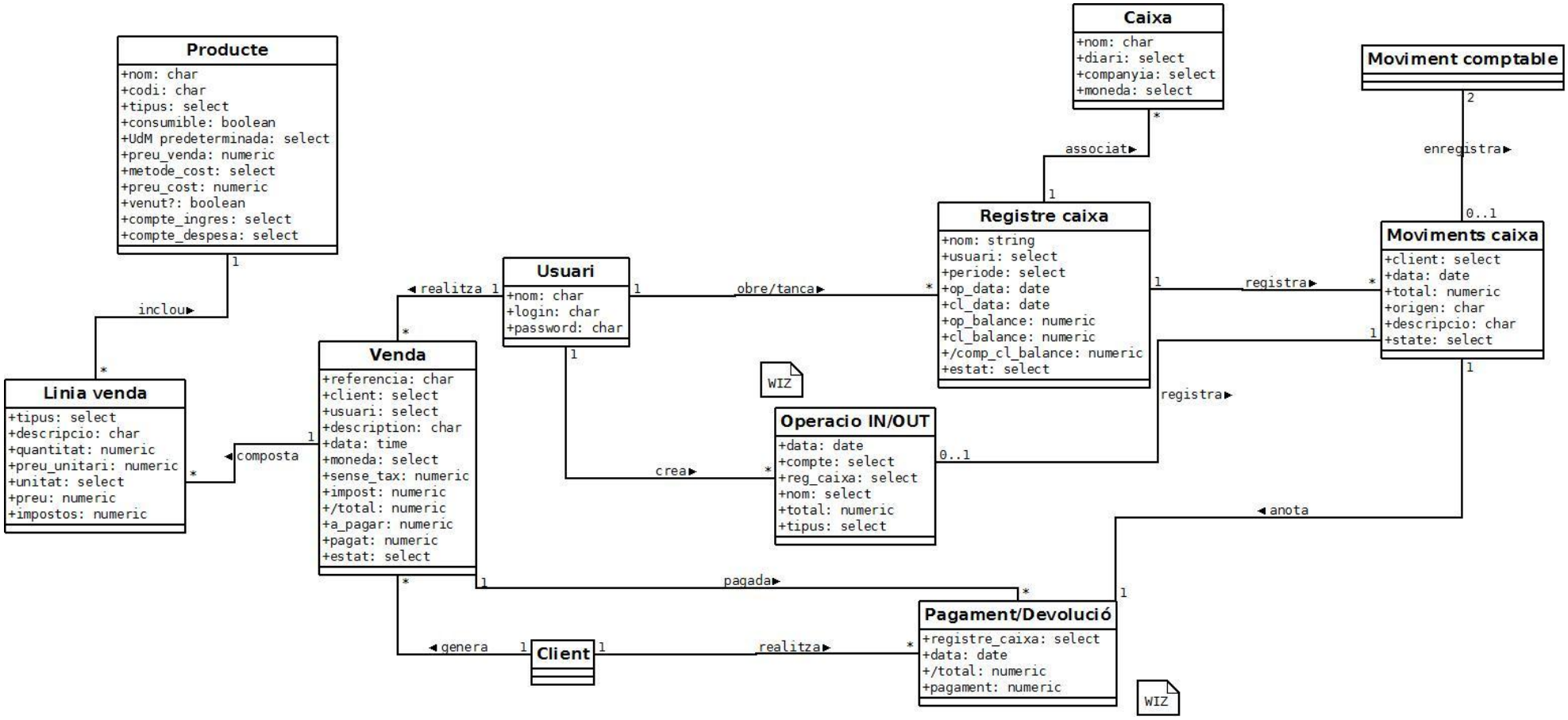


Figura 5: Model conceptual 2

4.5. Actors

Hi ha dos tipus d'actors que interactuen amb l'aplicació, el primer i amb més privilegis és l'administrador. S'encarrega de crear operacions, de configurar diferents aspectes d'un (o més d'un) mòdul i de configurar les caixes. També existeix l'actor usuari. Aquest té menys privilegis que un administrador.

Durant el projecte sempre hem refereixo a un usuari de tipus administrador. En cas contrari s'especificarà detalladament. Per facilitar la comprensió dels privilegis que disposa cada grup d'usuaris primer es mostra un diagrama amb les accions relacionades al registre de caixa:

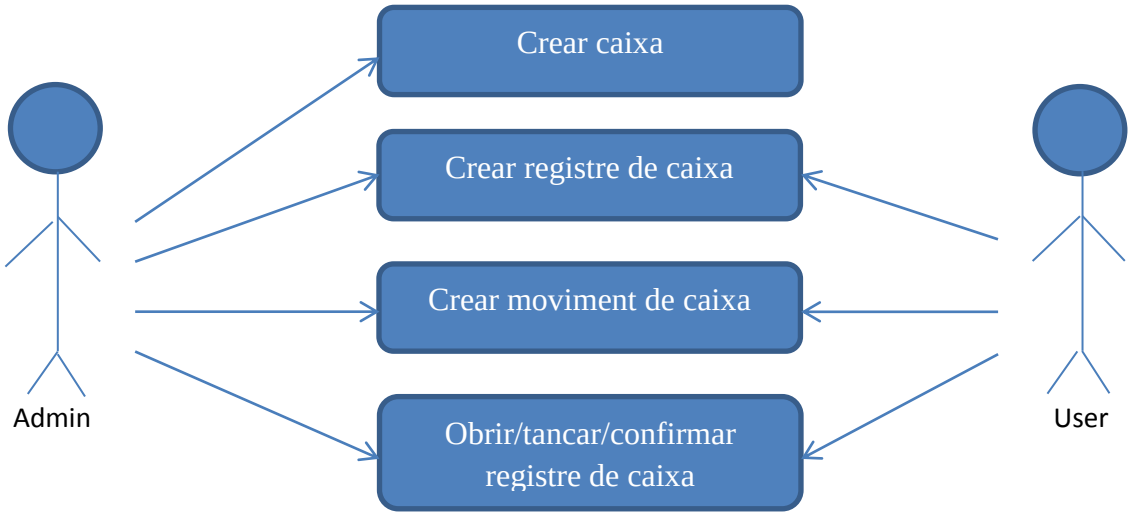


Figura 6: DC actors 1

- **Descripció:**

  - Crear caixa: Quan es crear una caixa a l'aplicació s'han d'emmagatzemar al sistema totes les dades necessàries.

  - Crear registre de caixa: Quan es crear un registre de caixa a l'aplicació, utilitzant la caixa s'ha de guardar al sistema totes les dades necessàries.

  - Obrir registre de caixa: Inicialitza el registre de caixa permetent registrar moviments de caixa.

  - Tancar registre de caixa: El registre de caixa ja no pot registrat més moviments de caixa. També es necessita calcular el saldo de tancament.

  - Confirmar registre de caixa: Quan es confirma el registre de caixa es creen un assentament amb dos apunts comptables per cada moviment de caixa. S'actualitza el saldo d'obertura de la caixa.

- Crear moviments de caixa: Quan es dona d'alta al sistema un moviment de caixa es guarda al sistema tota la seva informació.

A continuació es mostra un diagrama amb una visió general de les funcionalitats de l'aplicació. El diagrama també mostra els privilegis que tenen cada grup d'usuaris.

Figura 7: DC actors 2

- **Descripció:**

    - Crear venda: Quan es crea una venda minorista es guarda al sistema tota la seva informació.

    - Pagar/devolució venda: Quan es realitzen pagaments/abonament d'una venda són anotats per un registre de caixa. S'aniran anotant els pagaments/abonament al registre fins que l'import a pagar de la venda minorista sigui igual a zero. En aquest moment es crearà al sistema un nou albarà associat a la venda.

    - Processar venda: Quan es processa una venda es dóna d'alta al sistema una nova factura associada a la venda minorista.

    - Crear operació: Quan es dóna d'alta al sistema una nova operació es guarda al sistema tota la informació requerida.

    - Introduir operació: Genera un moviment de caixa al registre de caixa.

    - Configurar TPV: Permet configurar la informació que es mostrarà per defecte quan es vagi a crear una venda minorista al sistema.

III. Disseny

5. Disseny

5.1. Casos d'ús i models de col·laboració

5.1.1. CU 1 - Crear caixa

Descripció: *Una funcionalitat bàsica per poder començar a utilitzar el mòdul POS és la caixa. Aquest cas d'ús s'encarrega de la creació al sistema d'una caixa. S'inicialitza amb un saldo d'obertura igual a zero.*

- Contractes:
    - **new_cash** (idcompanyia, iddiari, name, moneda)
    - **end_cash** ( )

New cash (idcompanyia, iddiari, name, moneda)
És el formulari de creació d'una caixa.

PRE:

1. Existeix al sistema una *co* companyia relacionada amb l'identificador *idcompanyia.*
2. Existeix al sistema un *d diari* relacionat amb l'identificador *iddiari.*
3. *Mo* moneda correspon a la moneda associada a la *co* companyia.
4. El *d* diari té un compte pel debe i una altre per l'haver.
5. *Nm* és un nom vàlid per la caixa.

POST:

1. Es crea al sistema una nova *ca* caixa amb el nom *nm.*
2. S'associa la *co, mo* i d a la nova caixa *ca*.
3. El saldo d'obertura de la caixa s'inicialitza a 0.
4. La caixa *ca* es manté activa.

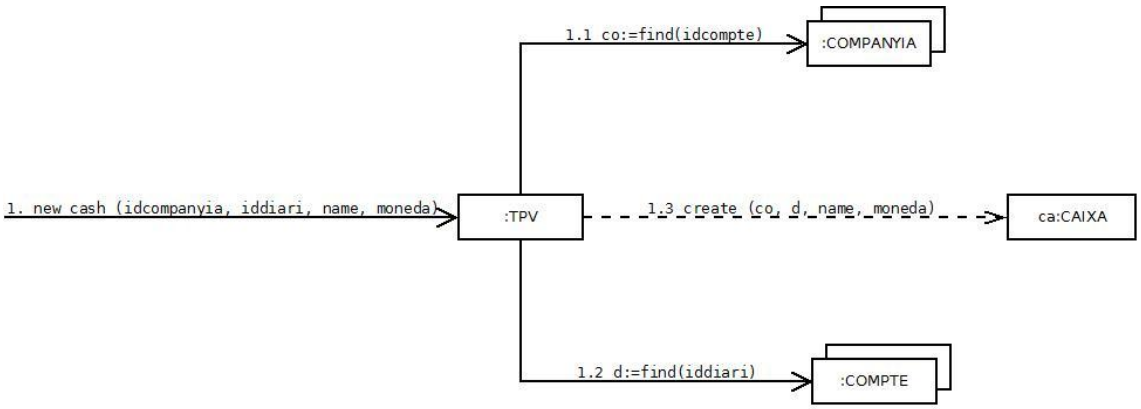


Figura 8: CU 1 – new cash

End_cash ( )

PRE:

1. Hi ha una *ca* caixa activa al sistema.
2. La *ca* caixa activa té associat una *co* companyia, *mo* moneda, *d* diari i un *nm* nom.

POST:

1. S'emmagatzema la nova *ca* caixa al sistema.
2. La nova caixa *ca* deixar d'estar activa al sistema.

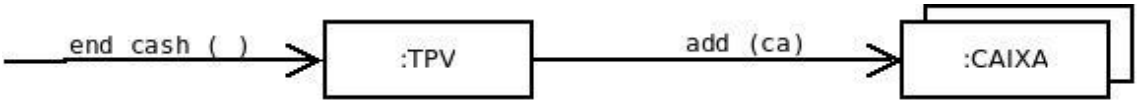


Figura 9: CU 1 – end cash

Diagrama de seqüència:

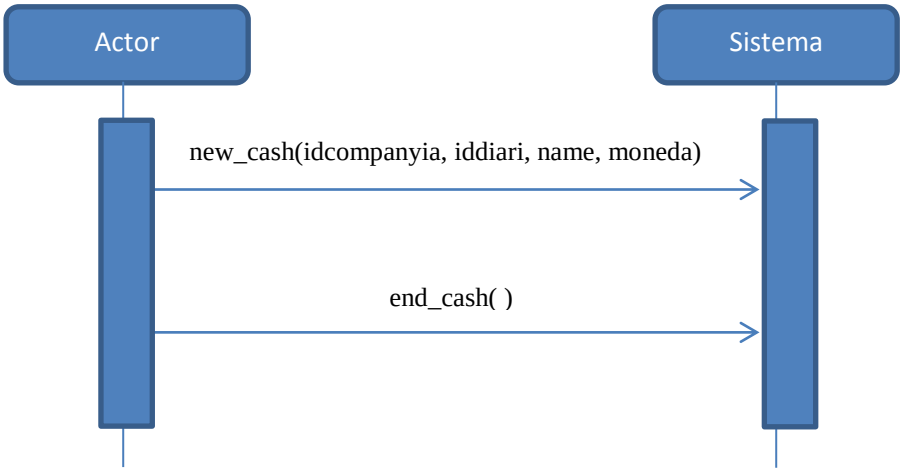


Figura 10: CU 1 – crear caixa

5.1.2. CU 2 – Crear registre de caixa

Descripció: *Un registre de caixa s'ocupa de enregistrar moviments de caixa. Per defecte es crea el registre de caixa amb l'estat draft. En aquest estat no pot enregistrar cap moviment de caixa. D'aquí en endavant, l'usuari sempre estarà actiu ja que si no fos així no es podria fer cap acció. En cas contrari s'especifica detalladament. Aquesta funcionalitat la gestiona un mòdul ja implementat en Tryton.*

- Contractes:
    - **new_cash_reg** (idperiode, idusuari, idcaixa)
    - **end_cash_reg** ( )

New_cash_reg (idperiode, idcaixa, nom, op_saldo)
Formulari d'entrada de dades per la creació del registre de caixa. Per defecte es crea en estat draft. Per tant, d'aquí en endavant l'usuari sempre es actiu fins que no es faci logout. En cas contrari s'especificarà detalladament.

PRE:

1. Existeix al sistema un *pe* període relacionat amb l'identificador *idperiode.*
2. Existeix al sistema una ca caixa relacionada amb l'identificador idcaixa.
3. Hi ha un *us* usuari actiu al sistema.

POS:

1. Es crea al sistema un nou *re* registre de caixa.
2. S'associa el *pe* període, *us* usuari i la *ca* caixa al nou *re* registre de caixa.
3. L'estat per defecte del *re* registre de caixa és *draft.*
4. El saldo d'obertura del *re* registre de caixa s'actualitza amb el saldo d'obertura de la *ca* caixa
5. El nou *re* registre de caixa es manté actiu al sistema.

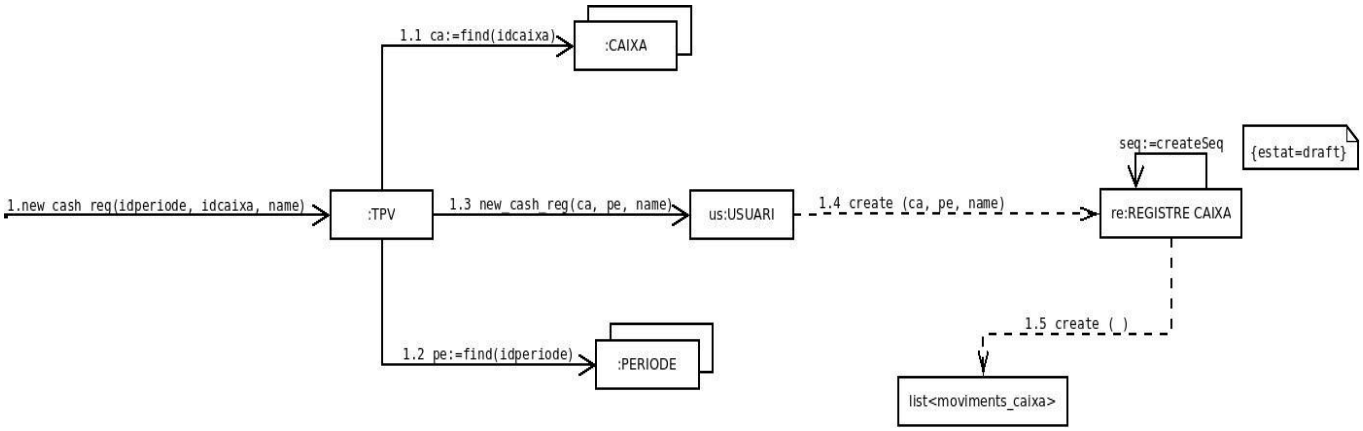


Figura 11: CU 2 – new_cash_reg

End_cash_reg ( )

PRE:

1. Hi ha un *re* registre de caixa actiu al sistema.
2. El *re* registre de caixa té associat un *pe* període, una *ca* caixa i un *us* usuari i té un saldo d'obertura.

POST:

1. S'emmagatzema el nou *re* registre de caixa al sistema.
2. El *re* registre de caixa ja no es actiu al sistema.

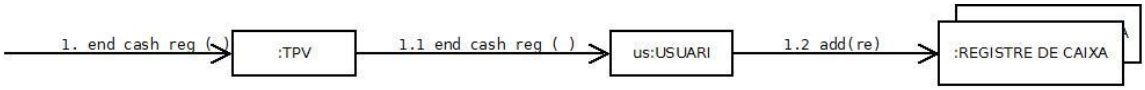


Figura 12: CU 2 – end_cash_reg

Diagrama de seqüència:

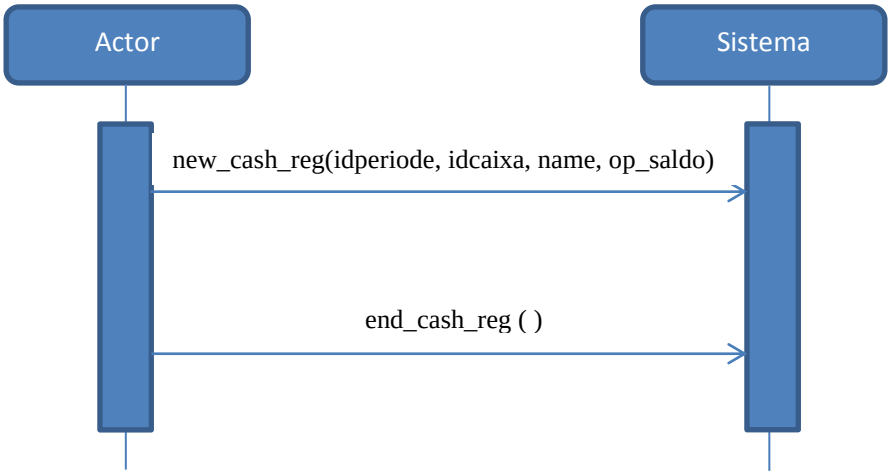


Figura 13: CU 2 – Crear registre de caixa

5.1.3. CU 3 – Obrir registre de caixa

Descripció: *És l'encarregat de modificar l'estat d'un registre de caixa de draft a open. A partir d'aquest moment el registre de caixa, en estat open, ja podrà registrar moviments de caixa. Quan s'obre el registre de caixa també es crea una seqüència; és un sistema de numeració automàtica gestionat en un altre mòdul de Tryton i que es pot aprofitar en aquest cas d'ús.*

- Contractes:
    - **open_cash_reg** (idreg)

Open_cash_reg (idreg)
Es polsa sobre el botó obrir que hi ha en el formulari del registre de caixa.

PRE:

1. Existeix al sistema un *re* registre de caixa corresponent a l'identificador *idreg.*
2. Hi ha un *us* usuari actiu.
3. El *re* registre de caixa es troba en estat *draft.*

POST:

1. El *re* registre de caixa adopta l'estat *open.*
2. Si el *re* registre de caixa té associat un *nm* nom, s'inicià una seqüència amb aquest nom.
3. Si el *re* registre de caixa no té associat un *nm* nom, aquest a vegada el valor de la *seq* seqüència.
4. Es *re* registre de caixa guarda la data d'obertura
5. El *re* registre de caixa es manté actiu.

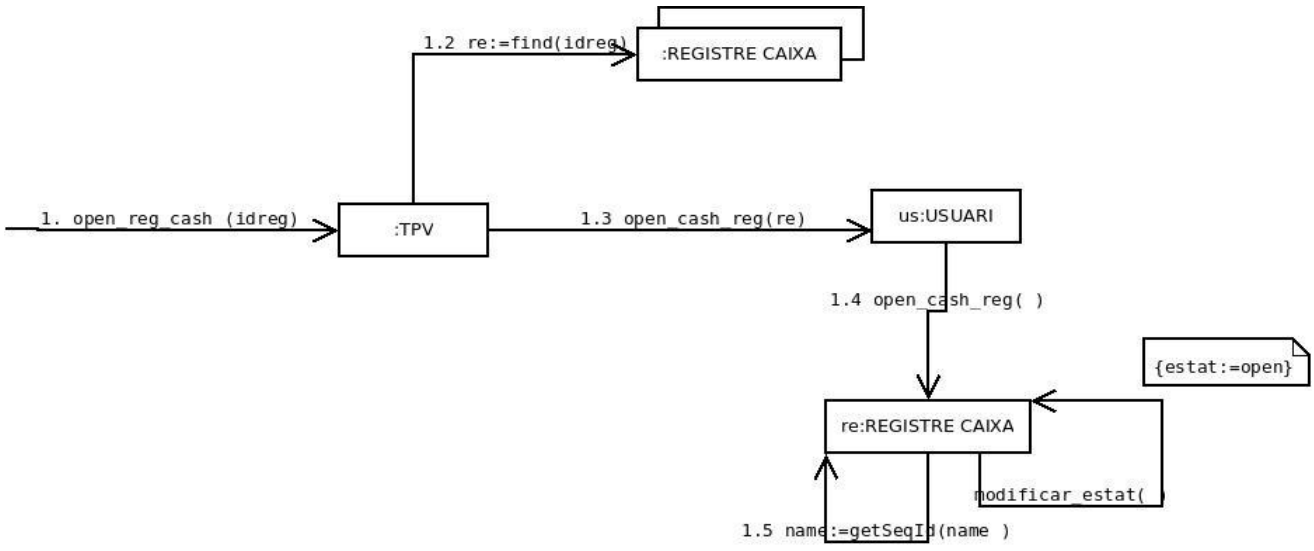


Figura 14: CU 3 – open cash reg

Diagrama de seqüència:

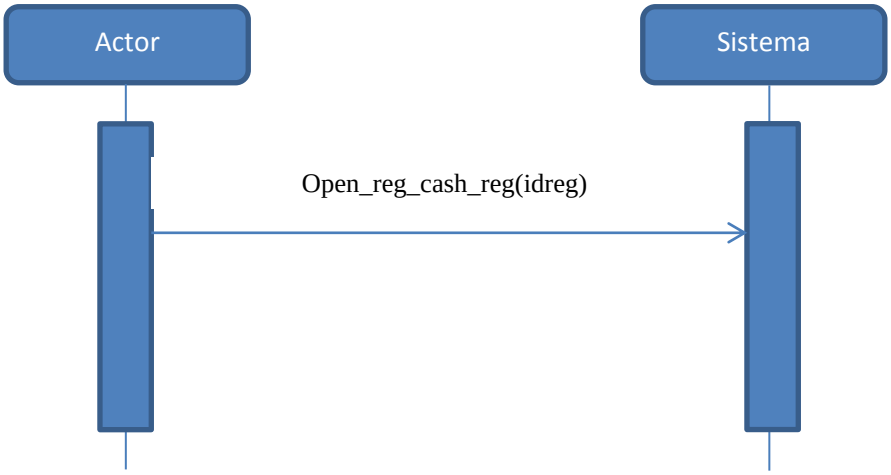


Figura 15: CU 3 – open cash reg

5.1.4. CU 4 – Crear moviments de caixa

Descripció: *Aquest cas d'ús correspon a la creació d'un moviment de caixa al registre de caixa. Un moviment de caixa pot ser creat des d'un pagament de venda TPV, des de la introducció d'una operació, o manualment per l'usuari. Com que el moviment de caixa pot ser de tres orígens diferents, per evitar confusions i aprofitant el polimorfisme dels llenguatges orientats a objectes, mostro tres esdeveniments de sistema corresponents a cada cas, en el diagrama de seqüència només es té en compte el primer cas.*

- Contractes:
    - **new_cash_mov** (idtercer, date, compte, import, origen, description)
    - **new_cash_mov**(idtercer, date, idcompte, import, origen, description, pg)
    - **new_cash_mov**(op, data, import)
    - **end_cash_mov** ( )

New_cash_mov (idtercer, date, idcompte, import, origen, description)
Formulari del registre de caixa. S'editen els moviments de caixa.

PRE:

1. La data es troba en format vàlid.
2. Existeix al sistema un *c* compte corresponent a l'identificador *idcompte.*
3. Existeix al sistema un *tr* compte corresponent a l'identificador *idtercer*.

4. L'import no es buit
5. Hi ha un *re* registre de caixa actiu al sistema.
6. El *re* registre de caixa actiu es troba en estat *open.*

POST:

1. S'ha creat al sistema un *mv* moviment de caixa.
2. S'associa al nou *mv* moviment de caixa la data, l'import, l'origen, el *c* compte i la descripció.
3. S'associa el *tr* tercer al nou *mv* moviment de caixa
4. L'estat per defecte del nou *mv* moviment de caixa és *draft.*
5. L'origen per defecte és *register.*
6. El nou *mv* té associat com a mínim l'import i el *c* compte.
7. El *re* registre de caixa es manté actiu al sistema.

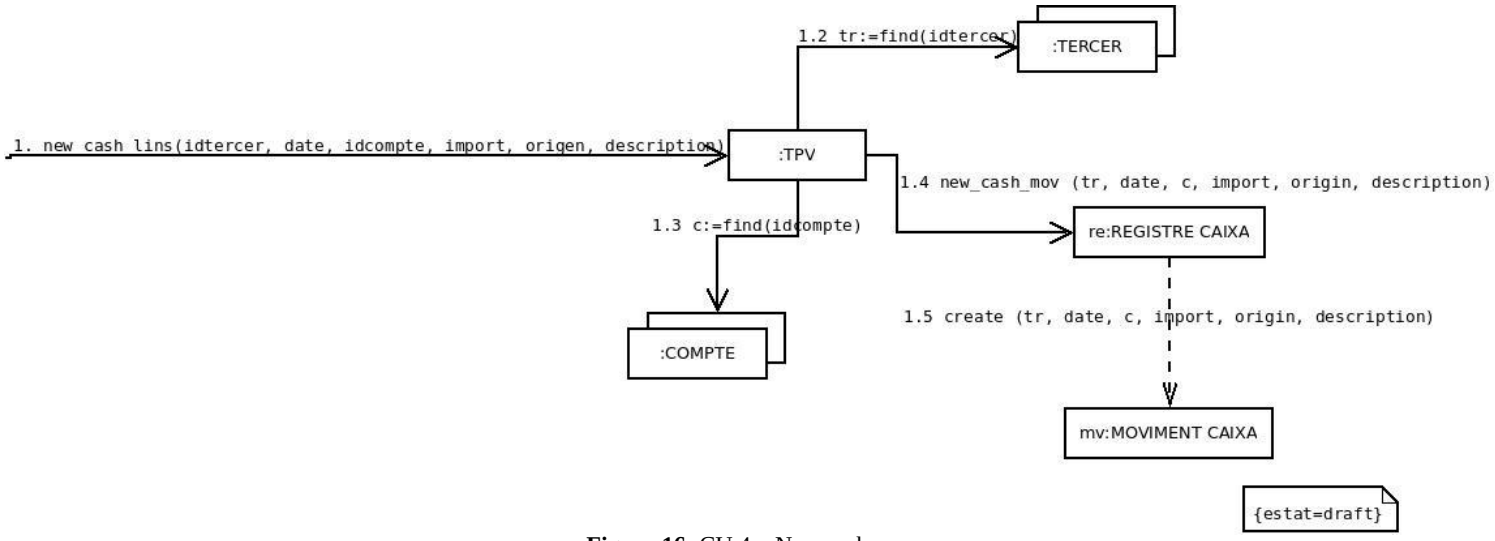


Figura 16: CU 4 – New cash mov

New_cash_mov (idtercer, date, idcompte, import, origen, description, pg)
Moviment de caixa provinent d'una venda.

PRE:.

1. Hi ha un *re* registre de caixa actiu al sistema.
2. El *re* registre de caixa actiu es troba en estat *open.*
3. La *v* venda es activa al sistema
4. El *pg* pagament és actiu al sistema.

POST:

1. S'ha creat al sistema un *mv* moviment de caixa.
2. S'associa al nou *mv* moviment de caixa al *pg* pagament/devolució.
3. L'estat per defecte del nou *mv* moviment de caixa és *draft.*
4. L'atribut *origen* del *re* registre de caixa és *sale*
5. El nou *mv* té associat com a mínim l'import i el *c* compte.

6. El *re* registre de caixa es manté actiu al sistema.

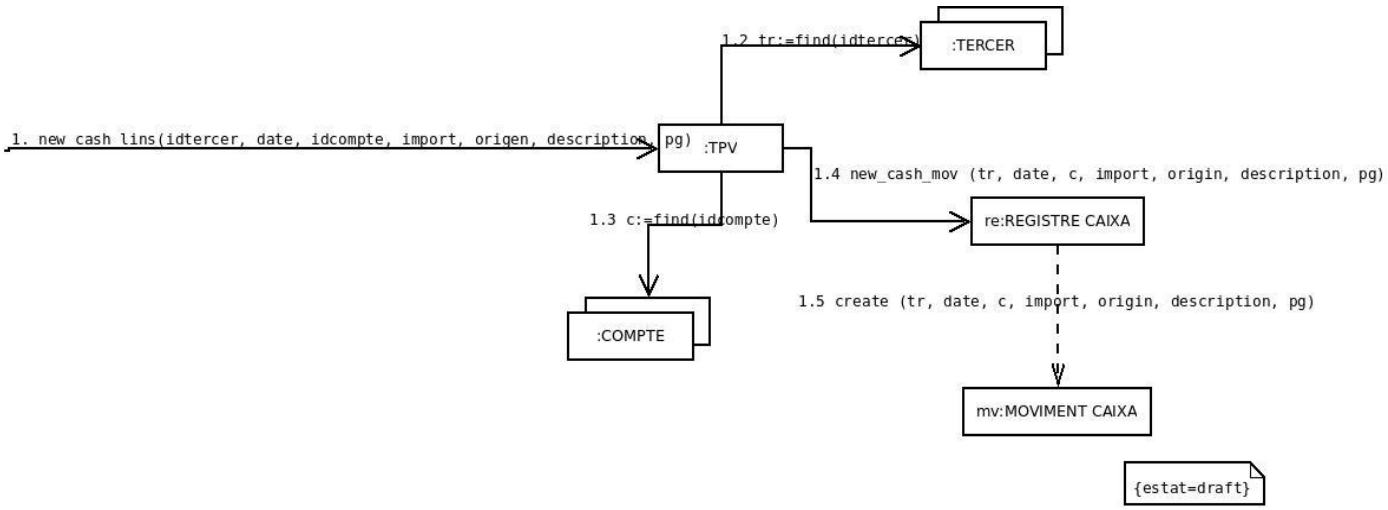


Figura 17: CU 4 – New cash mov sale

New_cash_mov (op, data, import)
Moviment de caixa provinent d'una operació d'entrada o de sortida.

PRE:

1. Hi ha una *op* operació activa al sistema.
2. Hi ha un *re* registre de caixa actiu al sistema.
3. El *re* registre de caixa actiu es troba en estat *open.*

POST:

1. S'ha creat al sistema un *mv* moviment de caixa.
2. S'associa al nou *mv* moviment de caixa amb l'*op* operació d'entrada o sortida, amb la data, i amb l'import.
3. L'estat per defecte del nou *mv* moviment de caixa és *draft.*
4. L'atribut *origen* del *re* registre de caixa és *operation.*
5. El nou *mv* té associat com a mínim l'import i el *c* compte.
6. El *re* registre de caixa es manté actiu al sistema.

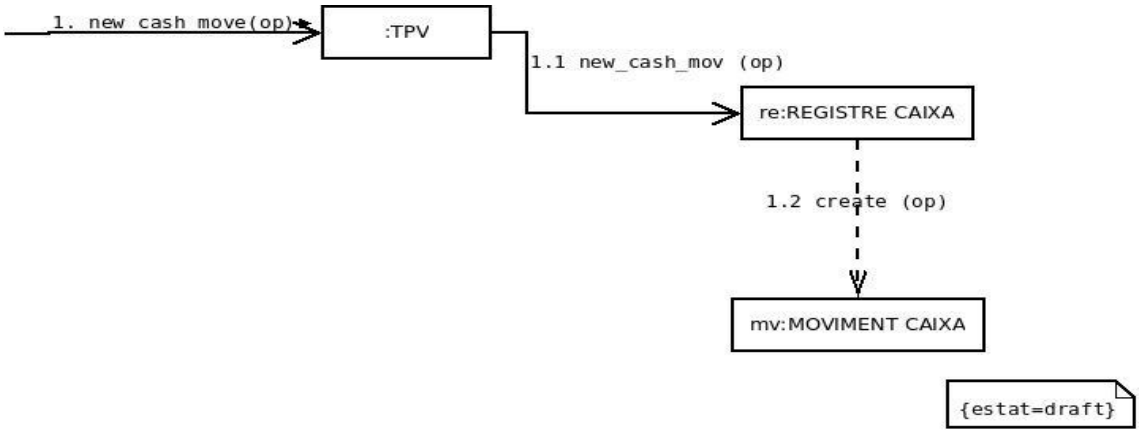


Figura 18: CU 4 – New cash mov sale amb operació

End_cash_mov ( )
Es finalitza la introducció de moviments de caixa al registre de caixa emmagatzemant els moviments en el respectiu registre. Aquest es manté actiu.

PRE:

1. Hi ha un *re* registre de caixa actiu.
2. L'estat del *re* registre de caixa es obert.

POST:

1. S' emmagatzema el nou *mv* moviment de caixa
2. El *re* registre de caixa es manté actiu.

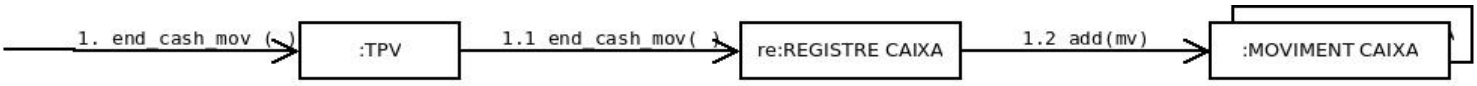


Figura 19: CU 4 – End cash mov

Diagrama de seqüència:

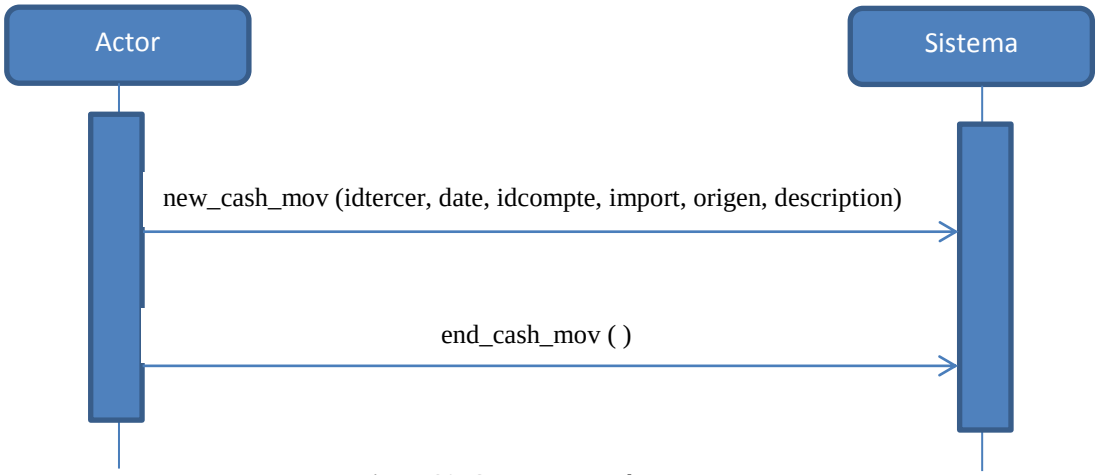


Figura 20: CU 4 – New cash mov

5.1.5. CU 5 – Tancar registre de caixa

Descripció: *A l'hora de tancar el registre de caixa es calcula la suma dels imports dels moviments de caixa. S'activa un camp per poder quadrar la caixa i el registre de caixa adopta l'estat close. En aquest estat no es poden registrar més moviments de caixa.*

- Contractes:
    - **close_reg_cash** ( )

Close_reg_cash ( )
En el formulari de registre de caixa es polsa el botó tancar.

PRE:

1. Hi ha un *re* registre de caixa actiu.
2. El *re* registre de caixa es troba en estat *open*.
3. Hi ha un *us* usuari actiu.

POST:

1. El *re* registre de caixa calcula el total dels imports dels *mv* moviments de caixa.
2. El saldo es igual a saldo_mov + op_saldo.
3. El *re* registre de caixa guarda la data de tancament.

4. El *re* registre de caixa adopta l'estat *close*.
5. El *re* registre de caixa es actiu al sistema.

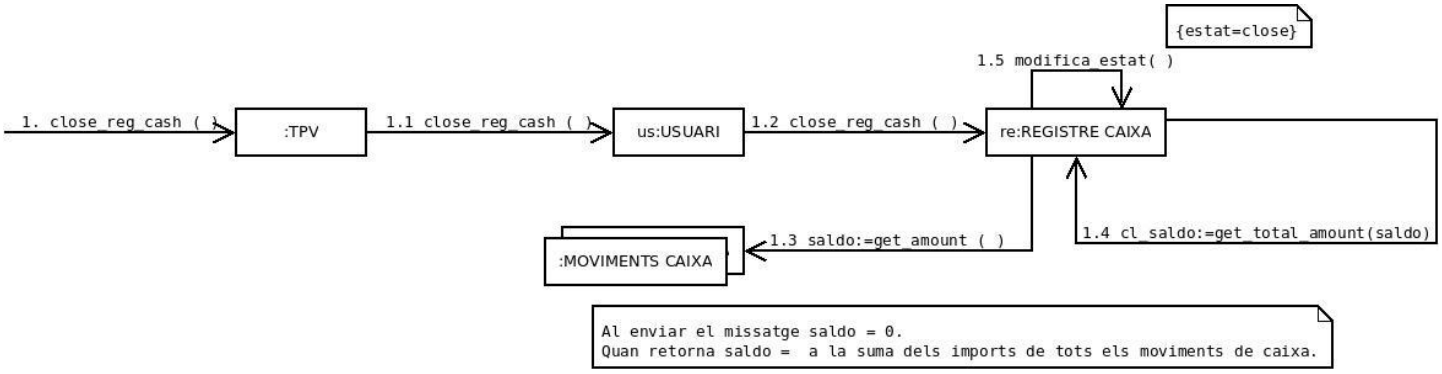


Figura 21: CU 5 – close_reg_cash

Diagrama de seqüència:

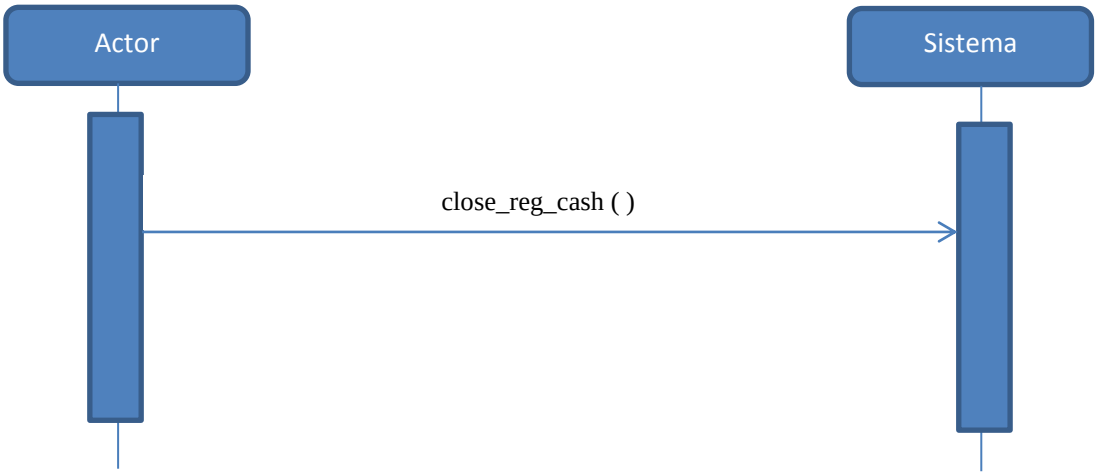


Figura 22: CU 5 – close_reg_cash

5.1.6. CU 6 – Confirmar registre de caixa

Descripció: *Quan es confirma el registre de caixa, primer s'ha de quadrar el balanç de la caixa. Per quadrar-ho l'usuari ha de ficar la suma de les operacions. Ha de ser igual al saldo de tancament (suma moviments de caixa + saldo obertura). Si quadra, es crea un moviment comptable per cada operació. El moviment comptable inclou 2 apunts comptables.*

- Contractes:
    - **confirm_reg_cash** (cl_balanç)
    - **add_move** ( )
    - **actualitza_caixa** ( )
    - **end_confirm** ( )

Confirm_reg_cash (cl_balanç)
Formulari de registre de caixa. Es polsa el botó confirmar

PRE:

1. Hi ha un *re* registre de caixa actiu al sistema.
2. Hi ha un *us* usuari actiu al sistema.
3. El *re* registre de caixa té l'estat *close*.
4. No hi ha cap *mc* moviment comptable pel mateix *mv* moviment de caixa.

POST:

1. Es comprova si el *re* registre de caixa quadra.
2. Si el *re* registre de caixa quadra, es crea al sistema un nou *mc* moviment comptable.
3. S'associa el període, la cl_data i el diari comptable associat a la *ca* caixa associada al *re* registre de caixa amb el nou *mc* moviment comptable.
4. El *re* registre de caixa es manté actiu al sistema.
5. El *mc* moviment comptable es manté actiu al sistema.
6. El *mc* moviment de caixa es manté actiu al sistema.
7. L'usuari *us* es manté actiu al sistema.

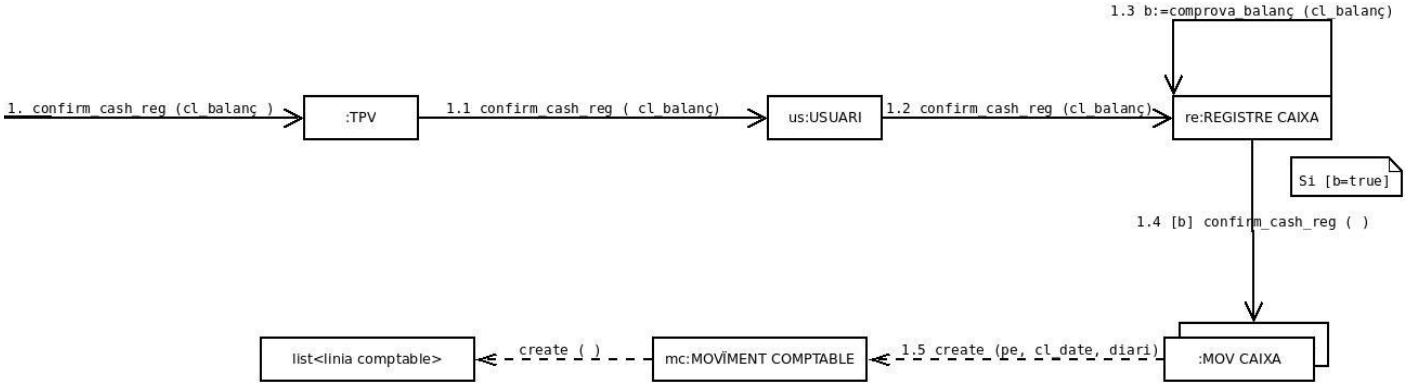


Figura 23: CU 6 – confirm_reg_cash

Add_move ( )
En aquest cas d'ús es creen al sistema dos línies comptables (assentaments) associats al moviment comptable.

PRE:

1. Hi ha un *re* registre de caixa actiu al sistema.
2. Hi ha un *us* usuari actiu al sistema.
3. Hi ha un *mc* moviment comptable en actiu al sistema.
4. Hi ha un *mc* moviment de caixa actiu al sistema.
5. El *mc* moviment de caixa té l'estat *draft.*
6. El *re* registre de caixa es troba en estat *close.*

POST:

1. S'han creat un *ass* assentaments comptables al sistema associat amb la descripció, total_amount, i compte comptable.
2. S'emmagatzema al sistema l'assentament comptable.
3. El *mc* moviment de caixa adopta l'estat *confirmed.*
4. El *mc* moviment de caixa es manté actiu al sistema.
5. El *re* registre de caixa es manté actiu.
6. El *mco* moviment comptable es manté actiu al sistema.
7. L'usuari *us* es manté actiu.

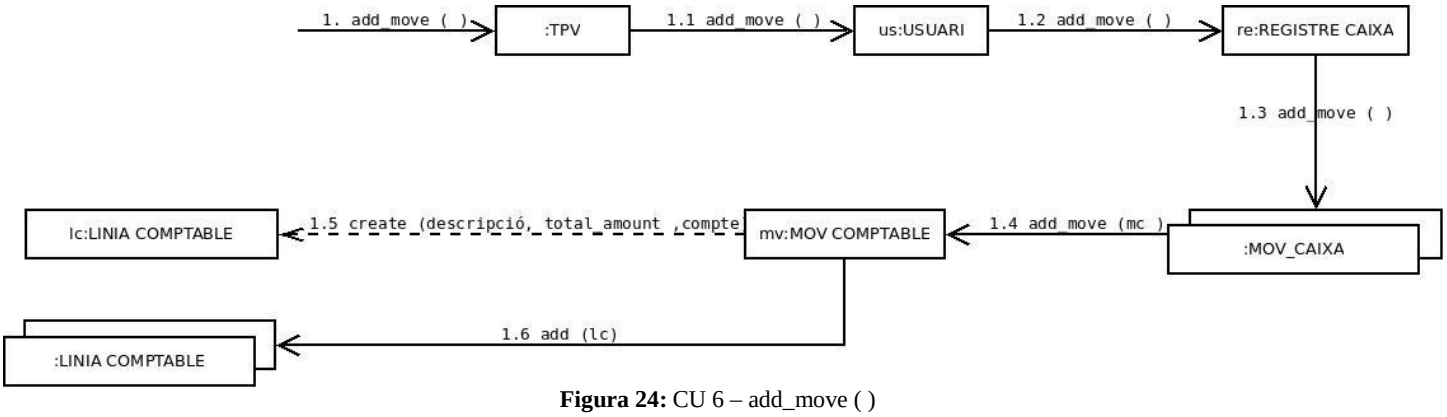


Figura 24: CU 6 – add_move ( )

Actualitza_caixa ( )

PRE:

1. Hi ha un *re* registre de caixa actiu al sistema.
2. Hi ha un *us* usuari actiu al sistema.
3. El *re* registre de caixa es troba en estat *close.*

POST:

1. S'actualitza el saldo d'obertura de la *ca* caixa, associada al *re* registre de caixa.
2. El *re* registre de caixa adopta l'estat *close.*
3. El *re* registre de caixa es manté actiu.
4. L'usuari *us* es manté actiu al sistema.

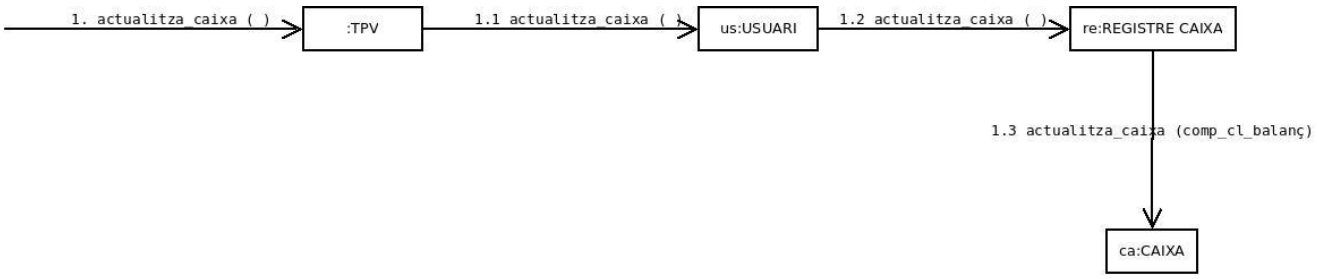


Figura 25: CU 6 – actualitza_caixa ( )

End_confirm ( )

PRE:

1. Hi ha un *re* registre de caixa actiu al sistema.
2. Hi ha un *us* usuari actiu al sistema.
3. Hi ha un *mv* moviment comptable actiu al sistema.
4. El *re* registre de caixa té l'estat *confirmed.*

POST:

1. S'emmagatzema al sistema el *mv* moviment comptable.
2. El *mv* moviment comptable ja no es actiu al sistema.
3. El *mc* moviment de caixa ja no és actius.
4. El *re* registre de caixa ja no es actiu al sistema.

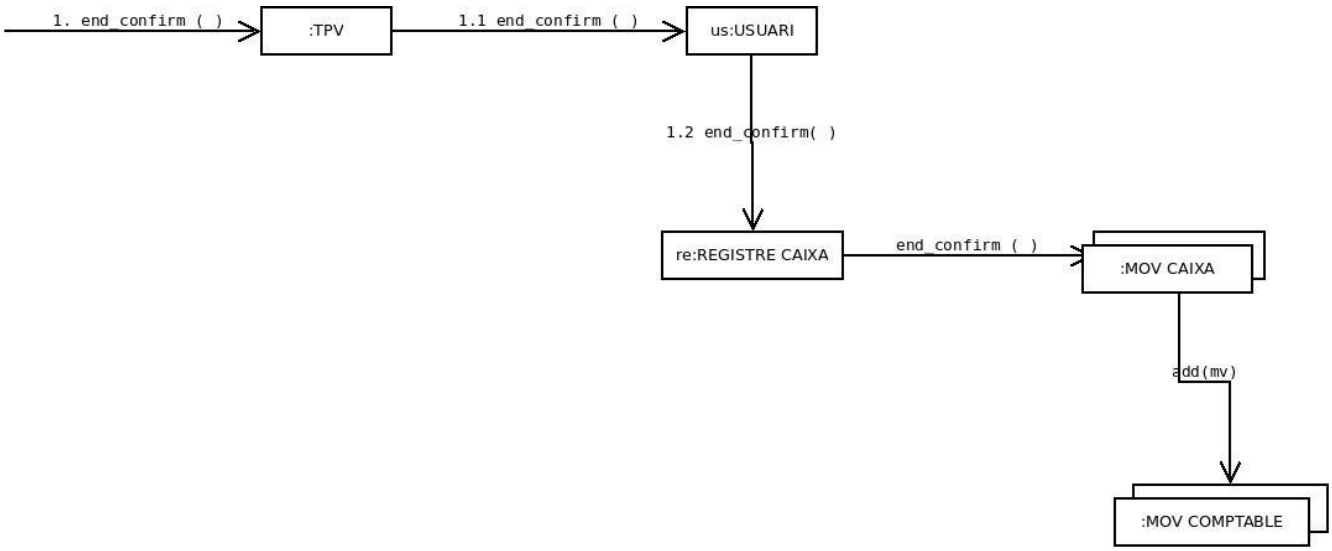


Figura 26: CU 6 – end_confirm ( )

Diagrama de seqüència:

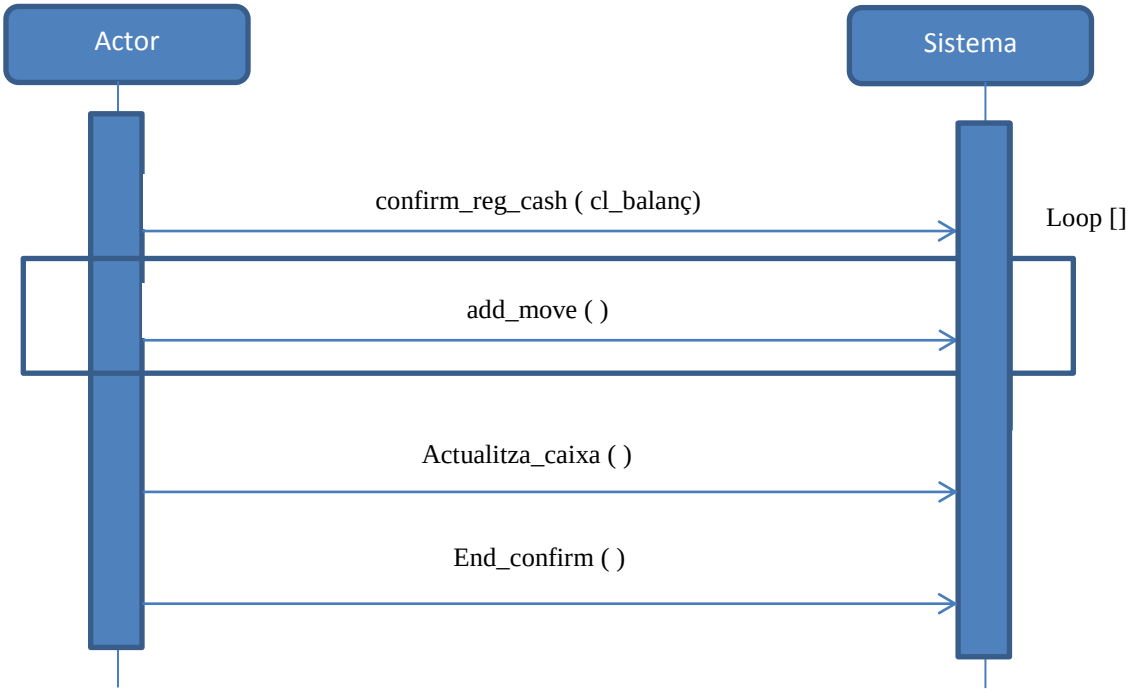


Figura 27: CU 6 – confirm_reg_cash

5.1.7. CU 7 – Nova venda

Descripció: *S'encarrega de crear una nova venda al sistema amb les seves línies.*

- Contractes:
    - **New_sale** (idtercer, descripció, idmoneda)
    - **add_sale_line** (idproduct)
    - **end_sale** ( )

New_sale (idtercer, descripció)
Formulari de creació una venda al sistema.

PRE:

1. Existeix al sistema un *tr* tercer corresponent a l'identificador *idtercer*.
2. Existeix al sistema una *mo* moneda corresponent a l'identificador *idmoneda*.
3. Hi ha un *us* usuari actiu al sistema.

POST:

1. S'ha creat al sistema una nova *v* venda associat al *tr* tercer, *mo* moneda.
2. La nova *v* venda té l'estat *draft.*
3. La nova *v* venda es manté activa al sistema
4. L'usuari *us* es manté activa al sistema.

L'esdeveniment de sistema *obtenir_tercer (idtercer)* aprofita el cas d'ús *configurar TPV* per si de cas n'hi ha un tercer definit per defecte quan es crea una nova venda.

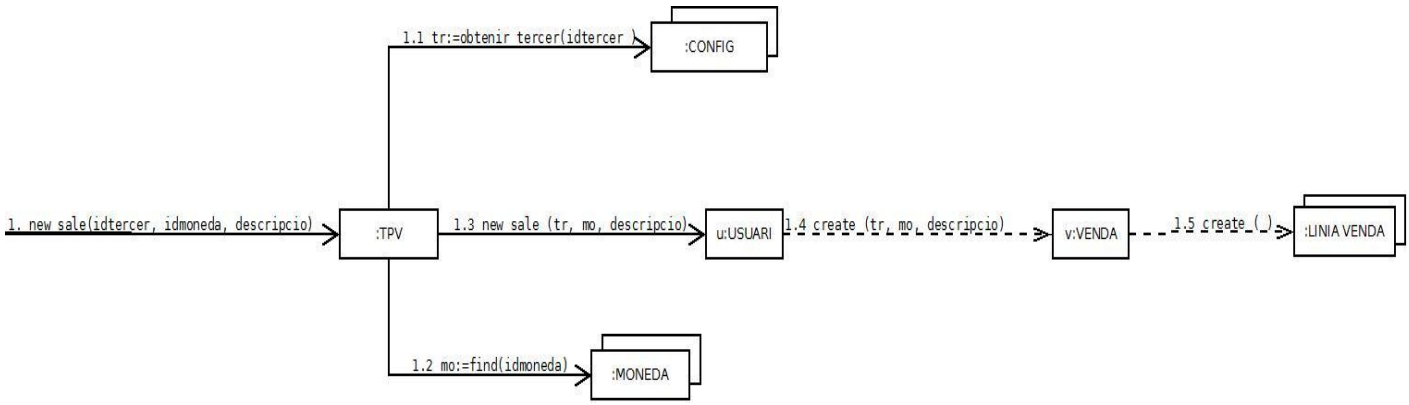


Figura 28: CU 7 – new_sale

Add_sale_line (idproduct, unitats)
Es creen línies de venda a la venda. A medita que es van introduint línies de venda a la venda, es va actualitzant els imports d'aquesta.

PRE:

1. Hi ha una *v* venda activa al sistema
2. Hi ha un *us* usuari actiu al sistema.
3. Les unitats es un número natural major que 0.
4. Existeix al sistema un *p* product corresponent a l'identificador *idproduct.*

POST:

1. S'ha creat una nova *lv* línia de venda al sistema associada al producte.
2. El total de la *v* venda s'actualitza amb el preu*unitats del *p* producte.
3. S'emmagatzema al sistema la nova *lv* línia de venda.
4. La *v* es manté activa al sistema.
5. L'usuari *us* es manté actiu al sistema.
6. Les PRE no es modifiquen.

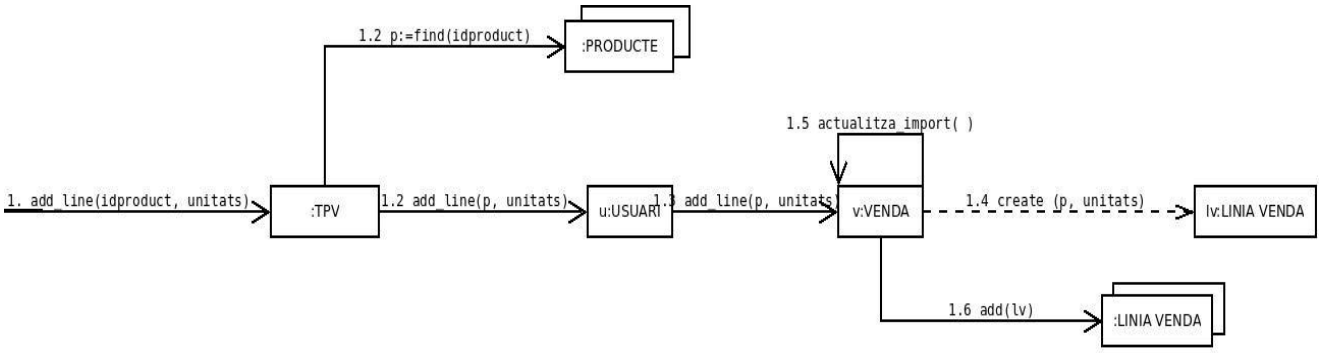


Figura 29: CU 7 – add_line

End_sale ( )

PRE:

1. Hi ha *v* venda activa al sistema
2. Hi ha un *us* usuari actiu al sistema.
3. La *v* venda té l'estat *draft.*

POST:

1. S'emmagatzema la *v* venda activa al sistema.
2. La *v* venda ja no es activa.

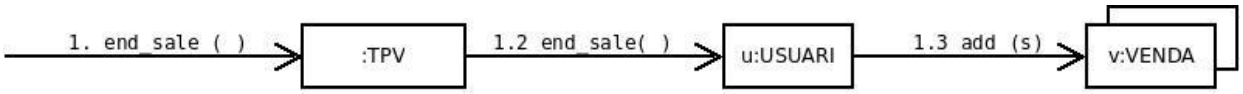


Figura 30: CU 7 – end_sale

Diagrama de seqüència:

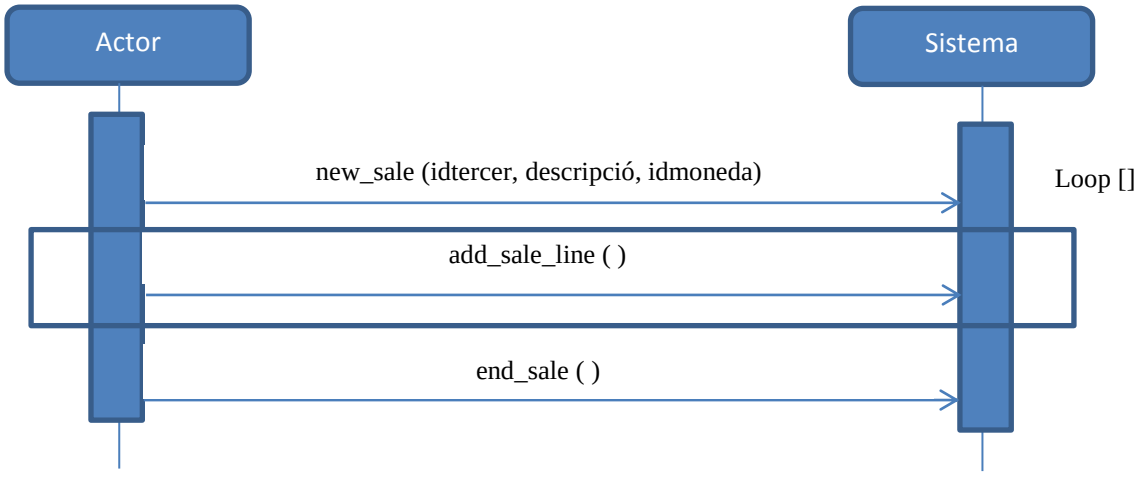


Figura 31: CU 7 – new_sale

5.1.8. CU 8 – Pagar/abonar venda

Descripció: *Per pagar una venda és necessari realitzar un pagament/devolució. Un pagament crea un moviment de caixa al registre de caixa seleccionat amb l'import introduït per l'usuari.*

- Contractes:
    - **New_pay** (idreg, idvenda, import, date)
    - **pay_sale** ( )
    - **end_pay** ( )

New_pay (idreg, idvenda, import, date)
Des del formulari de venda es clica sobre el botó pay. Es mostra un assistent per realitzar el pagament. Per crear el pagament al sistema, l'usuari envia al registre de caixa la responsabilitat de crear el moviment de caixa a partir del pagament.

PRE:

1. Existeix al sistema un *re* registre de caixa corresponent a l'dentificador *idreg.*
2. Existeix al sistema una *v* venda corresponent a l'identificador *idvenda.*
3. El *re* registre de caixa té l'estat *open.*
4. La data es la actual.
5. Hi ha un *us* usuari actiu al sistema.
6. La *v* venda té l'estat *draft.*

POST:

1. S'ha creat al sistema un nou *pg* pagament/devolució.
2. S'ha associat al nou *pg* pagament/devolució el *re* registre de caixa, la *v* venda, la data actual i l'import.
3. S'actualitza l'import a pagar de la *v* venda.
4. El *re* registre de caixa s'ocupa de crear el moviment de caixa.
5. La *v* venda es manté activa al sistema.
6. El *re* registre de caixa es manté actiu al sistema.
7. L'usuari *us* es manté actiu al sistema.

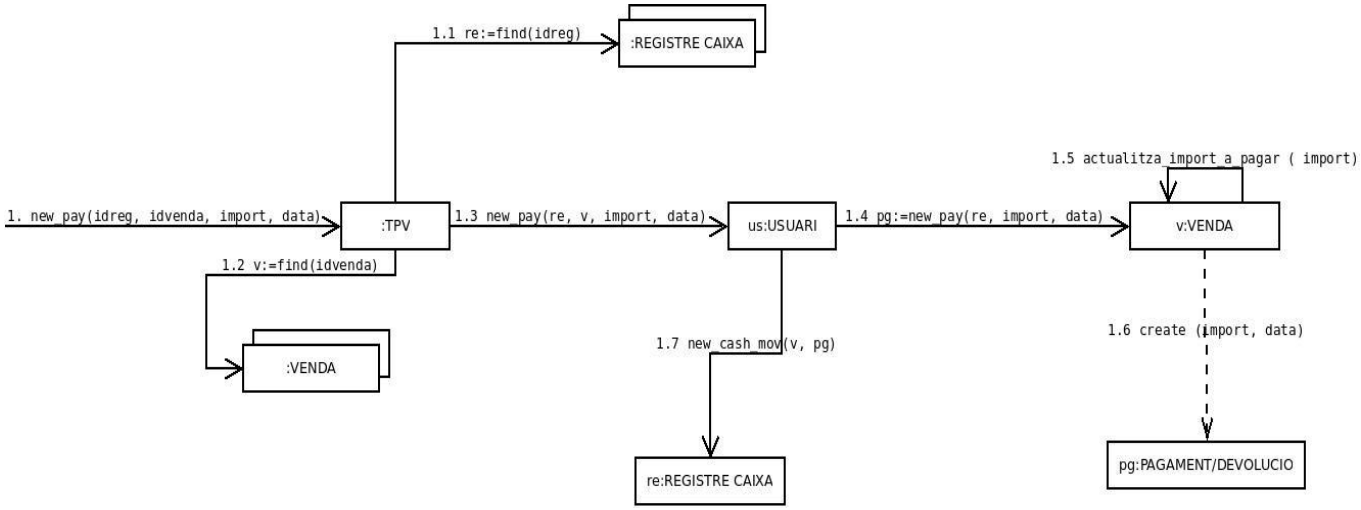


Figura 32: CU 8 – new_pay

Pay_sale ( )

PRE:

1. Hi ha un *pg* pagament actiu al sistema.
2. Hi ha un *re* registre de caixa actiu al sistema.
3. Hi ha una *v* venda actiu al sistema.
4. La *v* venda té l'estat *draft.*

POST:

1. Si l'import total de la *v* venda activa es igual al total a pagar aquesta adopta l'estat *confirmed.*
2. La *v* venda es manté activa al sistema.
3. L'usuari *us* es manté actiu al sistema.
4. El *re* registre de caixa es manté actiu al sistema.

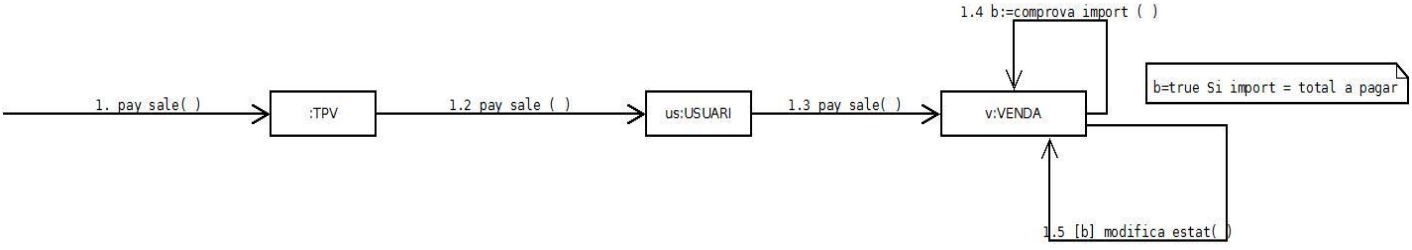


Figura 33: CU 8 – pay_sale

End_pay ( )

PRE:

1. Hi ha una *v* venda activa al sistema.
2. La *v* venda té l'estat *confirmed.*
3. Hi ha un *re* registre de caixa actiu al sistema.
4. El *re* registre de caixa té l'estat *open.*
5. Hi ha un *us* usuari actiu al sistema.
6. Hi ha un *pg* pagament actiu al sistema.

POST:

1. S'emmagatzema el *pg pagament* al sistema.
2. S'associa el *pg* pagament amb el *re* registre de caixa actiu.
3. El *re* registre de caixa es manté actiu al sistema.
4. La *v* venda ja no es activa al sistema.
5. L'usuari *us* es manté actiu al sistema.
6. El *pg* pagament ja no es actiu al sistema.

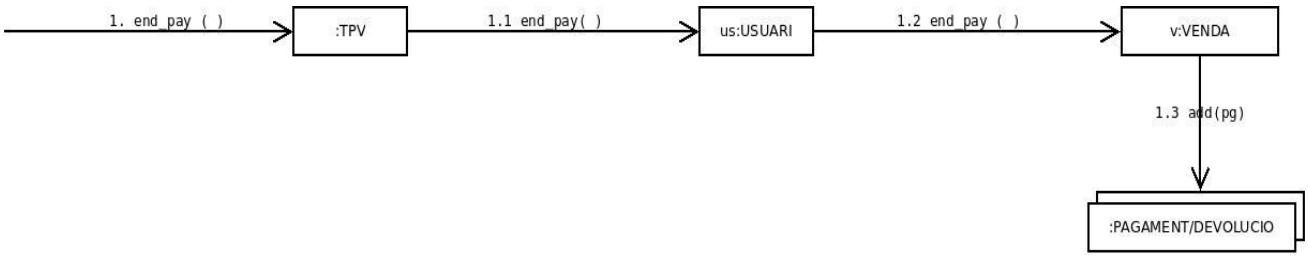


Figura 34: CU 8 – end_pay

Diagrama de seqüència:

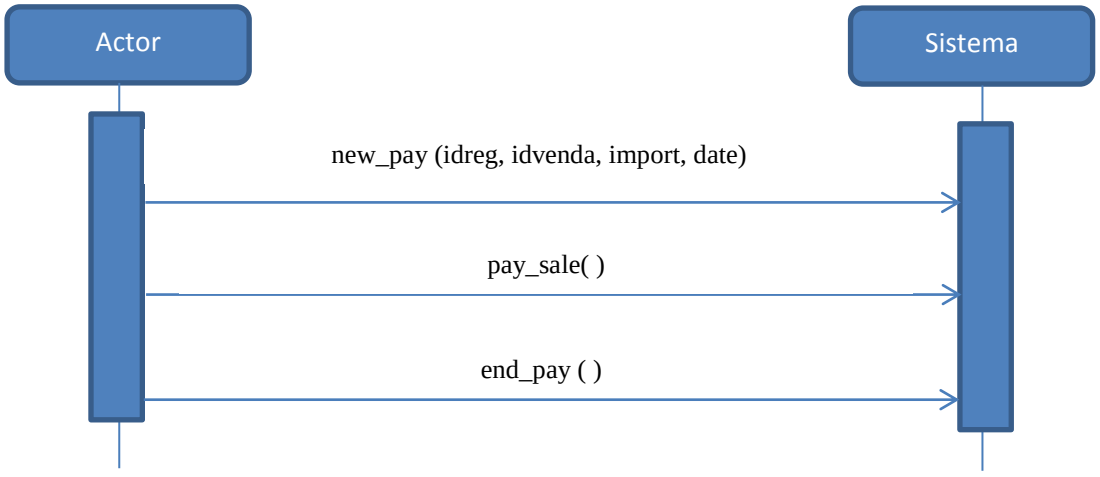


Figura 35: CU 8 – new_pay

5.1.9. CU 9 - Processar venda

Descripció: *Aquest cas d'ús s'encarrega de facturar la venda pos.*

- Contractes:
  - **process_sale** (idvenda)
  - **end_process** ( )

Process_sale (idvenda)
Formulari de la venda. Es polsa el botó process. La factura es crea des de la venda però la gestió d'aquestes es realitza des d'un altre mòdul de Tryton. Per aquest motiu la venda crida al mètode create_invoice(), per referir-se en que en aquest punt es factura la venda.

PRE:

1. Existeix al sistema una *v* venda corresponent a l'identificador *idvenda.*
2. Hi ha un *us* usuari actiu al sistema.
3. No hi ha cap *f* factura associada a la *v* venda.
4. La *v* venda es troba en l'estat *confirmed.*

POST:

1. Es crea la *f* factura al sistema.
2. S'emmagatzema la *f* factura al sistema associant la *v* venda activa.
3. La *v* venda es manté actiu al sistema.
4. L'usuari *us* es manté actiu al sistema.

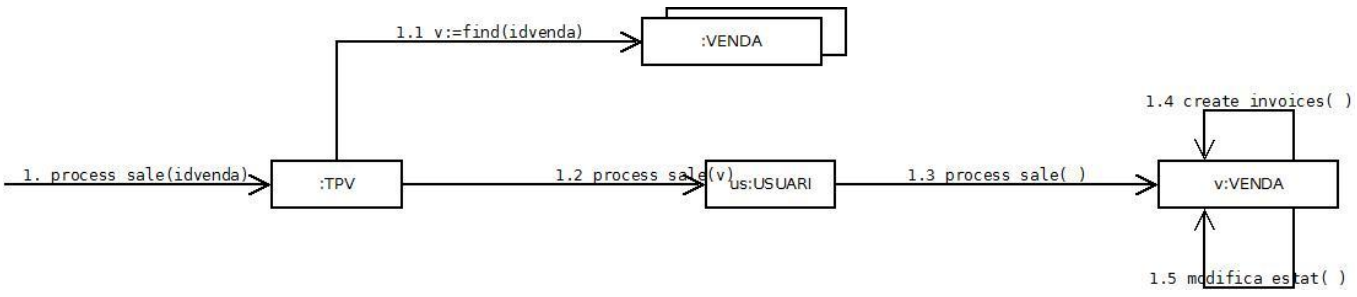


Figura 36: CU 9 – process sale

End_process ( )

PRE:

1. Hi ha una *v* venda activa al sistema
2. La *v* venda té associada una *f* factura.
3. La *v* venda té l'estat *confirmed.*
4. Hi ha un *us* usuari actiu al sistema.

POST:

1. La *v* venda ja no es activa.
2. L'usuari *us* es manté actiu.

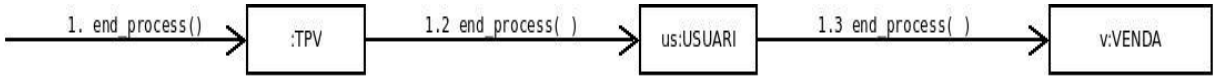


Figura 37: CU 9 – end process

Diagrama de seqüència:

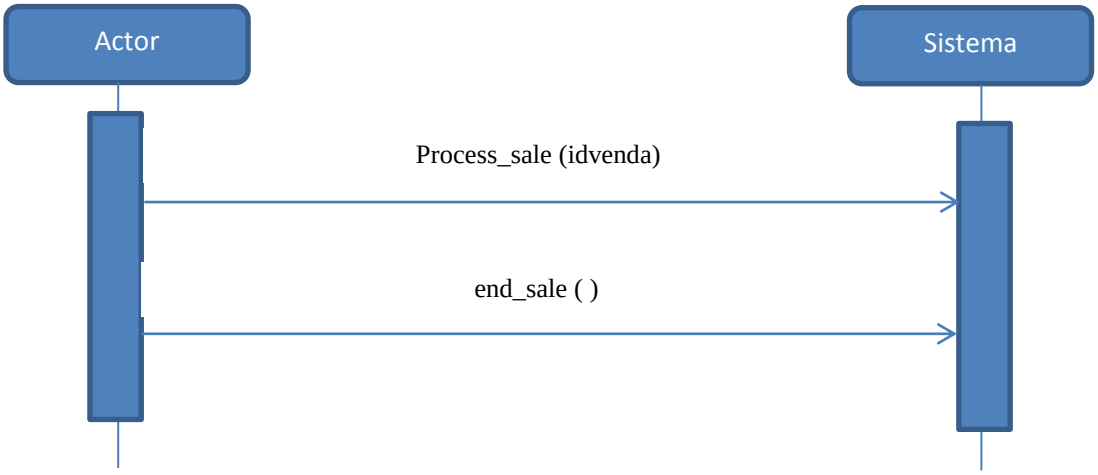


Figura 38: CU 9 – process sale

5.1.10. CU 10 – Nova operació

Descripció: *S'encarrega de crear al sistema una nova operació d'entrada o de sortida. Té associat un compte comptable, un nom i un tipus (in/out).*

- Contractes:
  - **New_operation** (idcompte, type, name)
  - **End_operation** ( )

New_operation (idcompte, type, name)
Formulari de creació d'una operació d'entrada o de sortida.

PRE:

1. Existeix al sistema un *co* compte corresponent a l'identificador *idcompte*.
2. Hi ha un *us* usuari actiu al sistema.

POST:

1. S'ha creat una nova *op* operació al sistema associada al *type*, *nom*.
2. La nova *op* operació es manté activa al sistema.
3. L'usuari *us* es manté actiu al sistema.

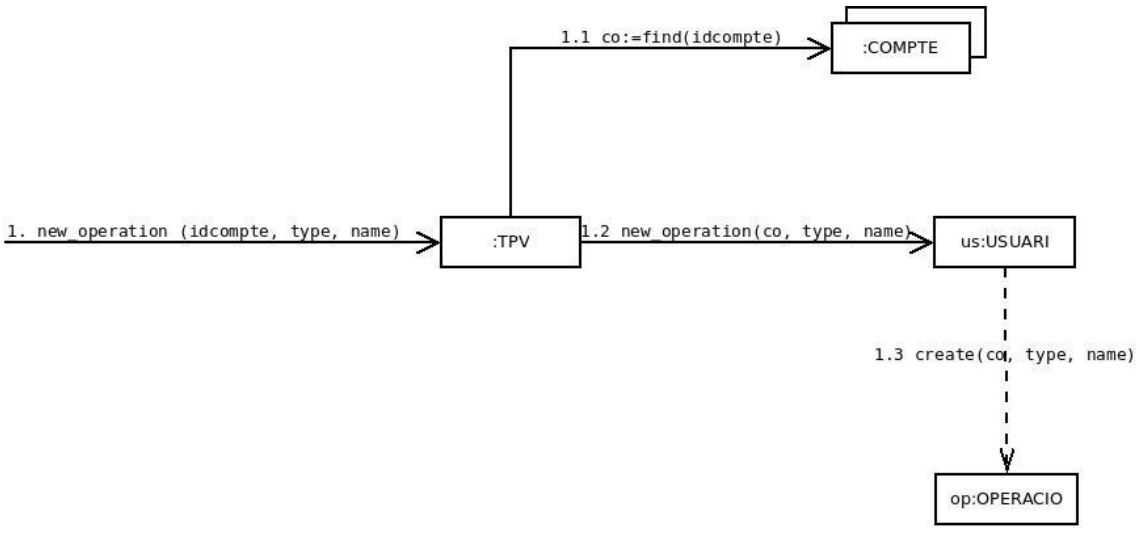


Figura 39: CU 10 – new operation

End_operation ( )

PRE:

1. Hi ha una *op* operació activa
2. Hi ha un *us* usuari actiu al sistema.
3. L'operació *op* té un nom, type i un compte associat.

POST:

1. S'emmagatzema l'*op* operació al sistema.
2. L'*op* operació ja no es activa al sistema.
3. L'usuari *us* es manté actiu al sistema.

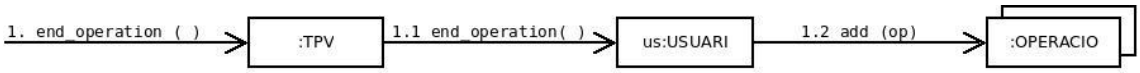


Figura 40: CU 10 – end operation

Diagrama de seqüència:

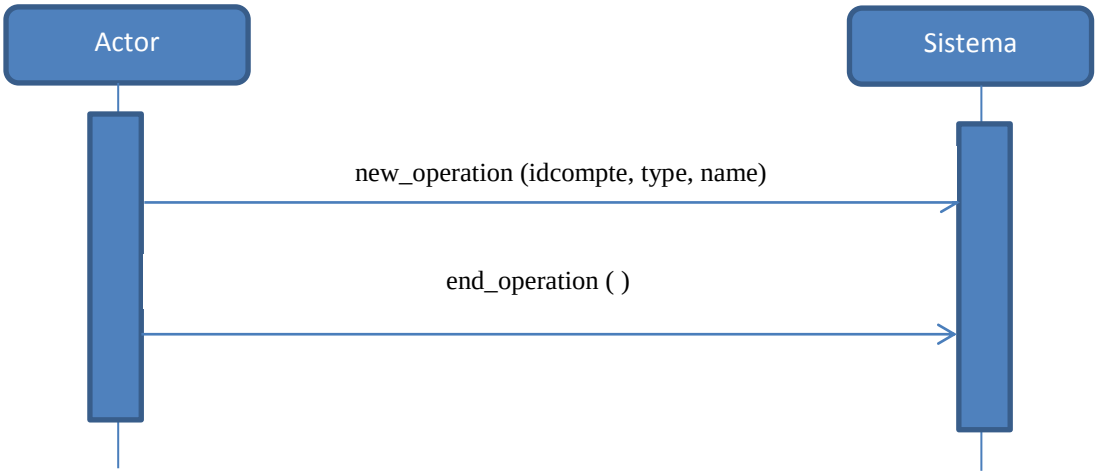


Figura 41: CU 10 – new operation

5.1.11. CU 11 – Introduir operació

Descripció: *Aquest cas d'ús, s'encarrega de crear un nou moviment de caixa.*

- Contractes:
    - **Into_operation** (idoperació, import, date, idreg)

Into_operation (idoperació, import, date, idreg)
Assistent d'introducció d'operacions.

PRE:

1. Existeix al sistema una *op* operació corresponent a l'identificador *idoperació.*
2. Existeix al sistema un *re* registre de caixa corresponent a l'identificador *idreg.*
3. Hi ha un *us* usuari actiu al sistema.
4. El *re* registre de caixa té l'estat *open.*

POST:

1. Es crea al sistema un nou *mc* moviment de caixa al sistema associat a la *op* operació, data i import a través del *re* registre de caixa.
2. Si l'operació *op* es de sortida l'import és multiplica per -1.
3. Si l'operació *op* es d'entrada l'import és positiu.
4. El *re* registre de caixa es manté actiu
5. L'usuari *us* es manté actiu al sistema.

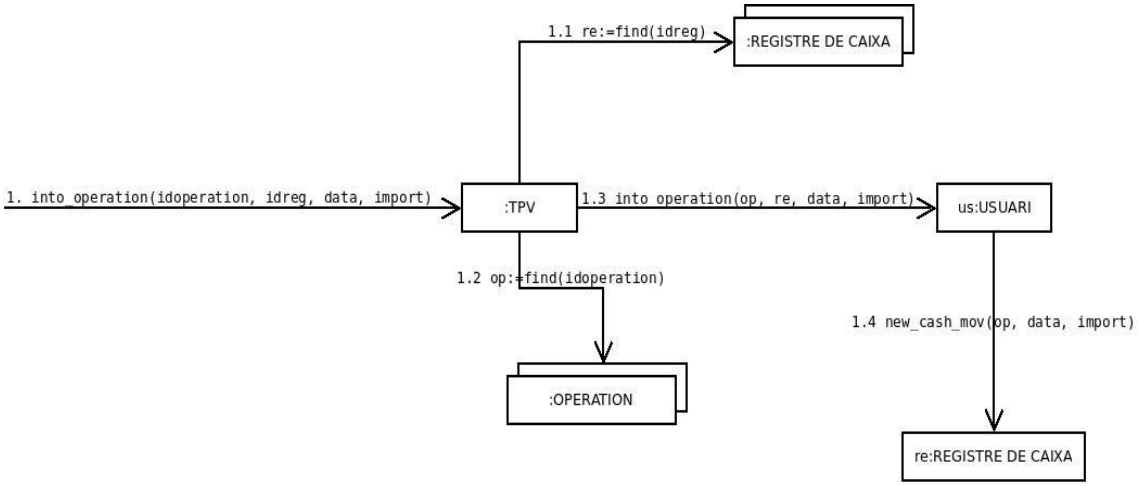


Figura 42: CU 11 – introduir operació

Diagrama de seqüència:

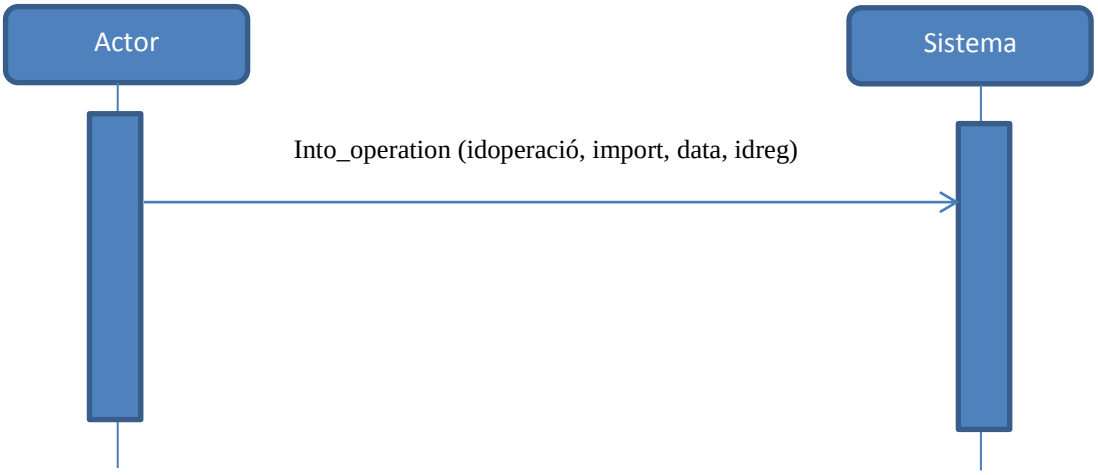


Figura 43: CU 11 – introduir operació

5.1.12. CU 12 – Configurar TPV

Descripció: *Aquest cas d'ús, s'encarrega de guardar o crear els elements que s'utilitzaran quan es crea una venda al sistema.*

- Contractes:
    - **Pos_config** ( )
    - **Afegir_info** ( )
    - **End config**( )

Pos_config ( )
Formulari d'introducció de dades. El magatzem i les adreces d'enviament i de facturació es mostren per defecte.

PRE:
1. No hi ha cap configuració al sistema.

POST:

1. S'ha creat al sistema una nova *cf* configuració.
2. La *cf* configuració es manté activa al sistema.

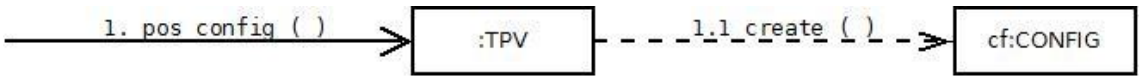


Figura 44: CU 12 – pos configuration

Afegir_info ( )

PRE:

1. Hi ha una *cf* configuració al sistema.

POST:

1. S'ha creat al sistema un nou *tr* tercer
2. S'ha creat al sistema un nou *pt* mètode de pagament.
3. El nou *tr* tercer es manté actiu al sistema.
4. El nou *pt* mètode de pagament es manté activa al sistema.
5. La *cf* configuració es manté activa al sistema.

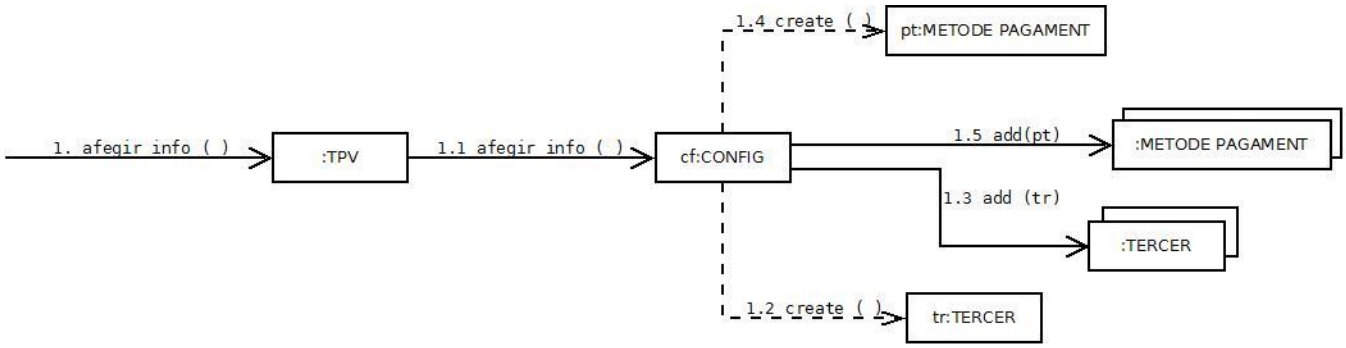


Figura 45: CU 12 – afegir informació

End_config ( )

PRE:

1. Hi ha un *tr* tercer actiu al sistema
2. Hi ha un *pt* mètode de pagament actiu al sistema.
3. Hi ha una *cf* configuració activa al sistema.

POST:

1. S'ha emmagatzemat al sistema el *tr* tercer actiu.
2. S'ha emmagatzemat al sistema el *pt* mètode de pagament actiu.
3. El *tr* tercer ja no es actiu al sistema.
4. El *pt* mètode de pagament ja no es actiu al sistema.
5. La *cf* configuració ja no es activa al sistema.

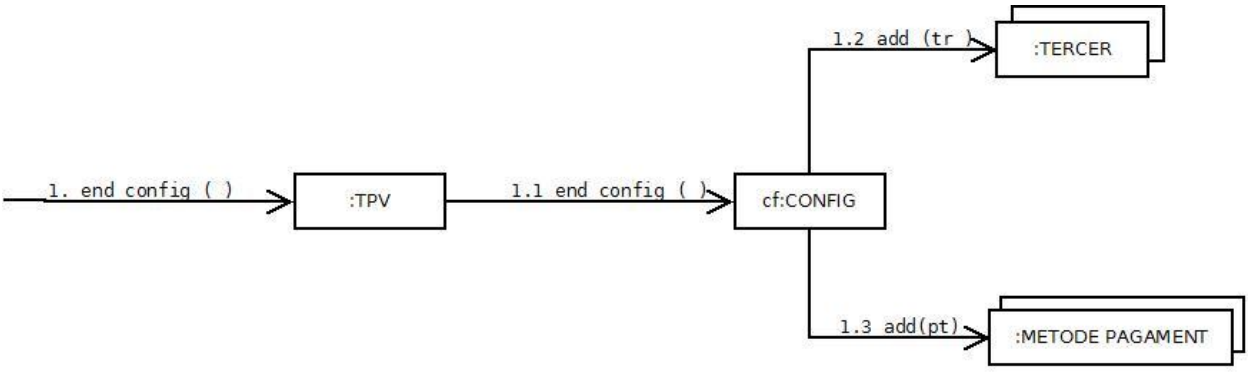


Figura 46: CU 12 – fi configuració

Diagrama de seqüència:

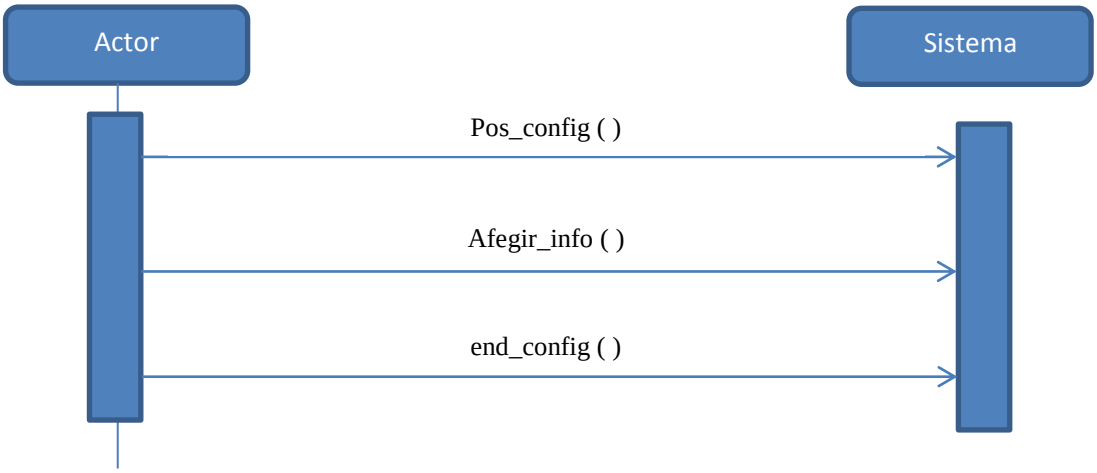


Figura 47: CU 11 – pos configuració

5.2. Model de components

Per major claredat s'ha dividit el model de components en dos. Si juntes tots dos models no s'entendria gaire a caixa de la gran quantitat de visibilitats que hi ha.

5.2.1. Model de component 1

A continuació es mostra el model de components 1. Mostra el flux per registrar moviments de caixa i d'actualització la comptabilitat de l'empresa.

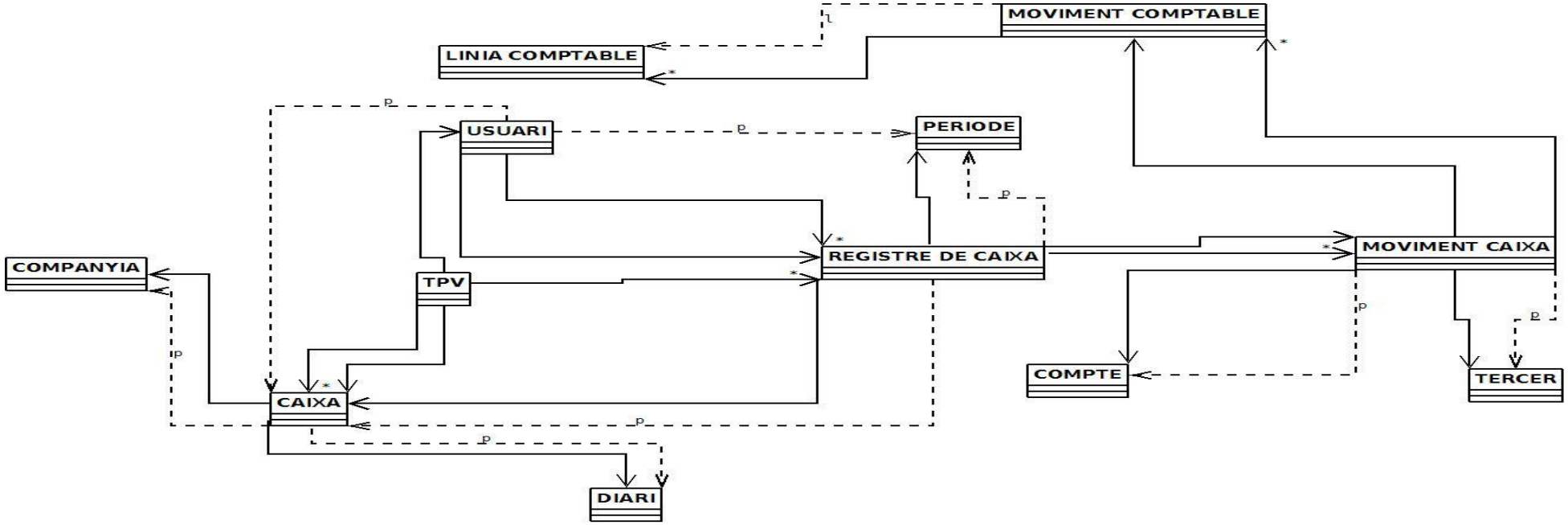


Figura 48: Model de components del mòdul account_cash

En el model de components anterior hi ha visibilitats entre components que no s'han representat per causes de senzillesa. Es tracta d'enllaços multi avaluats del controlador a companyia, a diari, a compte comptable i a període.

5.2.3. Mètodes dels components 1 apareguts.

Concepte	Mètodes
TPV	New_cash (idcompanyia, iddiari, name, moneda)
	End_cash ( )
	New_cash_reg (idperiode, idcaixa, name)
	End_cash_reg( )
	Open_cash_reg(idreg)
	New_cash_mov(idtercer, date, idcompte, import, origen, descripció).
	New_cash_mov(op, date, import).
	New_cash_mov(v, pg).
	End_cash_mov( )
	Close_cash_reg ( )
	Confirm_cahs_reg(cl_balance)
	Add_move ( )
	Actualitzar_caixa( )
	End_confirm ( )
Registre de caixa	Open_cash_reg(idreg)
	Name:=getSeqId( )
	Modificar_estat ( )
	New_cash_mov(idtercer, date, idcompte, import, origen, descripció).
	New_cash_mov(op, date, import).
	New_cash_mov(v, pg).
	End_cash_mov( )
	Saldo := get_total_amount ( )
	b:= comprova_balanç (cl_balance)
	Add_move ( )
	Actualitza_caixa ( )
	End_confirm ( )
USUARI	New_cash_reg (idperiode, idcaixa, name)
	End_cash ( )
	Open_cash_reg(idreg)
	Close_cash_reg ( )
	Confirm_cash_reg(cl_balance)
	Add_move ( )
	End_confirm ( )
MOVIMENTS CAIXA	Saldo:=get_amount ( )
	Confirm_cash_reg(cl_balance)
	Add_move ( )
	End_confirm ( )
CAIXA	
	Actualitza_caixa ( )

5.2.4. Model de components 2

A continuació es mostra el model de components 2 relacionat amb les vendes TPV i el seu pagament. Per claredat del model s'omet el controlador TPV.

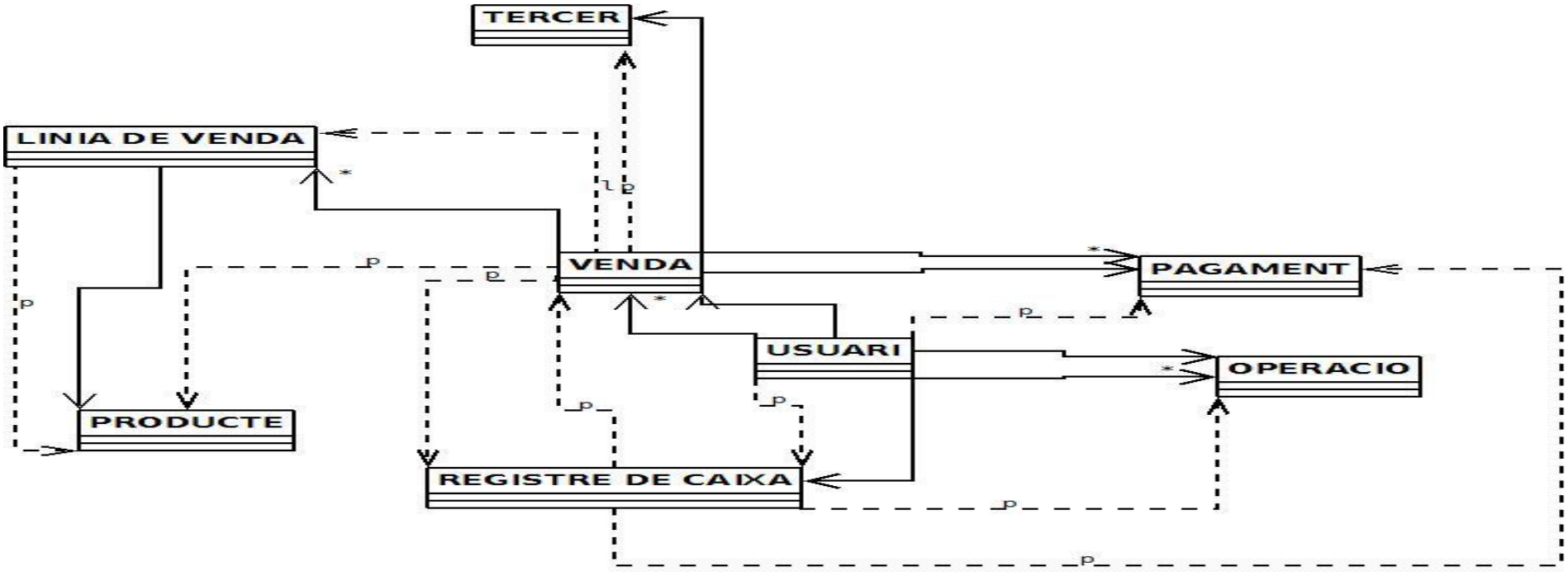


Figura 49: Model de components del mòdul pos

5.2.5. Mètodes dels components 2 apareguts

Concepte	Mètodes
TPV	New_sale(idtercer, descripció, idmoneda)
	Add_sale_line(idproduct)
	End_sale()
	New_pay(idreg, idvenda, import, date)
	Pay_sale()
	End_pay()
	Process_sale(idvenda)
	End_process ( )
	New_operation(idcompte, type, name)
	End_operation ( )
	Into_operation ( )
	Pos_config ( )
	Afegir_info ( )
	End_config ( )
VENDA	Add_sale_line(idproduct)
	Actualitza_import ( )
	Pg:=new_pay(re, import, data)
	Actualitza_import_a_pagar(import)
	Pay_sale()
	b:= compara_import( )
	Modifica_estat(confirmed)
	End_pay( )
	Process_sale ( )
	Create_invoices ( )
	End_process ( )
USUARI	New_sale(idtercer, descripció, idmoneda)
	Add_sale_line(idproduct)
	End_sale()
	New_pay(idreg, idvenda, import, date)
	Pay_sale()
	End_pay()
	Process_sale ( )
	End_process ( )
	New_operation(idcompte, type, name)
	End_operation ( )
	Into_operation ( )

5.3. Disseny modular

Al començament del projecte el mòdul desenvolupat per mi es composava per una caixa, vendes minoristes, un registre de caixa i operacions. A continuació es mostra l'estructura original del mòdul que es va desenvolupar a l'inici del projecte:

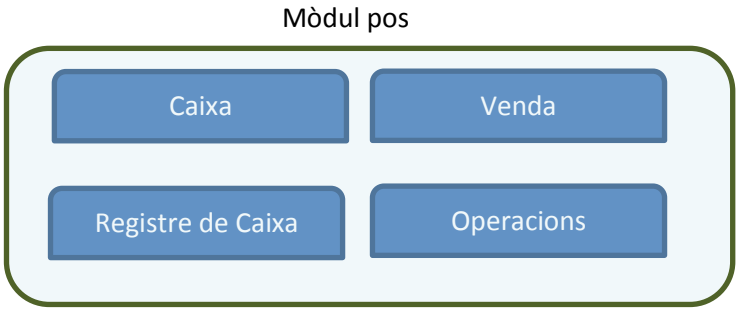


Figura 50: Diagrama de la estructura del mòdul pos

Mentre el desenvolupament del projecte avançava ens vam adonar que l'única relació que hi havia entre el registre de caixa, les vendes i les operacions consistia en que la caixa registradora s'encarrega d'enregistrar moviments referents a diners mentre que les vendes i les operacions generen moviments de diners. La informació en comú que comparteixen tots aquests elements només recau en els moviments de diners.

La divisió del mòdul es basa en la idea d'afegir dos mòduls diferents a Tryton, un que enregistra moviments de diners independentment de l'origen i que també guardi els diners en una caixa, mentre que l'altre mòdul s'encarrega de crear el sistema vendes i operacions.

Per acabar de decidir si la divisió del mòdul TPV era viable, vaig analitzar si podia aprofitar el desenvolupament fet fins al moment. Com que no era necessari realitzar gaires modificacions al codi, gràcies al baix acoblament del registre de caixa amb els altres components del sistema, vaig decidir que la divisió del mòdul TPV en dos mòduls independents es concretés.

El resultat de la divisió són dos mòduls, un anomenat ***pos*** i un altre anomenat ***account_cash.*** El següent gràfic mostra el resultat de la divisió del mòdul:

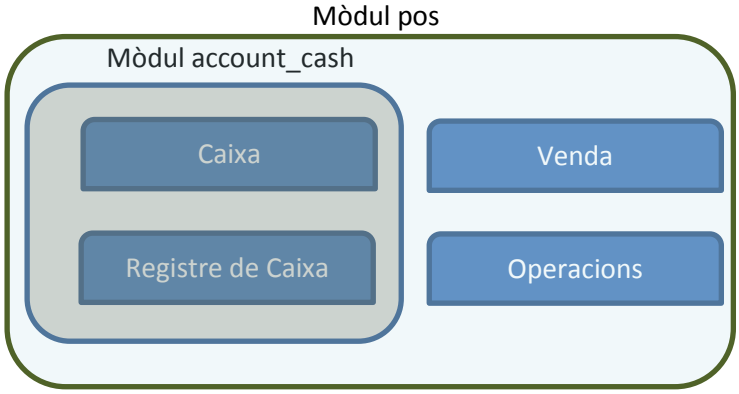


Figura 51: Diagrama dels mòduls

- **account_cash:**

  Aquest mòdul s'encarrega de crear una caixa i el registre de caixa on s'utilitzarà aquesta. Permet introduir tota classe de moviments de diners i comptabilitzar-los. Es composa per dos elements:

  - **Registre de caixa:** El registre de caixa és l'encarregat de crear i enregistrar moviments de diners.

    Cada moviment enregistrat esta associat, com a mínim, a un import de diners i un compte comptable. Quan es comptabilitzin els moviments anotats al registre s'actualitzarà automàticament la comptabilitat de la empresa creant un assentament comptable amb dos apunts o línies comptables..

  - **Caixa:** La caixa és l'encarregada de guardar la suma del valor dels imports de tots els moviments de diners enregistrats per un registre de caixa. S'utilitza per iniciar el registre de caixa amb l'últim saldo generat.

    Es per aquest motiu que un registre de caixa no pot enregistrar cap moviment de diners si no està associat a una caixa.

  A continuació es mostra un gràfic que es pot observar la relació que hi ha entre les caixes i els registres de caixa.

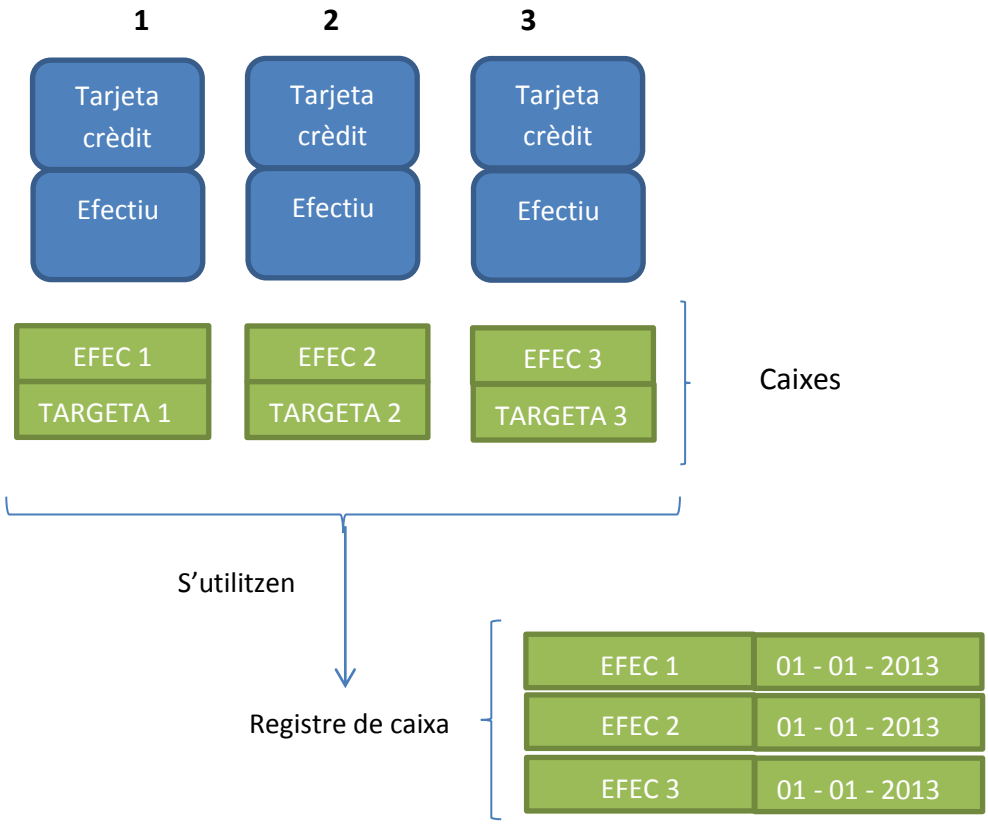


Figura 52: Relació entre les caixes i els registres de caixa

Ara bé, el mòdul a*ccount_cash* aprofita les funcionalitats ja implementades en altres mòduls de Tryton. Així doncs, per poder instal·lar el mòdul a*ccount_cash* es necessari instal·lar les seves dependències (mòduls). A continuació es detallen les dependències que té mòdul *account_cash*:

- ir: Aquest mòdul s'instal·la per defecte a Tryton; s'encarrega de subministrar les funcionalitats del framework de Tryton.

- res: El mòdul *res* també s'instal·la per defecte. S'encarrega de crear i de gestionar els grups, accions i permisos dels usuaris.

- account: Permet gestionar tota la comptabilitat de l'empresa (exercicis fiscals, períodes, comptes comptables, assentaments, apunts, balanços). És necessari per poder instal·lar el mòdul que gestiona les factures.

- account_product: Permet gestionar els comptes comptables dels productes.

- account_invoice: Aquest mòdul permet gestionar les factures de l'empresa. Depèn dels dos mòduls anteriors.

- **pos:**

Aquest mòdul s'encarrega de crear i enregistrar vendes i operacions d'entrada i sortida de diners i automatitza tots els processos relacionats.

El mòdul *pos* està compostat pels següents elements:

- **Venda:** Permet intercanviar bens amb un client a canvi d'una quantitat de diners. Genera moviments de diners que són enregistrats en un registre de caixa. A partir de la venda es genera l'albarà i la factura corresponents i actualitza l'estoc dels productes i la comptabilitat de l'empresa.

- **Operació entrada/sortida**: Són operacions prèviament introduïdes al sistema que s'utilitzen esporàdicament sense estar associades a cap procés de venda. Aquestes operacions, quan s'utilitzen, generen moviments de diners enregistrats per un registre de caixa.

El mòdul *pos*, igual que el mòdul *account_cash* utilitza funcionalitats ja implementades en altres mòduls. Per realitzar la instal·lació del mòdul *pos* també cal instal·lar les seves dependències. Aquestes són:

- account_cash: El mòdul pos depèn d'*account_cash*. Hereta totes les seves funcionalitats. A més, quan s'instal·la aquest mòdul, automàticament també s'instal·len totes les seves dependències, per tant, s'instal·len tots els components per poder crear una factura des de les vendes del TPV.

- sale: Aquest mòdul instal·la la funcionalitat que permet gestionar vendes majoristes on el procés de realització es diferent al procés de vendes creades en el mòdul TPV (minoristes). El mòdul tpv hereta algunes funcions de sale.

- stock: Gestiona tots els moviments de productes al magatzem. Es necessari per poder crear albarans de client i actualitzar l'estoc de productes.

Aprofitant el mòdul existent per gestionar els usuaris i els seus permisos, quan s'instal·la un dels dos mòduls, es defineixen al sistema dos grups d'usuaris per a cada mòdul:

- **Mòdul account_cash:**

  - Administrador Account cash: Els usuaris que necessiten permisos per crear caixes al sistema pertanyen a aquest grup. Per tant es pot considerar que els usuaris d'aquest grup són administradors del mòdul *account_cash.* Els usuaris d'aquest grup també han de tenir permís per poder crear assentaments comptables.

  - Account_cash: En aquest grup es troben els usuaris que han d'utilitzar el registre de caixa però que no tenen permís per crear caixes al sistema. Els usuaris d'aquest grup també han de tenir permís per poder crear assentaments comptables.

- **Mòdul pos:**

  - Administrador Pos: És el grup d'usuaris que pot configurar el mòdul així com també crear noves operacions i caixes. Per dependència directa també ha de ser membre del grup d'usuaris que té accés de lectura i escriptura sobre el mòdul de comptabilitat per poder llegir i modificar les caixes existents i poder actualitzar la comptabilitat.

  - Pos: Grup d'usuaris que només utilitza el mòdul pos. No pot configurar el mòdul ni tampoc crear caixes però ha de tenir permisos de lectura i escriptura sobre el mòdul de logística i comptabilitat per poder crear albarans i factures respectivament i actualitzar la comptabilitat.

Durant el projecte no es diferència entre l'ús del mòdul *account_cash* i *pos.* Només es fa referència al mòdul *pos* perquè des d'aquest es pot accedir a les funcionalitats que ofereix *account_cash.* En cas contrari s'especifica detalladament.

5.3.1. Caixa

El saldo emmagatzemat a la caixa s'utilitza per calcular el saldo inicial del registre de caixa quan s'inicialitza. Cada cop que aquest últim finalitza el seu propòsit, s'actualitza el saldo de la caixa. El valor dels moviments de diners es calcula de la següent forma:

Saldo inicial = Saldo inicial + (Operacions realitzades + entrada de diners – sortida de diners)

El saldo inicial és el valor que s'emmagatzema a la caixa. El primer cop que s'utilitza una caixa al registre de caixa el saldo inicial és zero. Les operacions realitzades corresponen als moviments de diners generats per les vendes i enregistrats directament al registre de caixa. Entrada i sortida de diners són moviments de diners generats a partir d'operacions d'entrada i sortida o moviments de diners introduïts manualment al registre de caixa.

La caixa també està associada a un diari comptable. Aquest últim disposa de dos comptes comptables, un per el deure i l'altre per l'haver. Per aquest motiu la caixa també és l'element que permet al registre de caixa actualitzar la comptabilitat de la companyia utilitzant els comptes comptables del diari comptable associat a la caixa.

5.3.2. Registre de caixa

En cada estat es pot processar la informació enregistrada de la manera adequada i actualitzar altres elements, com per exemple la comptabilitat de l'empresa. A continuació es detallen els estats que el registre de caixa pot adoptar i el procés que es realitza en cadascun d'ells:

- **Draft:** Aquest és l'estat per defecte i no permet enregistrar cap tipus de moviment de diners. En aquest estat només es pot introduir la informació necessària per iniciar (obrir) el registre de caixa i així poder enregistrar moviments de diners. De l'estat *draft* pot adoptar l'estat *open.*

- **Open:** Quan el registre de caixa es troba en aquest estat, pot enregistrar moviments de diners, també anomenats *moviments de caixa*. Els moviments de caixa són generats per vendes, operacions d'entrada i sortida de diners o introduïts en el registre directament per l'usuari.

  Si el registre de caixa té aquest estat pot tornar a adoptar l'estat *draft* en el cas que calgui modificar alguna dada d'inici del registre de caixa. També pot adoptar l'estat *close.*

- **Close:** En aquest estat el registre de caixa no pot enregistrar cap moviment de caixa. Es manté a l'espera de que l'usuari compti l'acumulat de tots els moviments de caixa enregistrats i l'introdueixi al registre de caixa. El saldo que s'ha d'introduir ha de ser igual al saldo de tancament generat pel sistema. El saldo de tancament es calcula de següent manera:

  Saldo de tancament = suma valor moviments de caixa + saldo inicial

  En el moment que el saldo introduït per l'usuari quadri amb el saldo de tancament, el registre de caixa podrà adoptar l'estat *confirmed*. També, sense

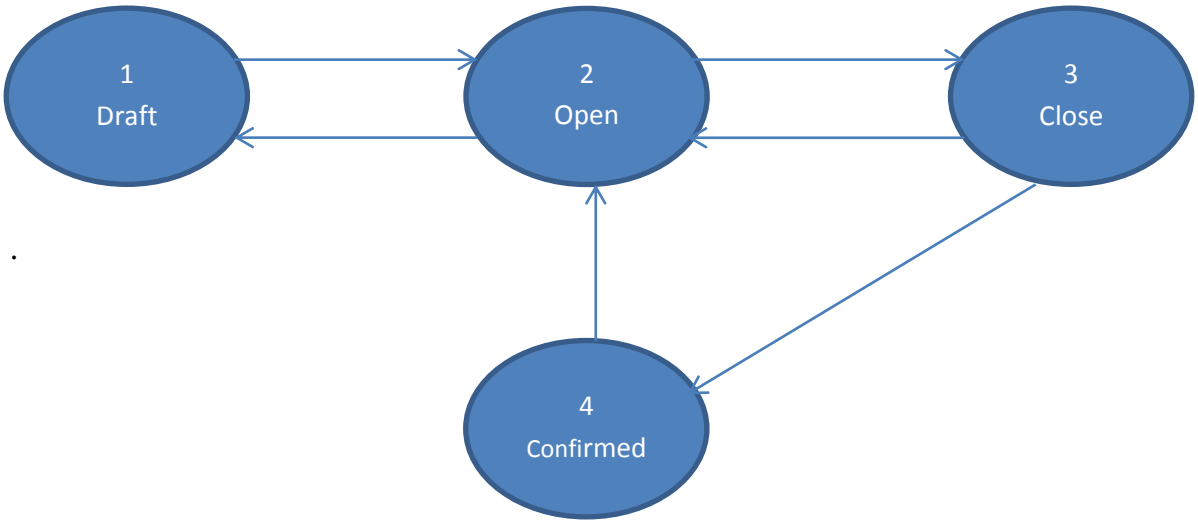
necessitat de que l'usuari introdueixi cap saldo, es pot tornar a obrir el registre de caixa (tornar a l'estat *open*) per tornar a enregistrar més moviments de caixa.

- **Confirmed:** Quan un registre de caixa adopta aquest estat, es comprova si el saldo de tancament introduït per l'usuari quadra amb el saldo de tancament generat pel registre. Si la quantitat introduïda és igual al saldo de tancament que ha generat el registre de caixa, s'inicià el procés d'actualització de la comptabilitat.

  En aquest procés també s'actualitza el saldo emmagatzemat en la caixa amb el saldo de tancament. Així el proper cop que s'utilitzi la mateixa caixa en qualsevol registre de caixa, aquest s'obrirà amb un saldo inicial actualitzat.

  El registre de caixa, un cop ha adoptat l'estat *confirmed* pot tornar a l'estat *open.*

A continuació es presenta un gràfic que mostra el workflow associat al registre de caixa.



.

Figura 53: Diagrama d'estats del registre de caixa

5.3.3. Moviments de caixa

Com he comentat anteriorment, el registre de caixa s'ocupa d'enregistrar moviments de diners a la caixa. Per aquest motiu, el registre de caixa conté un conjunt de moviments de caixa. Cada moviment de caixa és un conjunt d'informació detallada sobre cada operació de diners enregistrada. En aquesta aplicació els moviments de caixa poden ser de tres tipus:

- **Venda:** Aquests moviments de caixa es generen cada cop que es realitza un/a pagament/devolució d'una venda. Contenen el màxim d'informació possible per maximitzar el detall de la venda.

- **Operació entrada/sortida:** S'utilitzen per la introducció o retirada de diners a causa d'esdeveniments externs al sistema. Per exemple, "ingrés al banc". Contenen menys informació que els moviments generats per una venda perquè només cal un compte comptable i una quantitat de diners.

- **Registre de caixa:** Són moviments de caixa que genera l'usuari des del formulari del propi registre de caixa. S'utilitzen per enregistrar operacions de diners ràpides. Per exemple "pagament al servei de neteja"

Quan el registre de caixa arriba a l'estat *confirmed*, cada moviment de caixa modifica el seu estat, de *draft* a *confirmed* i genera un assentament comptable amb dos apunts o línies comptables. L'objectiu és automatitzar el procés d'actualització de la comptabilitat de la nostra empresa. Els estats que poden adoptar els moviments de caixa són:

- **Draft:** El moviments que tenen aquest estat encara no han generat l'assentament comptable amb dos apunts o línies comptables.

- **Confirmed:** Els moviments de caixa en estat *confirmed*, ja han generat han generat l'assentament comptable amb dos apunts o línies comptables.

5.3.4. Venda

Els estats definits en el workflow associat a la venda són:

- **Draft:** Quan la venda es troba en aquest estat es poden enregistrar línies de venda. Mitjançant aquestes, es realitzen els càlculs per saber el valor de la venda. Aquesta es mantindrà en aquest estat fins que sigui pagada a través d'un mecanisme (basat en assistents) que automatitza el procés de pagament; quan finalitza quest procés es crea automàticament al sistema l'albarà corresponent. De l'estat *draft,* la venda pot adoptar l'estat *confirmed.*

- **Confirmed:**. És l'estat que adopta la venda quan s'ha pagat completament. Correspon a l'estat que indica que ja s'ha creat l'albarà al sistema. De l'estat *confirmed* la venda només pot adoptar l'estat *processed.*

- **Processed:** Per finalitzar el workflow definit per la venda, aquesta ha d'adoptar aquest estat. Genera una nova factura al sistema associada a la venda per així comptabilitzar la venda.

Per optimitzar la gestió de les vendes s'ha desenvolupat una funcionalitat que permet seleccionar un conjunt de vendes que tinguin en comú el mateix tercer i que es trobin en l'estat confirmed per crear una única factura. Al finalitzar aquest procés, les vendes implicades modifiquen el seu estat de *confirmed* a *processed.* També es pot generar un

tiquet de venda en format *odt* on es mostra el resum de la venda, (línies de venda, cost de la venda i pagaments realitzats).

A continuació es mostra un diagrama corresponent al workflow associat a la venda:

Figura 54: Diagrama d'estats del registre de caixa

5.3.5. Línies de venda

Una línia de venda es composa per un producte, el número d'unitats del producte, impostos relacionats, preu unitari i cost total. Amb aquesta informació es pot calcular el cost total de la venda. També proporcionen la informació necessària per actualitzar l'estoc dels productes inclosos a les línies de venda. Cada cop que s'introdueix una nova línia de venda es re-calculen els imports de la venda. A continuació es mostra com es calcula el valor total de la venda:

$$Total_venda = (Sum(preu\ producte * quantitat)) + valor_impost$$

El *valor_impost* correspon al resultat d'aplicar l'impost (per exemple el IVA) sobre *sum(preu producte * quantitat).*

5.3.6. Pagament/abonament de la venda

Una línia de venda es composa per un producte, el número d'unitats del producte, impostos relacionats, preu unitari i cost total. Amb aquesta informació es pot calcular el cost total de la venda. També proporcionen la informació necessària per actualitzar l'estoc dels productes inclosos a les línies de venda. Cada cop que s'introdueix una nova línia de venda es re-calculen els imports de la venda. A continuació es mostra com es calcula el valor total de la venda:

$$Total_venda = (Sum(preu\ producte * quantitat)) + valor_impost$$

El *valor_impost* correspon al resultat d'aplicar l'impost (per exemple el IVA) sobre *sum(preu producte * quantitat).*

5.3.7. Operacions

S'utilitzen per realitzar operacions esporàdiques amb diners de la caixa. Per exemple, retirar diners per ingressar-ho al banc o al revés, ingressar diners com monedes per poder tenir canvi. El registre de caixa anota cada operació realitzada per actualitzar posteriorment la comptabilitat de l'empresa. Les operacions poden ser de dos tipus:

- **Entrada:** Les operacions d'aquest tipus introdueixen diners a la caixa sense estar associada a cap procés de la venda. Per exemple, introduir canvi a la caixa.

- **Sortida:** Aquest tipus d'operació anota al registre de caixa una sortida de diners. Per exemple, retirada de diners per ingrés al banc.

5.4. Disseny de la base de dades

A continuació es mostren les estructures de dades corresponents a cada mòdul:

- **account_cash:**

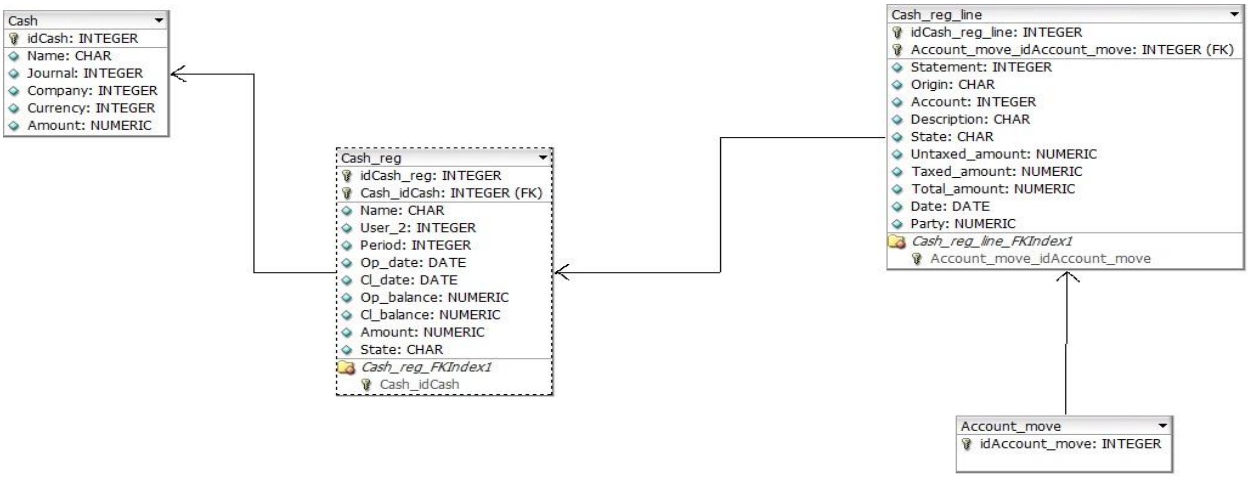


Figura 55: Estructura de dades del mòdul account_cash

- **pos:**

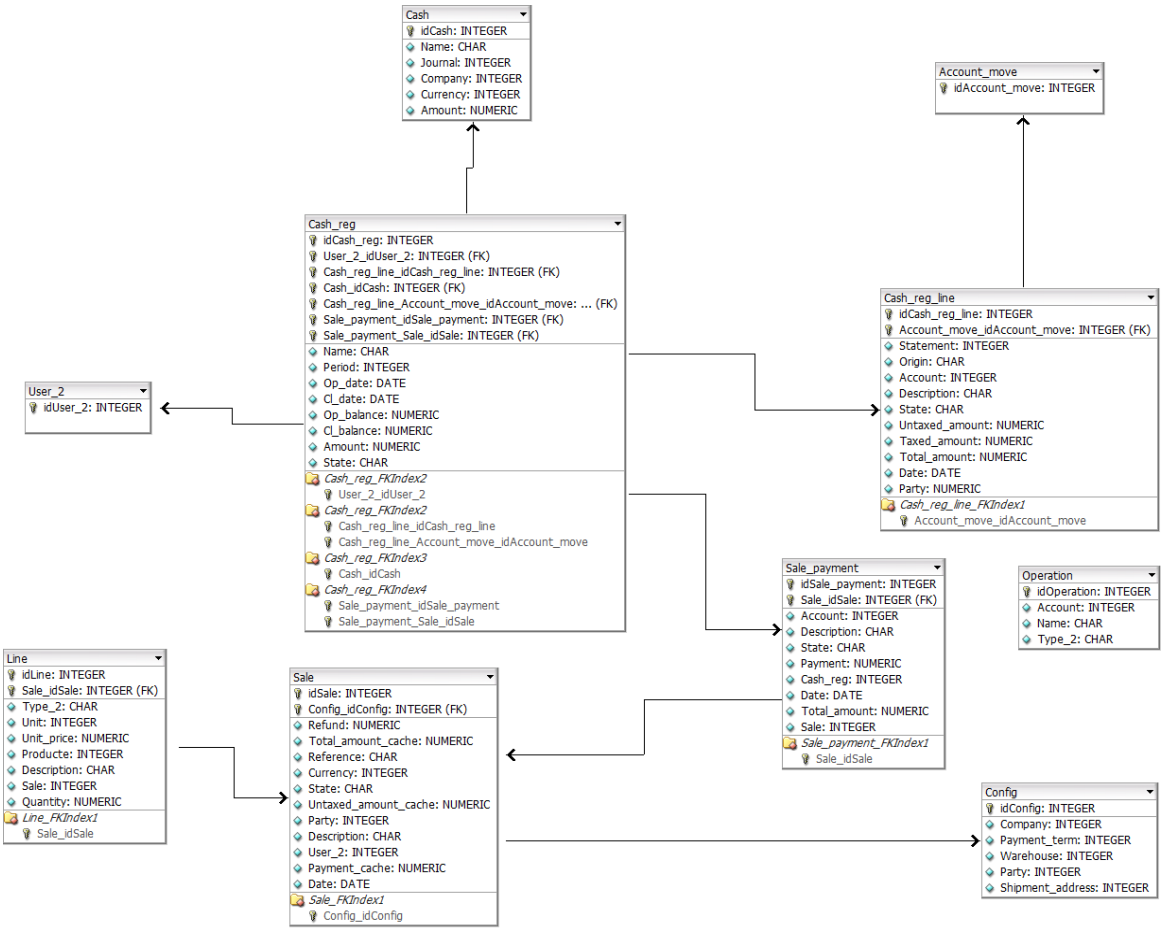


Figura 56: Estructura de dades del mòdul pos

5.4.1. Taules de la base de dades

A continuació es mostren les taules més importants que s'han dissenyat pel projecte. Estan dividides per mòduls, primer es mostren les taules relacionades amb el mòdul account_cash i posteriorment les taules corresponents al mòdul pos.

- **account_cash:**

Taula	Cash_reg	
Descripció	Emmagatzema els registre de caixa existents i tota la seva informació.	
Clau	Id, usuari període, caixa	
Camps	**Descripció**	**Tipus**
ID	Identificador del registre de caixa.	Numeric
Nom	Indica el nom del registre de caixa.	Char
Usuari	ID de l'usuari associat al registre de caixa.	Many2One
Període	ID del període associat al registre de caixa.	Many2One
Caixa	ID de la caixa relacionada al registre de caixa.	Many2One
Op_date	Data de l'última obertura del registre de caixa.	Date
Cl_date	Data de tancar el registre de caixa.	Date
Op_balance	Indica el saldo d' obertura del registre de caixa.	Numeric
Cl_balance	Indica el saldo de del registre de caixa.	Numeric
Amount	Indica l'import que ha introuit l'usuari i si és igual al cl_balanç	Numeric
State	Indica l'estat actual del registre de caixa.	Selection
Taules relacionades		
User		
Period		
Cash		

Taula	Cash_reg_line	
Descripció	Emmagatzema els moviments de caixa.	
Clau	ID, statement, account, move, party	
Camps	**Descripció**	**Tipus**
ID	Identificador de cada moviment de caixa.	Numeric
Statement	Registre de caixa relacionat.	Many2One
Origin	Indica l'origen del moviment de caixa, pot ser operació, venda o registre.	Selection
Account	Compte comptable per poder crear els assentaments comptables.	Many2One
Description	Descripció del moviment de caixa.	Char
Move	Moviment comptable associat.	Many2One
State	Indica l'estat actual del moviment de caixa.	Selection
Untaxed_amount	Suma total del valor dels moviments de caixa sense impostos.	Numeric
Taxed_amount	Total impostos.	Numeric

Total_amount	Suma total de untaxed_amount + taxed_amount.	Numeric
Date	Indica la data en que s'ha enregistrat en el registre	Date
Party	Tercer associat al moviment de caixa	Many2One
Taules relacionades		
Party		
Move		
Account		
Cash_reg		

Taula	Cash	
Descripció	Emmagatzema les caixes creades al sistema	
Clau	ID, journal, company, currency	
Camps	**Descripció**	**Tipus**
ID	Identifica la caixa.	Numeric
Name	Indica el nom de la caixa.	Char
Journal	Indica el diari associat a la caixa.	Many2One
Company	Empresa associada a la caixa.	Many2One
Currency	Indica la moneda en la que treballa la companyia.	Many2One
Amount	Últim saldo de tancament d'aquesta caixa.	Numeric
Taules relacionades		
Journal		
Company		
Currency		

- **pos**

Taula	Config	
Descripció	Emmagatzema la informació de configuració de les vendes TPV	
Clau	Id, company, payment_term, party, warehouse	
Camps	**Descripció**	**Tipus**
ID	Identifica el registre de la configuració.	Numeric
Company	Empresa associada al pos.	Many2One
Payment_term	Termini de pagament necessari per crear la factura de la venda.	Many2One
Party	Tercer que es mostrarà per defecte en la venda.	Many2One
Warehouse	Indica el magatzem on resideixen tots els productes i on s'aplicarà la actualització de l'estoc.	Many2One
Shipment_address	Indica la adreça d'enviament. S'utilitza per crear l'albarà al pagament de la venda.	Many2One
Taules relacionades		
Company		
Payment_term		
Party		
Warehouse		

Shipment_address

Taula	Operation	
Descripció	Emmagatzema les operacions d'entrada i sortida	
Clau	ID, account	
Camps	**Descripció**	**Tipus**
ID	Identificador de l'operació.	Numeric
Account	Compte comptable associat a l'operació.	Many2One
Name	Nom de l'operació d'entrada o sortida.	Char
Type	Indica el tipus d'operació. Pot ser IN/Out.	Selection
Taules relacionades		
Account		

Taula	Sale	
Descripció	Guarda les vendes creades al sistema i la seva informació	
Clau	ID, reference, currency, party, user	
Camps	**Descripció**	**Tipus**
ID	Identificador de la venda minorista.	Numeric
Refund	Mostra l'import que s'ha pagat de la venda minorista.	Numeric
Total_amount_cache	Guarda temporalment l'import total de la venda minorista.	Numeric
Reference	Indica la referencia de la venda minorista. Es genera automàticament	Numeric
Currency	Indica la moneda en la que es realitza la venda minorista.	Many2One
State	Indica l'estat actual de la venda minorista. Pot ser esborrany, confirmada, processada.	Selection
Untaxed_amount_cache	Guarda temporalment l'import sense impostos aplicats de la venda minorista.	Numeric
Party	Indica el client associat a la venda minorista. Si s'ha configurat es mostrarà un client per defecte	Many2One
Description	Descripció, si cal, de la venda minorista.	Char
User	Usuari que crea la venda minorista al sistema.	Many2One
Payment_cache	Guarda temporalment el pagament de la venda.	Numeric
Tax_amount_cache	Guarda temporalment l'import corresponent a l'aplicació d'impostos.	Numeric
Date	Indica la data de creació de la venda	Date
Taules relacionades		
Currency		
User		
Party		

Taula	Line	
Descripció	Emmagatzema les línies associades a cada venda.	
Clau	ID, product, sale	
Camps	**Descripció**	**Tipus**
ID	Identificador de la línia de venda.	Numeric
Type	Indica el tipus de línia de venda que és.	Selection
Unit	Indica la unitat en que es medeix el producte. Per exemple Kilogram	Many2One
Unit_price	Indica el preu associat al producte.	Numeric
Product	Producte associat a la línia de venda.	Many2One
Description	Descripció del producte.	Char
Sale	Indica la venda minorista associada a cada línia de venda.	One2Many
Quantity	Indica el numero d'unitats d'un mateix producte que s'inclouen a la línia de venda	Numeric
Taules relacionades		
UdM		
Product		
Sale		

Taula	Sale_payment	
Descripció	Emmagatzema els pagaments que es realitzen de una venda.	
Clau	ID, account, sale, cash_reg, party	
Camps	**Descripció**	**Tipus**
ID	Identificador del pagament/abonament.	Numeric
Account	Compte comptable associat a un pagament.	Many2One
Description	Descripció associada a l'origen d'una operació.	Char
State	Estat en que es troba el pagament/abonament	Selection
Sale	Venda minorista que genera el pagament.	Many2One
Payment	Import del pagament/abonament.	Numeric
Cash_reg	Registre de caixa associat al pagament. És el registre de caixa on es crea el moviment de caixa a partir del pagament.	Many2One
Date	Indica la data de realització del pagament/devolució.	Date
Party	Indica el client associat a un pagament.	Many2One
Total_amount	Indica el total a pagar.	Numeric
Taules relacionades		
Account		
Sale		
Cash_reg		
Party		

5.5. Interfícies gràfiques de l'usuari

A continuació es mostren les interfícies gràfiques dels mòduls. En primer lloc es mostren les interfícies corresponents al mòdul account_cash:

- **account_cash:**
    - Caixa:

La següent figura mostra la vista on es poden veure totes les caixes existents en el sistema. Es poden crear i consultar des de tots dos mòduls.

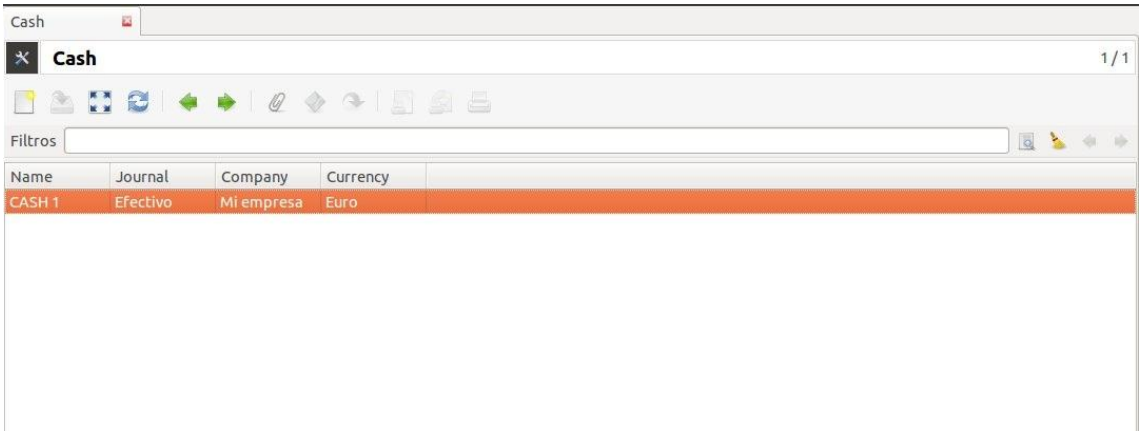


Figura 57: Llista de caixa

A continuació es presenta el formulari corresponent a la caixa. El disseny de la interfície es limita pel framework de Tryton. Es pot observar que tots els camps són requerits excepte el camp que guarda l'últim saldo de tancament.

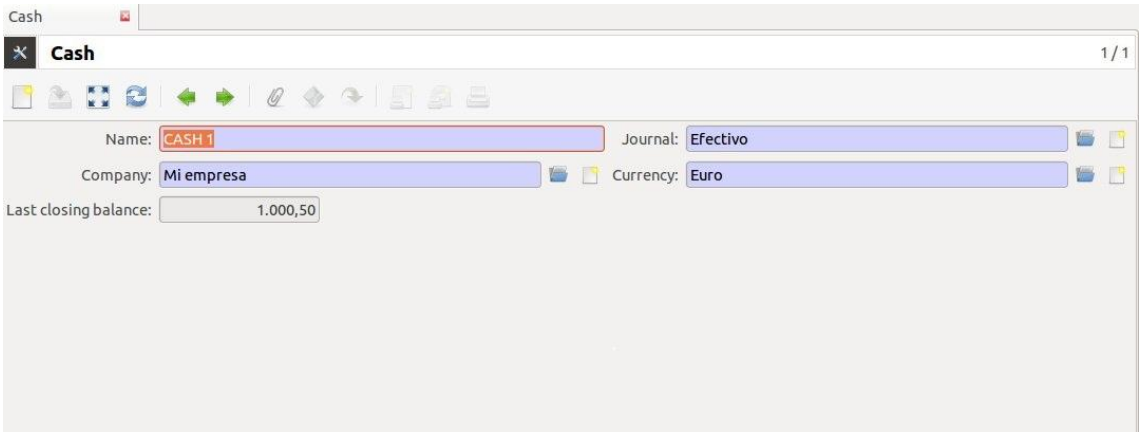


Figura 58: Formulari de caixa

- Registre de caixa:

A continuació es mostra la imatge corresponent a la llista de registres de caixa. Aquesta manera de mostrar la informació és molt característica de Tryton.

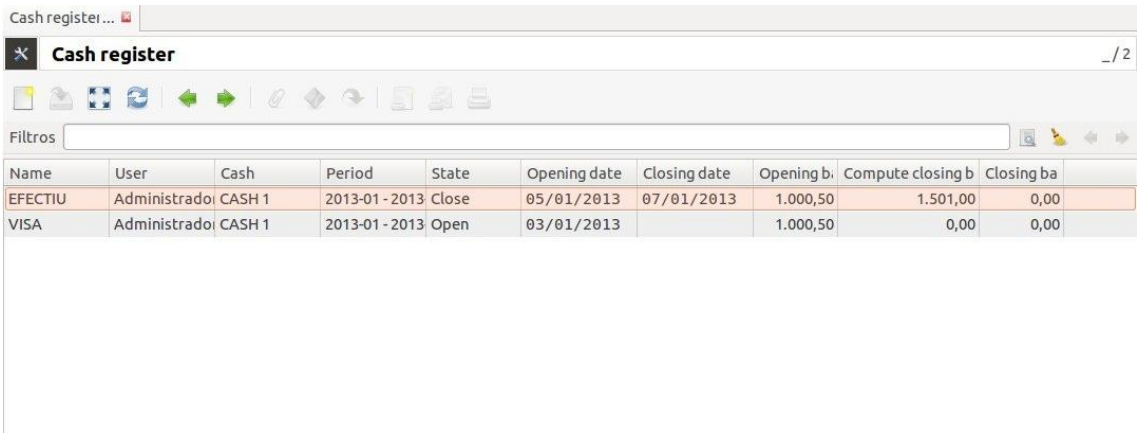


Figura 59: Llista de registres de caixa

A continuació es mostra una imatge referent al formulari del registre de caixa. Com es pot observar esta composada per dos pestanyes, una que mostra els moviments de caixa i una altre que mostra els apunts comptables d'una assentament comptable.

Si ens fixem en la imatge podem observar que les dades principals del registre de caixa es estan en la part superior de la interfície mentre que, els camps referents a saldos i dates estan situats en la part inferior del formulari.

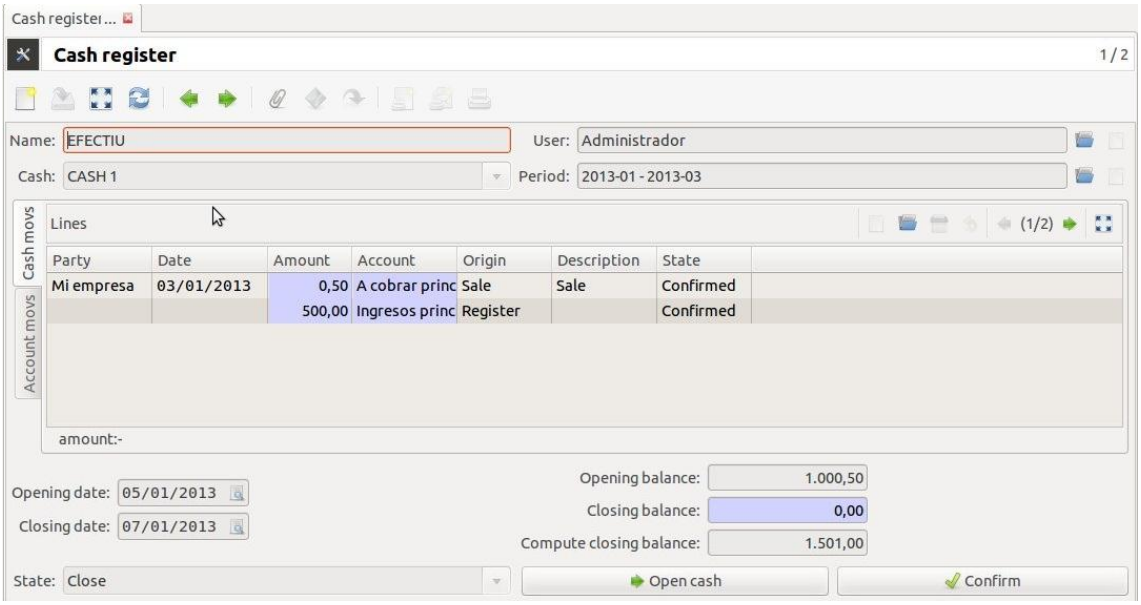


Figura 60: Formulari del registre de caixa

La imatge que es mostra a continuació correspon a la pestanya que mostra els apunts comptables generats a partir dels moviments de caixa registrats al registre de caixa.

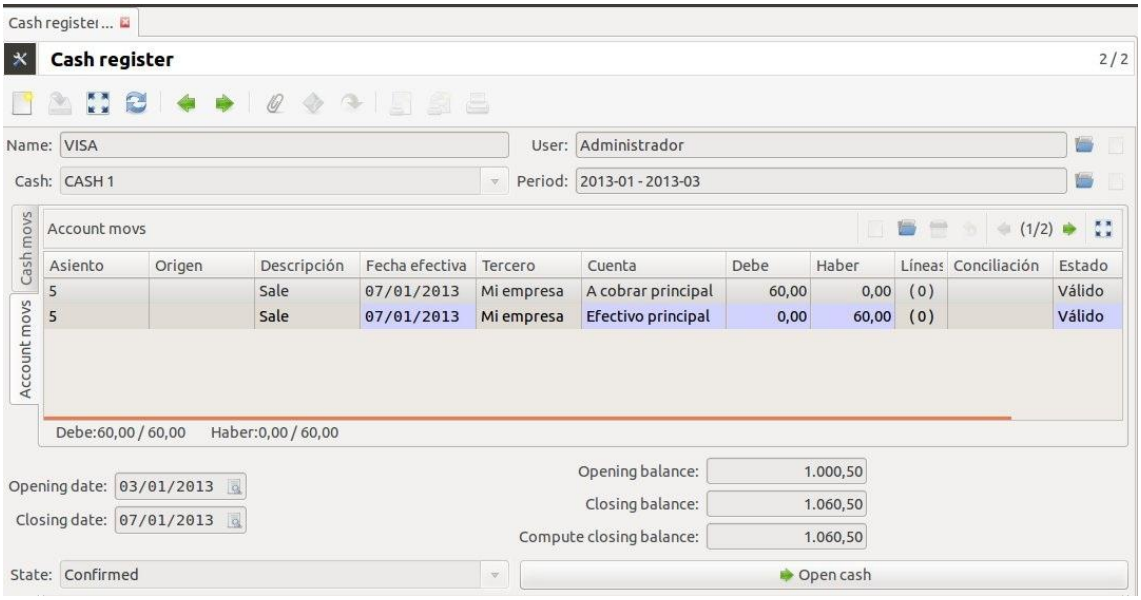


Figura 61: Apunts comptables generats pel registre de caixa

- **pos:**

A continuació es mostren totes les interfícies corresponents al mòdul pos. Les interfícies anteriors s'hereten al mòdul pos, per tant, són les mateixes que les interfícies mostrades anteriorment.

  - <u>pos configuration:</u>

La següent imatge mostra el formulari que permet configurar alguns aspectes del mòdul TPV com per exemple, crear un client que es mostrarà per defecte en les vendes minoristes. Per fer que aquest formulari sigui més entendible s'ha dividit en seccions:

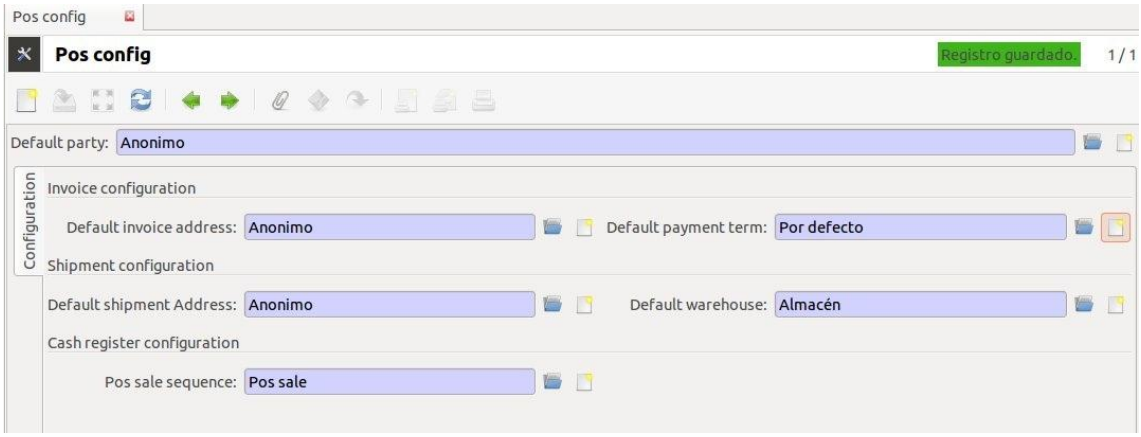


Figura 62: Formulari de la configuració del mòdul pos

- Operacions:

La següent imatge mostra la llista corresponent a les operacions existents al sistema:

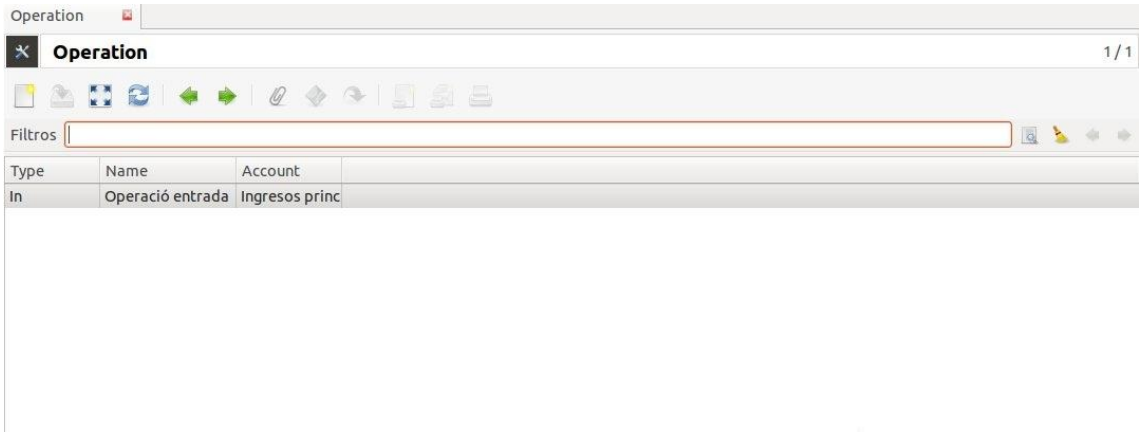


Figura 63: Llista d'operacions

Si s'observa la següent imatge es pot observar que, el formulari està composat per tres camps, un de tipus selecció (permet escollir el tipus d'operació que és), el nom que rep l'operació i el compte comptable associat. Tots els camps són requerits.

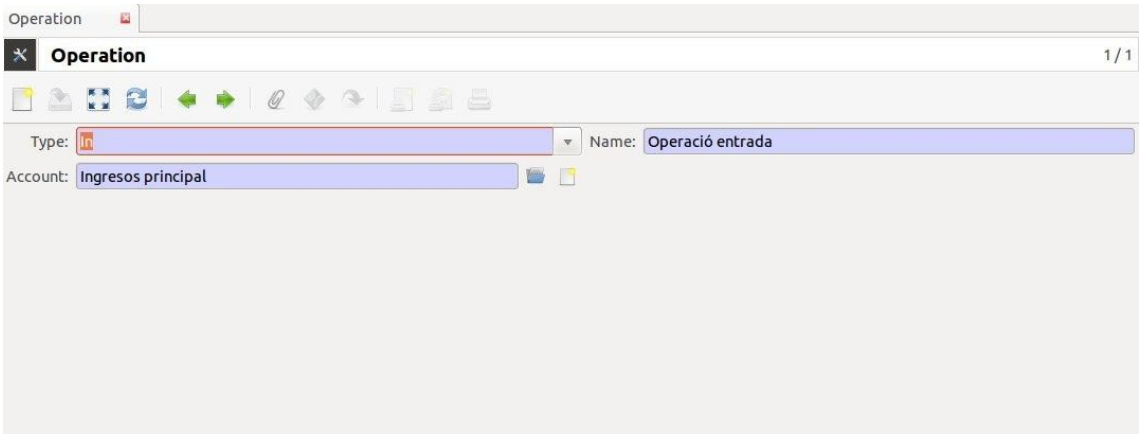


Figura 64: Formulari d'operacions

- Venda:

La següent imatge mostra la llista de vendes minoristes existents al sistema.

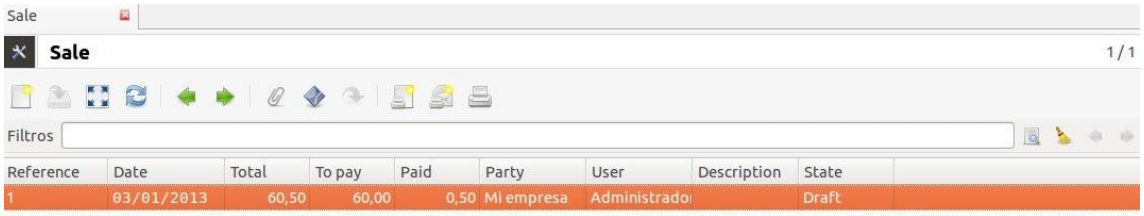


Figura 65: Llista de vendes

A continuació es mostra la imatge corresponent al formulari de vendes minoristes. Es pot observar com la informació referent a la venda s'ha agrupat en la part superior de la interfície mentre que, en la part inferior del formulari s'han agrupat tots els camps referents al costos. En mig del formulari hi ha tres pestanyes, la pestanya que mostra la figura correspon a les línies de venda.

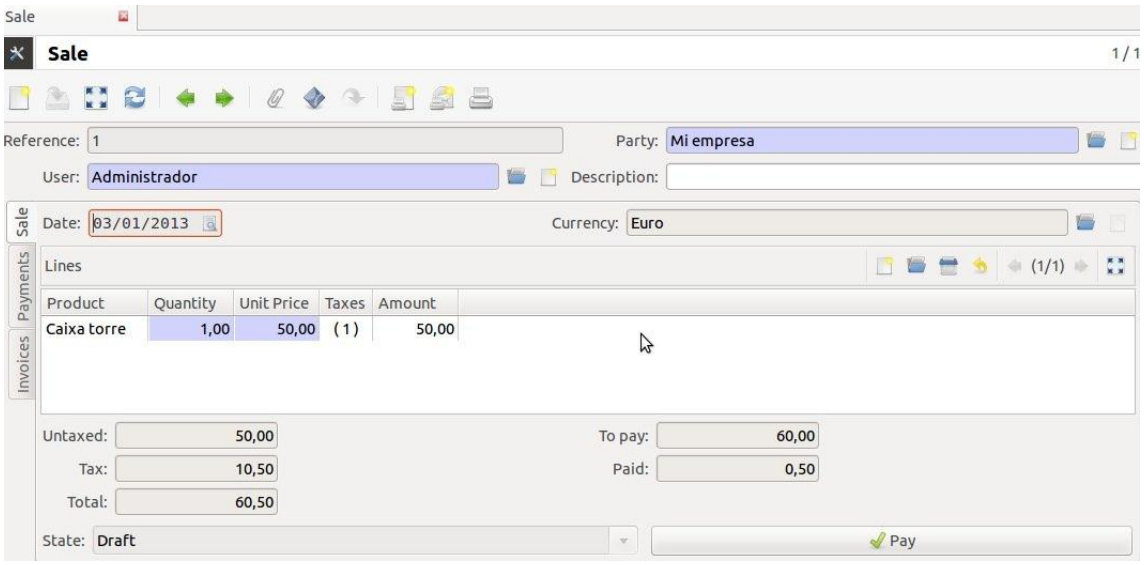


Figura 66: Formulari de venda

Aquesta figura correspon a la pestanya de pagaments/abonaments del formulari de venda minorista. Es pot observar com mostra una llista de tots els pagaments que s'han realitzar sobre la venda.

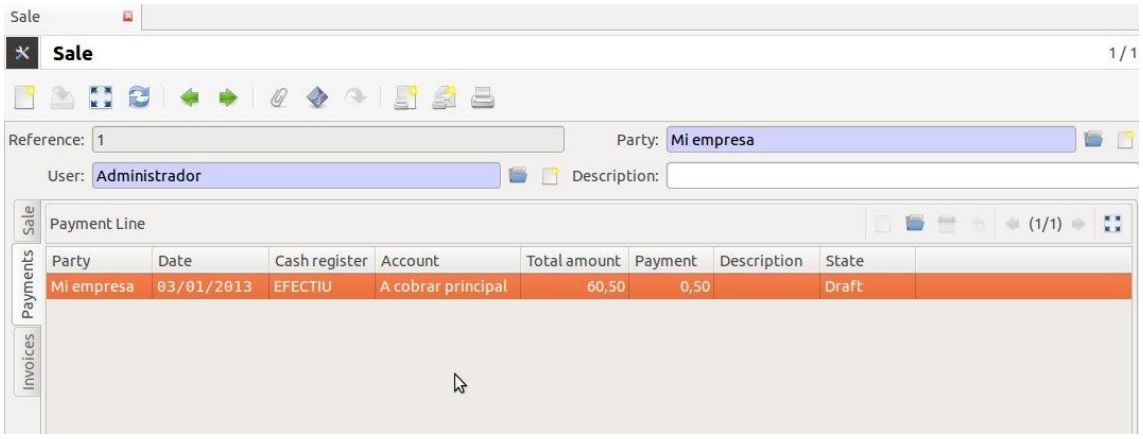


Figura 67: Pestanya de pagaments del formulari de venda

A continuació es mostra la imatge corresponent a la pestanya de factures del formulari de vendes minoristes. Mostra una llista de totes les factures generades per la venda minorista.

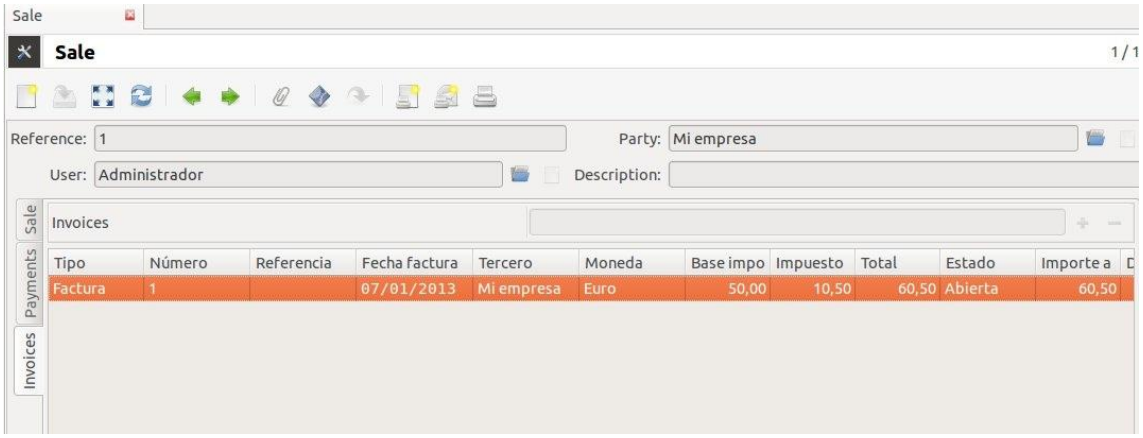


Figura 68: Pestanya de factures del formulari venda

- Pagament/abonament:

Els pagaments/abonament es realitzen mitjançant un *Wizard*. La següent imatge mostra el disseny de l'assistent (Wizard) que permet pagar una venda minorista. Mostra un camp de tipus selecció que permet escollir entre registres de caixa. Hi ha dos camps més que són numèrics. Un camp permet introduir el pagament que realitza el client mentre que l'altre camp mostra el saldo restant per abonar la venda.

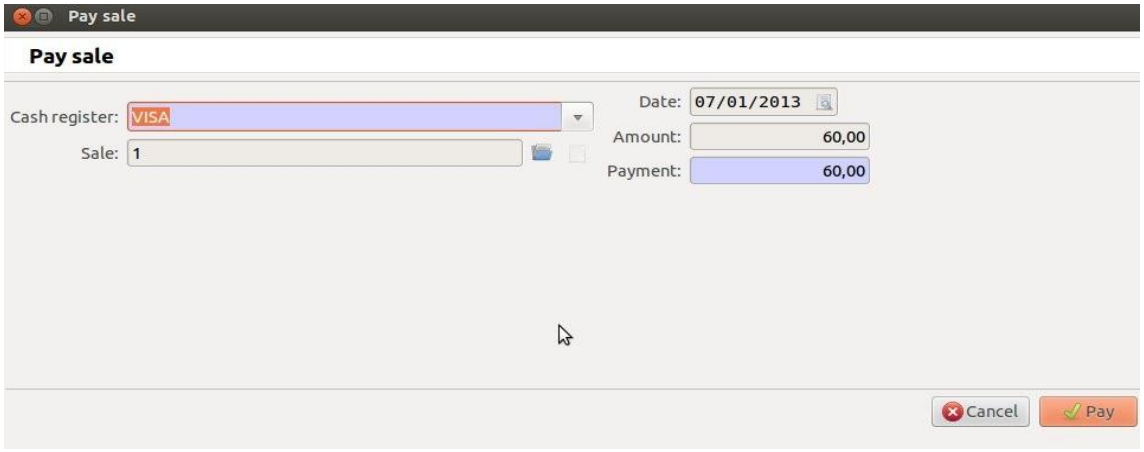


Figura 69: Assistent per pagar una venda

- Introduir operacions:

La introducció d'operacions també es realitza mitjançant un assistent. En aquest cas l'assistent esta composat per un parell de camps de tipus selecció. Un d'aquests camps permet escollir un dels registres de caixa que estiguin oberts mentre que l'altre camp permet seleccionar l'operació que volem. Hi ha un camp numèric per introduir l'import desitjat.

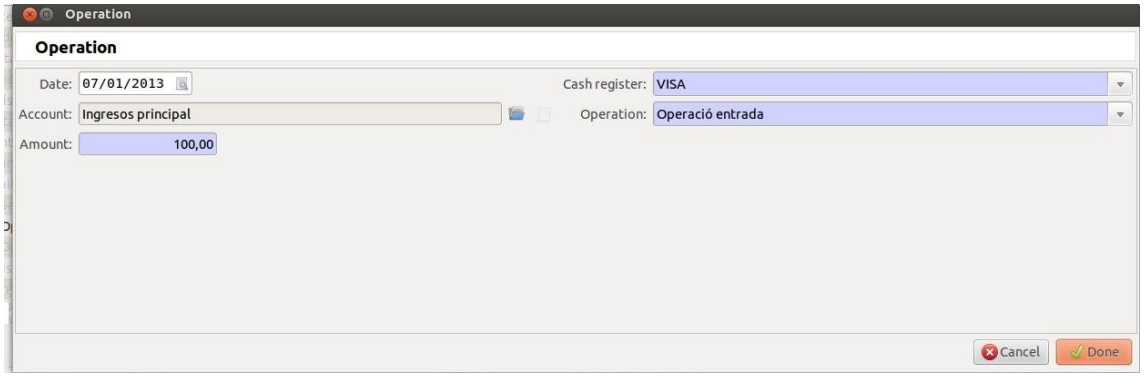


Figura 70: Assistent per introduir una operació

IV. Balanç i conclusions

6. Planificació final

La següent imatge mostra la planificació final del projecte. Les etapes estan definides més detalladament:

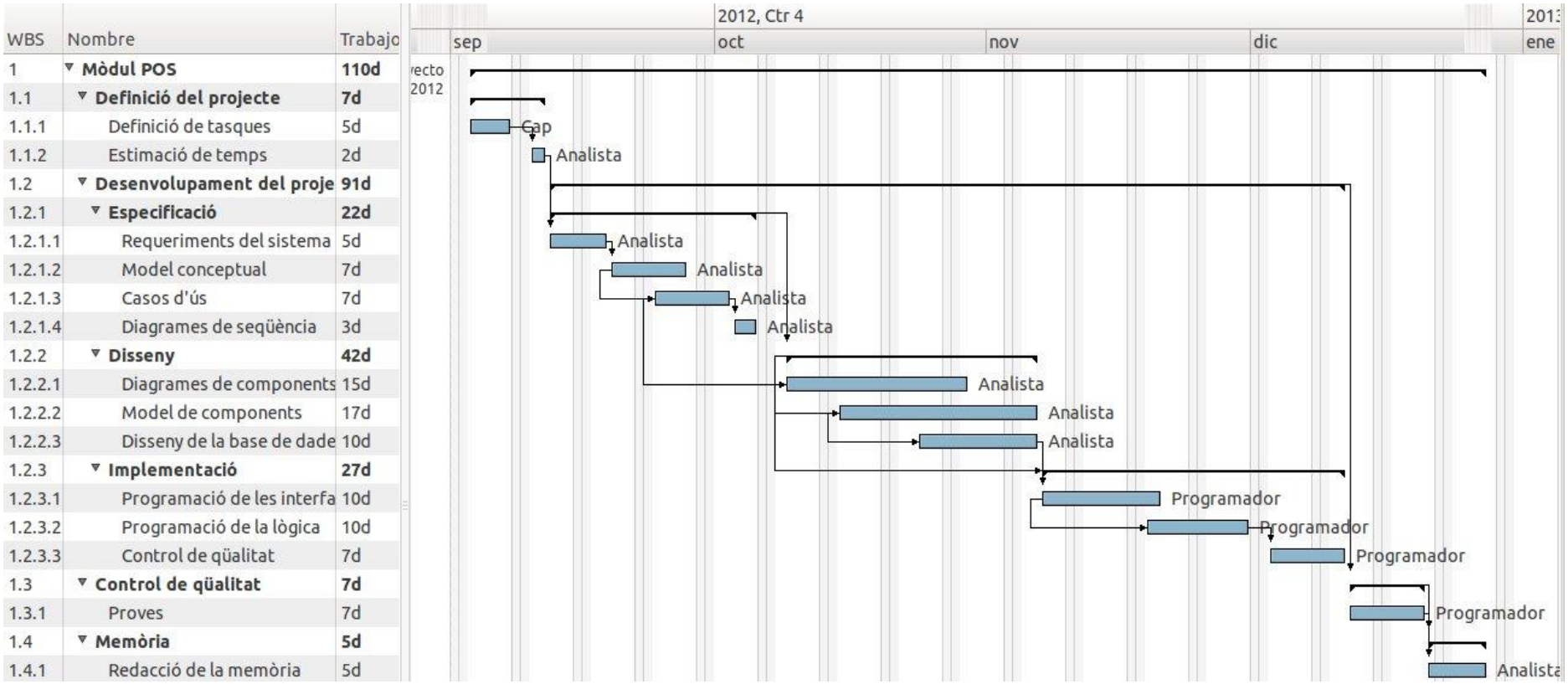


Figura 71: Planificació final

Com es pot observar en la planificació final hi ha hagut un retard de 8 dies respecta la planificació inicial.

Aquesta planificació s'ha fet segons els perfils professionals necessaris per realitzar un projecte d'aquest tipus, encara que el desenvolupament real del projecte s'ha realitzat amb un sol recurs (jo, que he actuat com a programador, analista, de cap i d'usuari). S'han inclòs els altres perfils i els seus costos per a que s'assembli a un projecte de software real i poder fer així un anàlisi de costos més acurat.

Abans de detallar els motius que han fet que la planificació inicial no es compleixi, s'analitzaran els costos del projecte segons la planificació final que és la que millor reflexa la realitat del projecte.

6.1. Anàlisis dels costos

La totalitat del software utilitzat per realitzar el desenvolupament d'aquest projecte és software lliure i OpenSource (FLOSS) (11), per tant no tenen cap cost associat. El cost del hardware utilitzat es basa en el cost de la connexió a Internet que ja disposo i el cost del PC que he utilitzat per realitzar el desenvolupament del projecte.

El cost de la connexió a Internet s'ha calculat a partir del cost de la quota mensual multiplicat per la durada del projecte. Per tant, el cost total de la connexió a Internet es calcula 30 €/mes * 5 mesos durada del projecte

Aquesta taula reflexa el cost total del software i hardware utilitzat.

Tipus	Nom	Cost
Software	Python	0 €
	PostgreSQL	0 €
	XML	0 €
	Tryton	0 €
Hardware	PC	600 €
	Internet	150 €
TOTAL		**750 €**

Per calcular el cost dels recursos humans utilitzats s'han tingut en compte les següents estimacions de cost per hora per cada perfil.

Perfil	Cost
Programador	25 €/h
Analista	55 €/h
Cap de projecte	70 €/h
Usuari final	0 €/h

Cost per perfil

En la següent taula es mostren les tasques realitzades per cada perfil, el cost que ha generat aquesta dedicació i l'interval de temps en que s'han realitzat les tasques.

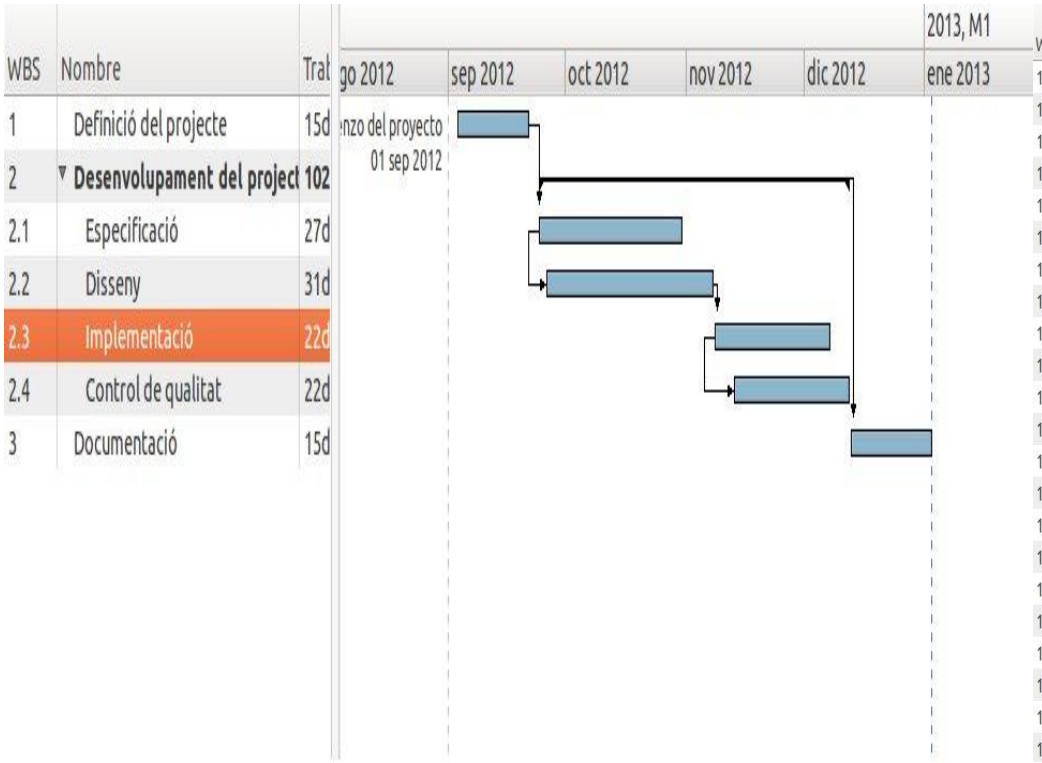
WBS	Nombre	Inicio	Fin	Trabajo	Duración	Coste	Asignado	% Completado
1	**Mòdul POS**	**sep 3**	**ene 1**	**110d**	**87d**	**41.560**		100
1.1	**Definició del projecte**	**sep 3**	**sep 11**	**7d**	**7d**	**3.680**		100
1.1.1	Definició de tasques	sep 3	sep 7	5d	5d	2.800	Cap	100
1.1.2	Estimació de temps	sep 10	sep 11	2d	2d	880	Analista	100
1.2	**Desenvolupament del projecte**	**sep 12**	**dic 11**	**88d**	**65d**	**32.960**		100
1.2.1	**Especificació**	**sep 12**	**oct 5**	**22d**	**18d**	**9.680**		100
1.2.1.1	Requeriments del sistema	sep 12	sep 18	5d	5d	2.200	Analista	100
1.2.1.2	Model conceptual	sep 19	sep 27	7d	7d	3.080	Analista	100
1.2.1.3	Casos d'ús	sep 24	oct 2	7d	7d	3.080	Analista	100
1.2.1.4	Diagrames de seqüència	oct 3	oct 5	3d	3d	1.320	Analista	100
1.2.2	**Disseny**	**oct 9**	**nov 6**	**42d**	**21d**	**18.480**		100
1.2.2.1	Diagrames de components	oct 9	oct 29	15d	15d	6.600	Analista	100
1.2.2.2	Model de components	oct 15	nov 6	17d	17d	7.480	Analista	100
1.2.2.3	Disseny de la base de dades	oct 24	nov 6	10d	10d	4.400	Analista	100
1.2.3	**Implementació**	**nov 7**	**dic 11**	**24d**	**25d**	**4.800**		100
1.2.3.1	Programació de les interfaces	nov 7	nov 15	7d	7d	1.400	Programa	100
1.2.3.2	Programació de la lògica	nov 19	nov 30	10d	10d	2.000	Programa	100
1.2.3.3	Control de qüalitat	dic 3	dic 11	7d	7d	1.400	Programa	100
1.3	**Control de qüalitat**	**dic 12**	**dic 20**	**7d**	**7d**	**1.400**		100
1.3.1	Proves	dic 12	dic 20	7d	7d	1.400	Programa	100
1.4	**Memòria**	**dic 21**	**ene 1**	**8d**	**8d**	**3.520**		100
1.4.1	Redacció de la memòria	dic 21	ene 1	8d	8d	3.520	Analista	100

Figura 72: Resum tasques per perfil

6.2. Comparació entre la planificació inicial i la planificació final

A continuació es mostra una comparativa entre la planificació inicial i la planificació final del projecte. Es pot observar com existeix un retard de la planificació final respecte a la planificació inicial:

Planificació inicial



Planificació final

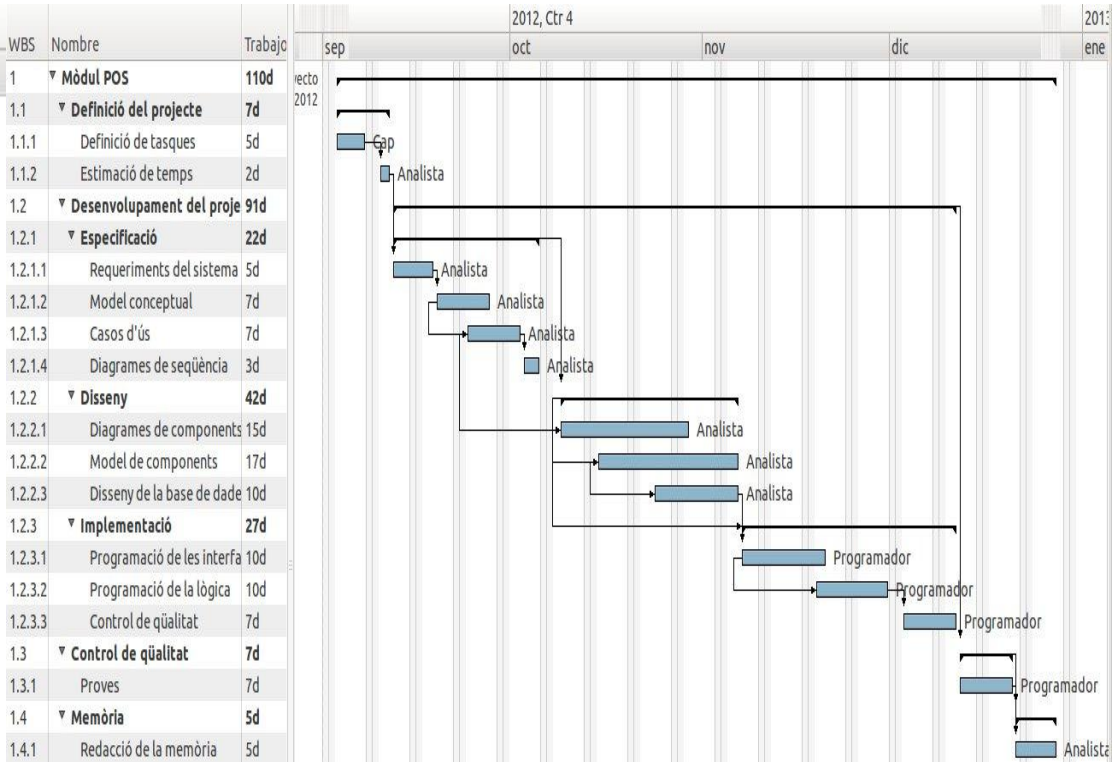


Figura 73: Comparació entre la planificació inicial i la planificació final

La primera planificació correspon a la planificació inicial del projecte. Té poques tasques perquè es van definir amb exactitud en la primera etapa.

L'etapa sobre el desenvolupament del projecte de la planificació final s'ha retardat respecte a la planificació inicial. Això és degut a que he tingut que modificar força els casos d'ús. Qualsevol canvi en els casos d'ús afecta a tot el disseny.

La implementació també s'ha retardat força respecte a la planificació inicial. És degut a que he tingut que aprendre a dissenyar les interfícies utilitzant XML i el framework de desenvolupament de Tryton. D'igual manera passa amb la programació de la lògica; he hagut d'aprendre les llibreries del framework de Tryton i a programar utilitzant-les.

La meva falta d'experiència en programació en Tryton i XML ha fet que gairebé totes les tasques de la implementació es retardin.

Un dels majors canvis de la planificació final a la planificació inicial, és la redacció de noves tasques per aconseguir més detall. Això es, com ja he comentat anteriorment, perquè les tasques les vam definir a l'inici del projecte. Mentre avançàvem en el projecte es van anant definit les mateixes tasques de la planificació inicial però amb més detall.

Altre canvi significatiu ha sigut separar la tasca de control de qualitat i la tasca de desenvolupament del projecte. És degut a que en principi s'anava a testejar l'aplicació mentre es desenvolupava però al final es va començar a testejar quan l'aplicació estava pràcticament finalitzada. Òbviament, mentre es desenvolupava l'aplicació es testejaven les funcionalitats bàsiques.

Per últim, crec que la planificació inicial, encara de ser molt senzilla reflexava amb bastant claredat les tasques principals de que es composa aquest projecte. El temps estimat en la planificació inicial no difereix gaire de la planificació final. Per ser un projecte curt i amb cap cost associat a recursos i que només s'ha demorat 8 dies, penso que la planificació inicial és bastant encertada.

7. Possibles millores

L'aplicació compleix tots els requeriments estipulats inicialment i afegeix alguna funcionalitat addicional que no s'havien contemplat, com la facturació agrupada de vàries vendes TPV.

Les següents millores, són possibles noves funcionalitats que podrien semblar atractives pels usuaris de l'aplicació TPV:

- Codi EAN 13: Afegir un camp pel codi EAN 13 als productes que permeti introduir aquest codi i validar-lo. D'aquesta manera es podrien llegir de forma òptima els productes a través d'aquest codi amb un lector de codi de barres.

- Generar informes directament en format PDF en comptes d'en format ODT (OpenOffice document text). Això permetria enviar el tiquet de la venda TPV directament a la impressora.

- Descomptes: Modificar el mòdul TPV perquè agafi els preus a partir d'una tarifa enlloc del preu de venda del producte. Això permet fer descomptes per producte o per categories del producte.

8. Conclusions

Personalment estic molt satisfet amb el resultat del projecte per varis motius.

Des del punt de vista personal, aquest projecte m'ha fet créixer molt com a enginyer informàtic, he hagut d'aprendre de forma autodidacta tecnologies que no havia vist mai i, no només aprendre com funciona, sino desenvolupar dos mòduls com els aquí descrits. No només pels motius anteriors he crescut com a enginyer, sino el fet de realitzar un projecte informàtic des de zero.

Personalment, el més difícil que m'he trobat ha sigut iniciar el a desenvolupament d'una aplicació desconeguda per a mi i aprendre a utilitzar el framework de desenvolupament de Tryton. Però un cop après, m'he adonat que aquesta eina és molt útil i facilita molt la feina.

He pogut participar en una comunitat culturalment molt diversa i que es dediquen a millorar o desenvolupar noves funcionalitats per a Tryton. Això m'ha fet veure com funcionen aquestes comunitats i quines possibilitats hi ha de continuar desenvolupant aquests mòduls així com aprendre a compartir informació amb altres persones.

Des del punt de vista formatiu, el projecte final de carrera ha complert les expectatives, ja que al finalitzar-lo puc demostrar que sóc capaç d'analitzar i desenvolupar noves funcionalitats per qualsevol aplicació passant per totes les fases d'un projecte.

D'altra banda dir que s'han complert tots els objectius establerts al començament del projecte, la qual cosa hem produeix una gran satisfacció.

9. Bibliografia

Per l'elaboració de la memòria he cercat informació en les següents fonts. Totes les fonts s'han consultat en el període setembre 2012 al gener 2013.

- Informació sobre Tryton

  - www.tryton.org, és la pàgina oficial de Tryton. Es pot trobar notícies sobre l'aplicació, documentació de totes es versions existents, els mòduls oficials que es distribueixen, especificació tècnica i informació tècnica amb exemples per desenvolupadors. Ha sigut la principal font d'informació per poder realitzar la implementar de l'aplicació.

  - http://tryton.zikzakmedia.com/, aquesta pàgina web és una wiki de l'empresa ZikZakmedia. Té articles que expliquen propietats del Framework de Tryton, exemples de programació, consells per realitzar la instal·lació de Tryton i explicacions de diverses funcionalitats.

- Programació en Tryton:

  - http://code.google.com/p/tryton/wiki/, és la adreça de les comunitats de Tryton que existeixen a Google. Es pot trobar continguts de tot tipus, des de informació de programació fins a informació plans comptables.

  - http://tryton.zikzakmedia.com/, com ja he comentat anteriorment, en aquesta adreça es pot trobar informació i exemples de programació sobre Tryton.

  - www.tryton.org, com ja he comentat anteriorment, en aquesta adreça es pot trobar informació sobre les últimes notícies de Tryton entre altres.

- Tecnologies utilitzades (Python, PostgreSQL, XML):

  - http://docs.python.org, pàgina web oficial sobre Python. Es pot trobar tota la informació relacionada amb aquest llenguatge. L'especificació, exemples de codi, explicacions d'objectes, mètodes de classes, ...

  - http://www.postgresql.org/, pàgina oficial on es pot trobar consells per aplicar a l'hora d'utilitzar aquest gestor de base de dades, accés a les comunitats, descarregar l'aplicació i enllaços per descarregar manuals d'utilització.

  - http://www.xml.com/, pàgina web des d'on es pot participar en comunitats. Es pot trobar informació sobre aquesta tecnologia.

Per aprendre a utilitzar l'aplicació Tryton he participat en les comunitats que hi ha en Internet i he consultat grans quantitats de fòrums, wikis i pàgines webs. Podria dir que els recursos més importants que he utilitzat han sigut:

- Wiki de l'empresa ZikZakmedia:
    - http://tryton.zikzakmedia.com/, com ja he comentat anteriorment aquesta pàgina web ha sigut una de les més consultades per tots els recursos útils que ofereix.
- Comunitat de Tryton:
    - https://groups.google.com/forum/?fromgroups#!forum/tryton-es, aquest enllaç pertany a la comunitat de Tryton d'Espanya. Es pot trobar tot tipus de continguts i es molt útil per resoldre dubtes específics sobre qualsevol tema de Tryton, sigui de programació o funcional.
    - https://groups.google.com/forum/?fromgroups#!forum/tryton-ca, semblant al cas anterior però amb la diferència que aquest enllaç pertany a la comunitat de Tryton de Catalunya.
    - https://groups.google.com/forum/?fromgroups#!forum/tryton-en, per últim, aquesta adreça web pertany a la comunitat de Tryton d'idioma anglès. El contingut és més extens que els enllaços anteriors i es pot trobar informació de tot tipus.

9.1. Bibliografia complementària

(1) (http://es.wikipedia.org/wiki/Sistema_de_planificaci%C3%B3n_de_recursos_empresariales)

(2) (http://es.wikipedia.org/wiki/Terminal_punto_de_venta)

(3) (www.tryton.org)

(4) (http://www.gnu.org/licenses/gpl-3.0.html)

(5) (http://es.wikipedia.org/wiki/XML-RPC)

(6) (http://es.wikipedia.org/wiki/JSON)

(7) (http://www.python.org/)

(8) (http://www.xml.com/)

(9) (http://www.postgresql.org.es/)

(10) (http://es.libreoffice.org/)

(11) (http://es.wikipedia.org/wiki/Software_libre_y_de_c%C3%B3digo_abierto)