Universitat Oberta de Catalunya

uoc.edu

Disseny i implementació de la base de dades d'un Sistema de generació d'apunts comptables

Dani Franquès
Grau d'Enginyeria Informàtica
Bases de dades

David Porti Pujal
Josep Curto Díaz

01/07/2024

FITXA DEL TREBALL FINAL

Títol del treball:	*Disseny i implementació de la base de dades d'un Sistema de generació d'apunts comptables*
Nom de l'autor:	*Daniel Franquès i Marsal*
Nom del consultor/a:	*David Porti Pujal*
Nom del PRA:	*Josep Curto Díaz*
Data de lliurament (mm/aaaa):	*07/2024*
Titulació o programa:	*Grau d'Enginyeria Informàtica*
Àrea del Treball Final:	*Bases de dades*
Idioma del treball:	*Català*
Paraules clau	*Sistema comptable, automatisme, escalable.*

Resum del Treball (màxim 250 paraules): *Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball*

La finalitat d'aquest treball és el disseny i la implementació d'un sistema de base de dades per ser usat en el desenvolupament d'una aplicació que permeti la generació dels apunts comptables d'una empresa. A partir d'una sèrie reduïda d'informació es puguin omplir tots els registres relacionats, com són els apunts comptables, els acumulats de comptes i els registres de factures.

S'ha optat per utilitzar la metodologia de desenvolupament en cascada(*waterfall*) pel tipus de projecte que és i també perque tenim unes dates d'entrega prefixades.

Per fer el projecte, s'han analitzat els requisits, els riscos i s'ha elaborat la planificació per a la seva execució.

Un cop teníem els requisits, per tal de tenir un bon control del projecte en l'inici s'ha planificat cada una de les fases en un diagrama de gantt i d'aquesta manera he pogut comprovar si s'anaven complint els terminis. La gestió dels riscos m'ha ajudat a tenir-los en compte i poder aplicar accions de prevenció per evitar la seva aparició.

El resultat obtingut ha sigut un projecte de base de dades que ens permetrà registrar de forma àgil els assentaments comptables.

Soc conscient de que hi ha molta feina a fer per tenir un producte acabat, tot i que ja és usable. També m'ha servit per posar en pràctica els coneixements adquirits i aprendre'n de nous.

Puc concloure que si bé els objectius del projecte s'han complert, aquest pot ser ampliat en un futur dotant-lo de més funcionalitats sense haver de canviar en gran manera els dissenys dels models.

Abstract (in English, 250 words or less):

The purpose of this work is the design and implementation of a database system to be used in the development of an application that allows the generation of accounting notes for a company. From small information, but necessary, you can fill in all the related records, such as the accounting notes, the accumulated accounts and the invoice records.

Due to the type of project the methodology chosen has been waterfall development, because that we have pre-set delivery dates.

The requirements and risks have been analysed and made the planning to do the project and get a great product.

To have a good control of the project, each of the phases was planned with a gantt diagram and in this way I was able to check whether the deadlines were being met. Also, risk management has helped me to take under control into account and to be able to apply preventive actions to avoid their occurrence.

The result obtained has been a database project that will allow us to register the accounting entries in an agile way.

I am aware that there is a lot of work to be done to have a finished product, but it's available to use. It has also helped me to put into practice the knowledge acquired and learn new ones.

In conclusion, all the objectives of the project have been met, and it can be expanded in the future by giving it more functionality without great changes in the designs of the model.

Índex

Llista de figures

1. Introducció

1.1 Context i justificació del Treball

En qualsevol organització empresarial hi ha diversos departaments, però n'existeix un que té una gran importància a l'hora d'analitzar resultats sobre l'evolució del negoci. Aquest és el departament d'administració, que engloba entre d'altres el de finances, tresoreria així com relacions amb l'administració, com més destacables.
Una de les feines més feixugues avui en dia del procés administratiu és la de l'entrada dels assentaments comptables[1]. Aquesta és una tasca en que es va registrant la informació comptable i permet saber l'estat financer de l'organització.
A l'igual que molts altres departaments de l'organització, agilitzar l'entrada d'informació en els sistemes de l'empresa permetrà ser més efectius i evitar errades en els processos, que a més a més són molts cops repetitius.

Si bé, existeixen diferents sistemes d'informació on el mòdul comptable té més o menys funcionalitats similars, aquests però, són paquets complets o que formen part d'altres solucions de tipus ERP, com per exemple SAP Business One[2], Sage[3], Microsoft Dynamics[4], ODOO[5]...

El que es pretén amb aquest projecte és el de crear un sistema que pugui ser independent del sistema d'informació de l'empresa i a la vegada potenciar l'àrea comptable disminuint les tasques que no aporten valor com és l'entrada gairebé manual dels assentaments comptables mitjançant una eina d'introducció dels mateixos totalment automatitzada on només sigui necessari entrar unes mínimes dades.

Un assentament comptable es composa com a mínim de: data, un numero d'assentament i dos comptes comptables que conformen els apunts. Tot assentament comptable ha d'estar quadrat, això vol dir que els moviments registrats com apunts al dèbit han de sumar el mateix que els que s'han registrat a l'haver. Aquests assentaments es poden agrupar per diferents tipus: Vendes, compres, nòmines, impostos, cobraments, pagaments...

A la vegada al ser una base de dades independent del sistema ERP de l'empresa es podria integrar en aquest mitjançant els protocols que el sistema d'informació de l'empresa proporcioni.

A la vegada que es genera la informació del diari comptable, com ara els assentaments, es vagin actualitzant les dades referents a acumulats i

[1] https://escholarium.educarex.es/Cursos/c156529_c5312851__Que_es_la_contabilidad.php
[2] https://www.sap.com/spain/products/erp/business-one.html
[3] https://www.sage.com/es-es/erp
[4] https://www.microsoft.com/es-es/dynamics-365
[5] https://www.odoo.com/es_ES

registres extra-comptables com poden ser els registres referents al registre d'IVA, o de factures emeses i rebudes.

Aquest projecte el que pretén és dotar d'un sistema d'automatismes definits perquè una aplicació d'escriptori, mòbil o web pugui utilitzar-los i retornar la informació per crear els assentaments.

Per tant, hem de tenir en compte aquest aspecte, ja que el que es vol és construir la base de dades i els procediments necessaris perquè un altre sistema se'n pugui beneficiar.

Per tal de resoldre la problemàtica comentada en el punt anterior, el treball a realitzar seria el disseny i implementació de la base de dades.

D'una banda tindríem la part de la configuració que estaria formada per unes taules que servirien per automatitzar la creació dels registres amb la informació que s'hauria de notificar al procediment d'entrada que generarà l'assentament comptable i altres apunts relacionats. En aquestes taules és on definiríem el comportament de l'entrada que permetrà dotar d'agilitat a la tasca d'incorporar informació comptable i de control de tresoreria de l'empresa.

D'una altra banda, tindríem les taules on s'emmagatzemarien les dades pròpies de la gestió, és on s'anirien creant els registres corresponents als apunts comptables, registres d'IVA, acumulats de comptes...

Tal com s'ha comentat la tasca principal del projecte consisteix en el disseny i implementació de la base de dades, l'aplicació de gestió podria ser desenvolupada en una altra fase. El desenvolupament d'una aplicació per implementar aquesta funcionalitat no és el motiu d'aquest treball, ja que ens cenyirem al procés del tractament de les dades. El mateix passaria amb l'aplicació d'administració, que seria una aplicació per fer la definició de l'estructura per crear els assentaments comptables.

Hem de tenir en compte que l'objectiu principal del projecte és el de convertir feines repetitives i amb probabilitat d'error en feines àgils. També s'ha de tenir en compte que la utilització del sistema que es pretén dissenyar és que l'usuari final no hagi de tenir grans coneixements sobre com s'han de realitzar els processos comptables. La seva 'feina' serà la d'escollir quin assentament predefinit utilitzar, donar-li la informació mínima requerida i el sistema ja ens generarà tots els registres associats, com ara el de l'assentament comptable, els acumulats de comptes, el registre d'IVA.
S'ha de tenir en compte que hi haurà 2 apartats diferenciats, per una banda el d'administració d'aquests assentaments predefinits i el de l'usuari que realitzarà l'entrada de la informació a partir de les definicions creades.

Al tenir un temps limitat, en un principi, quedarà fora de l'àmbit del projecte actual la creació de la definició dels assentaments, i per tant, partirem

d'uns preestablerts pel sistema. Dit d'una altra manera el que es dissenyarà serà la base de dades on es registraran aquests assentaments comptables, junt amb els procediments i les funcions que ho permetin fer, no la base de dades de definició dels assentaments, que tal com s'ha comentat anteriorment formaria part d'un altre projecte.

1.2 Objectius del Treball

Pel que fa als objectius que fan referència a la problemàtica exposada i que es volen assolir en el desenvolupament del projecte, són:

- Creació d'un sistema que permeti a l'usuari final de l'aplicació l'entrada d'assentaments de manera àgil i guiada.
- Creació dels registres necessaris per l'aplicació, que poden ser de tipus comptables, així com els registres d'IVA.

Així doncs, el que es vol, és obtenir una base de dades relacional, que sigui escalable en un futur.

A més d'aquests punts, que serien els requisits mínims, s'afegiran altre mecanismes que facilitin el manteniment del sistema, com poden ser bateries de tests per poder provar la funcionalitat del sistema, així com un sistema de log de les accions executades. Important per saber quins procediments s'han executat i el resultat d'aquestes execucions.

Els objectius d'aprenentatge que es volen assolir en el projecte, entre d'altres són:

- Posar en pràctica i consolidar els coneixements que s'han anat adquirint durant el Grau d'Enginyeria Informàtica, sobretot al respecte de l'estudiat en les assignatures de Ús de Base de Dades, Disseny de Bases de Dades, Enginyeria del programari...
- A partir d'una problemàtica, detectar les necessitats que ha de complir un sistema.
- Creació d'un disseny seguint la metodologia de Disseny de Bases de Dades. Això és: creació del disseny conceptual amb UML, disseny lògic i disseny físic segons el SGDB escollit.
- Realització del projecte segons la planificació establerta.
- Ampliar coneixements sobre l'ús de noves eines i bones pràctiques en el desenvolupament.

1.3 Enfocament i mètode seguit

Tenint en compte el tipus de projecte s'ha fet servir en un principi la metodologia *waterfall*[6] o en cascada, això vol dir que per passar a una

[6] Waterfall: <https://en.wikipedia.org/wiki/Waterfall_model> [Data de consulta: 6 d'abril de 2024].

fase, s'hauria d'haver finalitzat l'anterior i que aquesta sigui correcta. De totes maneres si cal fer un salt enrere seria possible, sobretot en el moment de la creació de tests.

S'ha escollit aquesta metodologia ja que al ser un projecte petit, en principi, no necessitarà de molta retroalimentació per part del client, tot i que presenta alguns inconvenients, d'aquí la documentació sobre els riscos, com per exemple que no s'ajusti exactament a com estava previst en els requeriments inicials. En aquest punt, podíem haver optat per utilitzar una metodologia *agile*[7], que ens hagués anat potser millor, ja que és més flexible i manté més col·laboració amb l'usuari/client a la vegada ens permet aprendre dels errors.

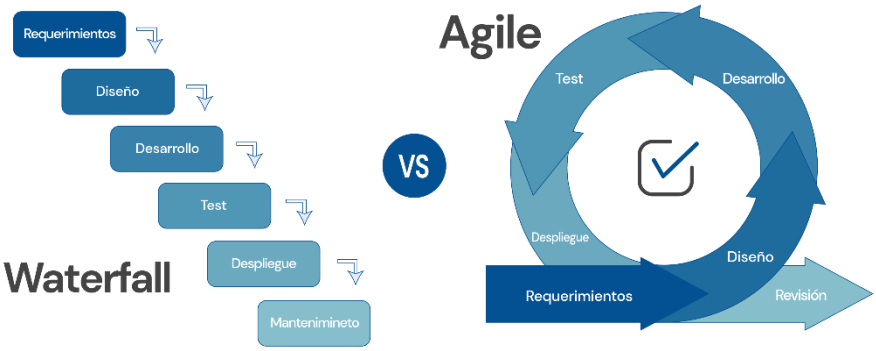


Figura 1: Comparativa Waterfall vs Agile.
Font:https://donetonic.com/es/metodologia-waterfall-vs-metodologia-agile/

1.4 Planificació del Treball

1.4.1 Recursos necessaris.

Els recursos necessaris per realitzar el projecte els podem dividir en recursos pel que fa al maquinari com pel que fa al programari, segons la relació que s'adjunta:

Maquinari

- Ordinador principal que serà el que farà de servidor de base de dades.
  - Disc dur. Aquest ordinador principal ha de tenir la suficient capacitat tant per que fa a l'emmagatzematge per poder tenir instal·lat a més del sistema operatiu, un sistema de màquines virtuals o amb contenidor de Docker[8] per poder fer proves en diferents entorns. L'ideal seria un disc dur d'1 TB, preferiblement de tipus SSD, per tenir major velocitat d'accés.
  - Memòria. Per la mateixa raó, hauria de disposar d'una memòria RAM mínim de 16 GB.

[7] Agile: <https://en.wikipedia.org/wiki/Agile_software_development> [Data de consulta: 6 d'abril de 2024].
[8] Docker: https://www.docker.com/

    - Processador. Al tractar-se d'un equip amb força càrrega de treball, el processador escollit hauria de ser a partir d'un Intel7 o similar.
- Pantalla de 24"-27", l'ideal seria disposar de 2 pantalles de 24" o bé una de 27" o superior, per tal d'incrementar l'àrea de treball i augmentar la visualització, això permetrà treballar i seguir els requeriments. També ens evitarà costos d'impressió, així com el fet de no haver d'estar contínuament canviant de pestanya o minimitzant/maximitzant finestres.
- Ordinador portàtil de *backup* com mesura correctora del risc R03, en aquest cas al tractar-se d'un equip d'emergència, els requeriments no caldria que fossin els mateixos que l'equip de treball, si bé, seria interessant que l'apartat de memòria i capacitat del disc dur s'acostin als valors mínims.

Programari

- SGBD – Sql Server Developer
- Eina de gestió de projectes (Microsoft Project)
- Eina de disseny de diagrames (Draw.io)
- Eina ofimàtica per generar documents i presentacions (MS Office)

Pel que fa a la valoració econòmica. Adjuntem quadre amb imports

Tal com s'especifica en el risc R03, a banda de l'ordinador principal, és interessant disposar d'un altre equip de substitució com a mesura per mitigar el risc.
En la valoració s'ha tingut en compte un cost de les hores del tècnic encarregat del desenvolupament (estimat en 30 € / hora), així com un cost orientatiu de material d'oficina i consumibles. Tot i que inicialment s'utilitza programari gratuït, es contempla el cost de les llicències del programari del sistema gestor de base de dades, en tot cas si es desitgés fer ús d'una versió al núvol, s'hauria de tenir en compte el cost d'aquest que pot diferir ja que la manera de valorar és diferent. El que aquí s'ha tingut en compte és el d'una instal·lació on-promise[9].

També es té en compte que tot i que el programari en la seva versió Developer és gratuït, no es podria utilitzar en un sistema de producció real, per tant s'ha de tenir en compte el cost de llicència d'una versió de SQL Server Standard[10].

Per tant, tenint en compte aquests criteris per fer la valoració econòmica del projecte s'ha inclòs per una banda el maquinari i per l'altra el programari necessari per realitzar el projecte.

[9] https://www.santanderconsumergs.com/articulo/https-www-ekon-es-blog-on-premise-vs-cloud-que-es-mejor
[10] https://www.microsoft.com/es-es/d/sql-server-2022-standard-edition/dg7gmgf0m80j

Pressupost estimat

Descripció	Unitats	Preu unitari	Import
HP Z1 G9 Intel Core i9-13900/32GB/1TB SSD/RTX 4060	1	2.408,96 €	2.408,96 €
Samsung Essential Monitor LS27C310EAUXEN 27" LED IPS FullHD 75Hz FreeSync	1	157,99 €	157,99 €
Acer Aspire 3 A315-59 Intel Core i5-1235U/16GB/512GB SSD/15.6"	1	579,00 €	579,00 €
Material ofimàtic	1	60,00 €	60,00 €
Hores tècnic	290	30,00 €	8.700,00 €
SQL Server Estàndard Edition	1	3.755,00 €	3.755,00 €
Microsoft Project	1	350,00 €	350,00 €
TOTAL			**16.010,95 €**

1.4.2 Tasques

A continuació es detallen les tasques realitzades, separades per fases:

- **Pla de treball.**
  - Revisió documentació assignatures anys anteriors.
  - Documentació del pla de treball.
    - Descripció de la problemàtica, objectius i tasques a realitzar
  - Anàlisi de Riscos.
  - Selecció de la metodologia de treball.

- **Definició del projecte.**
  - Disseny conceptual.
  - Transformació model de disseny conceptual al model lògic. Normalització.
  - Selecció i instal·lació del SGDB
- **Desenvolupament del projecte.**
  - Pas del disseny lògic al disseny físic.
    - Creació de taules
    - Creació de relacions
  - Desenvolupament de scripts[11] SQL
    - Creació de scripts per omplir les dades.
    - Creació de procediments d'inserció de dades.
- **Lliurament.**
  - Creació de consultes
  - Procediments de test
  - Generar documentació (memòria)
- **Defensa**

[11] https://es.wikipedia.org/wiki/Script

1.4.3 Planificació

En aquest apartat veurem la definició dels recursos necessaris per realitzar el projecte i una planificació temporal per cada àrea mitjançant un diagrama de Gantt.

La duració de les tasques s'ha definit en terme d'hores per tenir-ne una valoració més acurada i també per quantificar els recursos necessaris en personal.

S'ha tingut en compte que els dies laborables la dedicació en hores serà diferent a la dedicació que es faci en cap de setmana o festiu. Així partirem de forma inicial d'una dedicació de 2 hores diàries de dilluns a divendres, i de 4 hores els dies festius i cap de setmana. Aquesta dedicació d'hores equivaldria a unes 18 hores setmanals. Tanmateix es va quantificar una desviació d'entre un 10% i un 20 % per possibles imprevistos que poguessin sortir com ara buscar documentació per portar a terme el projecte, o bé fer proves, aprenentatge d'eines,... D'aquesta manera es va quedar ajustat a 15-16 hores setmanals.
Destacar, que en la fase del desenvolupament del projecte se n'han comptat més ja que havíem de tenir en compte que a banda del desenvolupament dels diferents scripts s'havia d'anar generant la documentació.

En la planificació mostrem les fites que serien les corresponents a les entregues i que tot i que en la planificació segons el diagrama no es mostren com tasques del projecte en sí, s'han de presentar les entregues pel que fa a les PAC i aquestes també les hauríem de considerar tasques a realitzar, així com la generació de la documentació relacionada amb la memòria.

La representació del diagrama de gantt està en l'annex: planificació.pdf.

1.4.4 Gestió de riscos.

Pel que fa a l'avaluació del nivell de risc, s'ha tingut en compte la relació entre la seva probabilitat i l'impacte en el projecte. El risc no es quantificable només per la probabilitat d'ocurrència, sinó també pel seu impacte sobre els objectius del projecte.

D'aquesta manera podem determinar un nivell, tal com es mostra en el quadre següent:

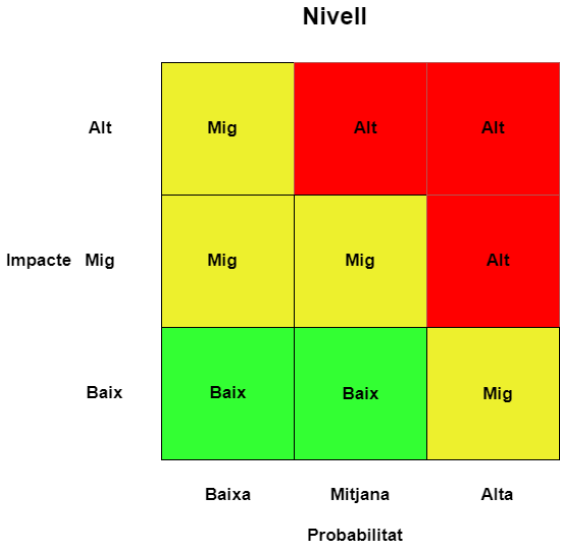


Figura 2. Matriu del producte de riscos

Hem de tenir en compte que tot projecte no està exempt de riscos i per tant cal estar preparat. L'objectiu d'aquesta gestió dels riscos és, d'una banda, augmentar la probabilitat i l'impacte dels esdeveniments positius (oportunitats), i de l'altra banda, disminuir la probabilitat i l'impacte dels esdeveniments negatius pel projecte (amenaces).

Pel que fa a aquest projecte la relació dels riscos que s'han identificat son els negatius. Que poden afectar directament a la consecució del projecte i que aquest no es finalitzi.

Tal com proposa el PMBOK, la gestió dels riscos inclou els processos següents:

- Planificació de la gestió dels riscos
- Identificació dels riscos
- Anàlisis dels riscos: tant qualitatiu com quantitatiu
- Planificació de la resposta als riscos.
- Implementació de la resposta.
- Control dels riscos.

Es detalla aquesta probabilitat i el seu impacte, també si hi ha cap tipus d'acció de prevenció, així com les mesures correctives que es proposin.

Figura 3. Gestió dels riscos del projecte.
Font:https://formawiki.diba.cat/pub/direccio_de_serveis/25ee7d1e_direccio_de_de_projectes/docs/8119_conti_origi_direc_curs_direc_unita_dp_m4_1.png

Els riscos, en funció de la fase on es produeixi, poden tenir més o menys impacte en el desenvolupament del projecte, d'aquí la importància de tenir-los identificats, amb les seves possibles causes, per tal de prendre mesures prèvies, així com mesures correctores.

En l'apartat següent es detallen els riscos rellevants que han sigut identificats en el projecte.

R01	**Incompliment de calendari**					
	Probabilitat	Alta	**Impacte**	Alt	**Nivell**	Alt
Descripció	Fer una mala estimació del temps o imprevistos					
Acció de prevenció	S'ha de fer la planificació tenint en compte aquesta possibilitat. Essent una mica conservador pensant en el pitjor escenari.					
Mesura correctiva	Fer una estimació incrementada en un percentatge. Incrementar temps dedicat al projecte.					

R02	**Destrucció de la base de dades de desenvolupament**					
	Probabilitat	Mitja	**Impacte**	Alt	**Nivell**	Alt
Descripció	Tot i ser poc probable, la possibilitat de malmetre la base de dades o d'esborrar-la es pot produir					
Acció de prevenció	Còpia de seguretat diària tant en dispositius externs com en ubicació al núvol. Utilitzar una metodologia de treball que permeti recrear l'estructura des d'un sol script.					
Mesura correctiva	--					

R03	**Fallada tecnològica**					
	Probabilitat	Baixa	**Impacte**	Alt	**Nivell**	Mig
Descripció	Problemes amb l'equip per a la realització del treball. Implica no poder continuar amb el projecte.					
Acció de prevenció	La mateixa que pel R02, disposar de còpies de seguretat és la millor prevenció. Així com tenir l'equip de treball correctament actualitzat i utilitzar només per al projecte.					
Mesura correctiva	Tenir un equip preparat per poder continuar en cas de fallada.					

R04	Normalització no adequada					
	Probabilitat	Baixa	**Impacte**	Mig	**Nivell**	Mig
Descripció	Error en el disseny de la base de dades tant en la fase del disseny conceptual com en les següents fases					
Acció de prevenció	Revisar cada una de les fases per tal de seguir la normalització.					
Mesura correctiva	Comprovar el disseny amb cura abans de passar a la fase següent.					

R05	Implementació de nous requeriments					
	Probabilitat	Mitja	**Impacte**	Mig	**Nivell**	Mig
Descripció	Detectats nous requeriments mentre avança el projecte					
Acció de prevenció	Revisió acurada del compliment dels requeriments.					
Mesura correctiva	Seria similar al R04, estudiar si els nous requeriments són adients per a la implementació de la fase en que es troben i si cal fer els passos necessaris per poder-los considerar.					

R05	Baixa per incapacitat de caràcter transitori					
	Probabilitat	Baixa	**Impacte**	Alt	**Nivell**	Mig
Descripció	Problemes de salut durant l'execució del projecte.					
Acció de prevenció						
Mesura correctiva	Si l'impacte és baix, serien les mateixes que en el R01. Si amb el temps destinat a imprevistos no n'hi ha prou, avisar al professor consultor.					

R06	Increment de càrrega de treball en l'àmbit laboral					
	Probabilitat	Baixa	**Impacte**	Alt	**Nivell**	Mig
Descripció	Possibilitat de puntes de feina en l'àmbit laboral que no permetin aconseguir la capacitat planificada de treball.					
Acció de prevenció						
Mesura correctiva	La mateixa que en el R05					

1.5 Breu sumari de productes obtinguts

Els productes que s'han obtingut en el desenvolupament d'aquest Treball Final de Grau són els següents:

- Base de dades: script de creació i inicialització de la base de dades.
- Memòria Final del Treball: Aquest document, on s'explica en que ha consistit el projecte, metodologia, fases, riscos, seguiment, etc.
- Presentació, Resum de la memòria.
- Annexos: Tots els documents addicionals, com ara són els scripts de creació de taules, procediments, així com jocs de proves. Aquests estan numerats i s'han d'executar per ordre.

1.6 Breu descripció dels altres capítols de la memòria

- Capítol 1: Introducció. Context i justificació del treball, els objectius del treball, l'enfocament i mètode seguit i la planificació d'aquest.

- Capítol 2: Disseny conceptual. Disseny d'entitats i relacions per crear el model UML. Detall de les decisions preses.
- Capítol 3: Disseny lògic. Traducció del disseny conceptual al disseny lògic.
- Capítol 4: Elecció SGBD. Selecció del programari per a dissenyar la base de dades.
- Capítol 5: Disseny físic. Traducció del disseny lògic amb el sistema escollit.
- Capítol 6: Seguiment del projecte. Explicació de les diferents etapes del projecte.
- Capítol 7: Conclusions. Resultat i anàlisi del que s'ha aprés i millores proposades.
- Capítol 8: Glossari. Definicions que s'han utilitzat en el desenvolupament del projecte.
- Capítol 9: Bibliografia. Referències a documentació utilitzada.
- Capítol 10: Annexos. Diversos documents addicionals.

1.7 Aspectes vinculats a ètica, sostenibilitat, responsabilitat social i/o drets humans i diversitat.

Referent als aspectes vinculats a ètica, sostenibilitat, drets humans i diversitat, pel meu projecte tindré en compte els següents principis:

- Compliment de la Llei Orgànica de Protecció de Dades(LOPD), amb la correcta utilització de les dades, de forma transparent i amb els consentiments necessaris.
- Protegir el sistema en diferents nivells d'accés depenent del tipus d'usuari amb contrasenyes i privilegis, així com sistemes d'auditories per registrar les accions que es realitzin.
- Dissenyar un sistema que no permeti l'ús amb fins no ètics, amb seguiment de les accions mitjançant registres de les operacions realitzades. Amb això el que es pretén és que el desenvolupament no permeti el seu ús per activitats no permeses en la llei.
- Optimitzar el consum de recursos, tant pel que fa al consum d'energia, com l'espai d'emmagatzemament dissenyant la base de dades de manera adequada buscant l'equilibri entre rendiment i que aquesta pugui ser sostenible. En referència a ser sostenible, es buscarà en la mesura del possible fer un disseny que optimitzi l'ús dels recursos, evitant tenir dades no necessàries, estalviant en espai d'emmagatzematge i que no sigui necessari calcular valors de manera continua. Fer-ho així minimitzarem l'impacte ambiental.
- Dissenyar un sistema que sigui usable per qualsevol persona, independentment del seu gènere, raça, discapacitat, orientació sexual, etc.

2. Disseny conceptual

2.1 Preàmbul

En referència a la nomenclatura del diagrama UML, s'ha utilitzat la grafia Pascal[12] per descriure les entitats i les relacions, i la grafia Camel[13] pels atributs. Les relacions d'integritat s'han elaborat mitjançant text.

2.2 Requeriments del sistema

La primera tasca a realitzar abans del disseny de la base de dades, ha sigut la recollida dels requeriments. La finalitat d'aquesta tasca és la d'obtenir tota la informació per extreure els requeriments que ha de complir el disseny de la base de dades. També s'hauran de tenir en compte alguns que puguin ser interessants en vistes a futures revisions.

Aquests requeriments els podem dividir en:

2.2.1 Requeriments Funcionals.

Són aquells que descriuen el comportament del sistema o les dades que han de persistir en el mateix.

2.2.2 Requeriments No Funcionals

Són aquells que descriuen les característiques esperades al sistema.

Codi	Descripció
RFN01	La base de dades ha de ser escalable i que pugui créixer segons les necessitats d'usuaris, així com del volum de dades.
RFN02	La gestió i accés a la informació es farà mitjançant procediments o consultes. Per tant s'han d'implementar els procediments d'alta, baixa i modificació de les entitats.
RFN03	Han d'incloure's mecanismes de control de la integritat de les dades.
RFN04	Els procediments han de poder tenir un control de transaccions.
RFN05	Els procediments han de disposar d'un paràmetre que indiqui si ha anat bé o no.
RFN06	Els procediments han d'estar correctament documentats.
RFN07	S'ha de poder disposar d'un registre d'esdeveniments per poder fer un seguiment i auditories.

[12] Pascal Case: https://www.theserverside.com/definition/Pascal-case
[13] Camel Case: https://www.techtarget.com/whatis/definition/CamelCase

2.3 Modelització i diagrama UML

En la figura es detalla el model UML de la base de dades.

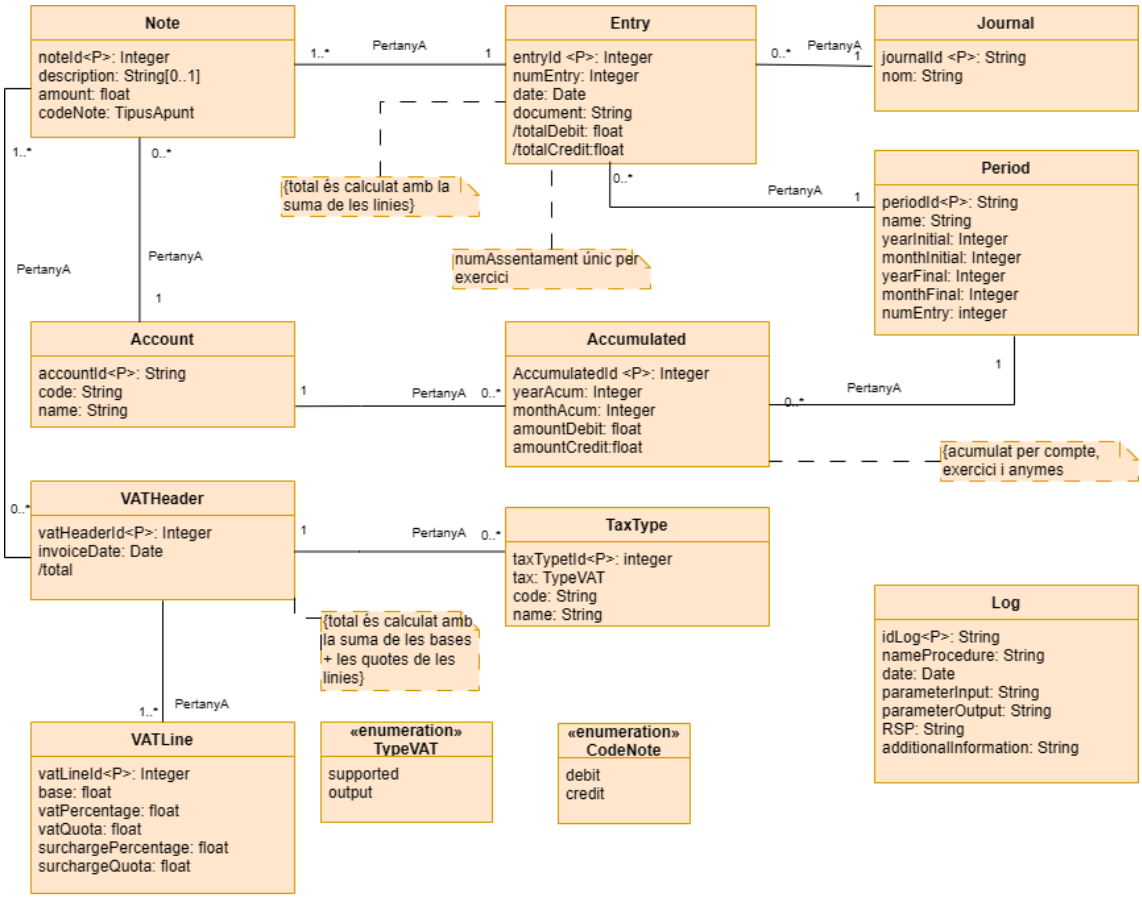


Figura 4. Model UML de la base de dades

Per a la realització del model, s'han definit les entitats i les seves relacions.

Els passos per la generació del diagrama han sigut:

1. Identificació de les entitats
2. Identificació dels atributs
3. Definició de les relacions entre entitats
4. Elaboració de les restriccions d'integritat que el model UML no permet representar.

2.3 Restriccions d'integritat

En l'entitat **Accumulated** els atributs **monthInitial** i **monthFinal** no poden ser inferiors a 1 ni superiors a 12.

En l'entitat **Period** els atributs **monthInitial** i **monthFinal** no poden ser inferiors a 1 ni superiors a 12.

En l'entitat **Period** si l'atribut **monthInitial** és superior a 1 l'atribut **yearFinal** ha de ser el **yearInitial** +1 i el **monthFinal** ha de ser el valor del **monthInitial** -1, en cas contrari, quan el **monthInitial** = 1, l'atribut **monthFinal** = 12 i els atributs **yearInitial** i **yearFinal** seran iguals.
Aquesta restricció ens servirà per obtenir el codi que identifica l'atribut del Period (**periodId**) a partir de la data entrada que sigui més gran o igual que l'any i mes inicial i més petit o igual que l'any i mes final. La restricció que també s'ha de tenir en compte és que si els atribut **yearInitial** i **yearFinal** són diferents tindrem que **monthFinal** = **monthInitial** - 1.

En l'entitat **Entry** tindrem un número d'assentament (atribut **numEntry**) únic per cada exercici, que serà obtingut a partir de la data i que aquesta data sigui més gran o igual que els atributs **yearInitial** i **monthInitial** i més petit o igual que els atributs **yearFinal** i **monthFinal**.

En l'entitat **VATLine**, l'atribut **percentatgeIVA** i **percentatgeRE** han de ser valors positius i inferiors al 100%.
En l'entitat **VATHeader**, l'atribut **invoiceDate** ha de ser inferior o igual que l'atribut data de l'entitat **Entry**.

2.4 Explicació de les entitats

Account:
Entitat que identifica un element del pla comptable al qual s'imputen càrrecs o abonaments. Per la naturalesa del pla comptable i poder obtenir informació agrupada, és defineix un codi de forma jeràrquica.

Atribut	Tipus	Descripció
accountId	Integer	Identificador de l'entitat. PK.
code	String	Codi identificador **únic**
name	String	Nom del pla comptable. No permet valors nuls.

Journal:
Entitat que permet classificar els assentaments comptable segons la seva naturalesa. Permetria tenir diferents classificacions per a una posterior consulta.

Atribut	Tipus	Descripció
journalId	Integer	Identificador de l'entitat. PK.
name	String	Nom del diari comptable. No permet valors nuls.

Period:
Defineix els diferents períodes comptables (de 12 mesos) que poden no tenir correspondència amb anys naturals.

Atribut	Tipus	Descripció
periodId	Integer	Identificador de l'entitat. PK.
name	String	Nom del període comptable. No permet valors nuls.
yearInitial	Integer	Any inicial del període. No permet valors nuls. Ha de ser superior a 0.

monthInitial	Integer	Mes inicial del període. No permet valors nuls. El seu rang ha d'estar entre 1 i 12.
yearFinal	Integer	Any final del període. No permet valors nuls. Ni pot ser inferior a l'atribut yearInitial
monthFinal	Integer	Mes final del període. No permet valors nuls. En el cas de que els atributs yearInitial i yearFinal siguin iguals, aquest atribut ha de ser més gran que montInitial, cas contrari ha de ser més petit (concretament l'anterior). El seu rang ha d'estar entre 1 i 12.
numEntry	Integer	Comptador d'assentaments del període. No permet valors nuls.

Entry:
Representa l'entrada d'un assentament comptable que engloba diferents apunts comptables en una data.
Aquesta entitat defineix un registre comptable en el període obtingut a partir de la data.

Atribut	Tipus	Descripció
entryId	Integer	Identificador de l'entitat. PK.
numEntry	Integer	número d'assentament únic per cada període.
date	Date	Data de l'assentament comptable, determinarà el període comptable.
document	String	Identificador del document registrat.

Note:
Representa l'entrada d'un apunt comptable.
Aquesta entitat defineix una anotació comptable identificat en si és un càrrec o un abonament (dèbit/haver).

Atribut	Tipus	Descripció
noteId	Integer	Identificador de l'entitat. PK.
import	Float	Import de l'anotació. No permet valors nuls.
codeNote	String	Tipus d'anotació. Només permet 2 valors: • D - Dèbit • H - Haver.
description	String	Descripció de l'anotació realitzada. Permet valors nuls.

Accumulated:
Defineix els acumulats per cada un dels elements del pla comptable en un període-any-mes.

Atribut	Tipus	Descripció
accumulatedId	Integer	Identificador de l'entitat. PK.
numAny	Integer	Any natural de l'anotació. No permet valors nuls. Ha de ser superior a 0.

numMes	Integer	Mes natural de l'anotació. No permet valors nuls. El seu rang ha d'estar entre 1 i 12.
monthInitial	Integer	Mes inicial del període. No permet valors nuls. El seu rang ha d'estar entre 1 i 12.
totalDebit	Float	Acumulat dels imports de les anotacions fetes al dèbit
totalCredit	Float	Acumulat dels imports de les anotacions fetes a l'haver

VATHeader:
Capçalera de la informació sobre el registre d'IVA, corresponent a una factura de compra o de venda.

Atribut	Tipus	Descripció
vatHeaderId	Integer	Identificador de l'entitat. PK.
invoiceDate	Date	Data de la factura
total	Float	Suma dels imports corresponents a les bases i les quotes de les línies.

VATLine:
Linies de les diferents bases, percentatges i quotes d'IVA o del recàrrec d'equivalència. Els atributs: **vatQuota** i **surchargeQuota** estan definits, ja que de vegades per diferències d'arrodoniment del cèntim a l'hora de quadrar una factura, el producte de la base pel percentatge no dona el mateix valor que la quota. Cas habitual de les factures de venda de productes al detall amb l'IVA inclòs.

Atribut	Tipus	Descripció
vatLineId	Integer	Identificador de l'entitat. PK.
base	Float	Base imposable de la factura. No permet nuls.
vatPercentage	Float	Percentatge d'IVA. No permet nuls.
vatQuota	Float	Quota de l'IVA. No permet nuls.
surchargePercentage	Float	Percentatge del Recàrrec d'equivalència. No permet nuls.
surchargeQuota	Float	Quota del Recàrrec d'equivalència. No permet nuls.

3. Disseny lògic

3.1 Preàmbul

Utilitzarem la notació proposada en l'assignatura de Disseny de Bases[14] de dades:

- Claus primàries subratllades amb línia contínua i les alternatives amb línia discontinua.
- Les claus foranies les descriurem amb un text.
- En negreta, els camps obligatoris (no nuls).

En general, per cada entitat que s'ha creat en el disseny conceptual té la seva correspondència en el model relacional, esmentar que les entitats associatives s'han representat amb claus foranes.

També s'ha buscat per tal de millorar l'eficiència no tenir molts atributs que puguin tenir valors nuls.

3.2 Disseny lògic

Account (accountId, code, **name**)

Journal (journalId, **name**)

Period (periodId, **name, yearInitial, monthInitial, yearFinal, monthFinal**)

Entry (entryId, periodId, numEntry, journalId, **data, totalDebit, totalHaver, document**)
{periodId} és foreign key de Period
{JournalId} és foreign key de Journal

Note (idNote, entryId, accountId, description, **import, tipusNote**)
{ entryId } és foreign key de Entry
{ accountId } és foreign key de Account

VATHeader (vatHeaderId, noteId, **tipusIVA**, **tipusRegistreIVA**, **total**, **dataFactura**)
{ noteId } és foreign key de LiniaNote

VATLine (vatLineId, vatHeaderId, **base, vatPercentage, vatQuota, surchargePercentage, surchargeQuota**)
{ vatHeaderId } és foreign key de VATHeader

Accumulated (accumulatedId, periodId, accountId, **monthAcum**, **yearAcum**, **amountDebit**, **amountCredit**)
{ accountId } és foreign key d'Account
{periodId} és foreign key de Period

[14] Burgués Illa, Xavier; Cuartero Olivera, Josep (2020). "Disseny lògic de bases de dades" Editorial UOC.

TaxType(taxTypeId, tax, code, name)

Log (LogId, **nameProcedure**, **date**, **RSP**, **parameterInput**, **parameterOutput**, **additionalInformation**)

3.3 Normalització

El disseny lògic s'ha validat mitjançant la teoria de la normalització, que defineix una sèrie de nivells, que s'anomenen formes normals i són:

1FN.
Es compleix ja que tots els atributs proposats són atòmics, és a dir que no hi ha cap atribut que pugui ser multivalor.

2FN.
Ha de complir que sigui 1FN i a més tot atribut que no forma part d'una clau candidata depèn completament de totes les claus candidates.

3FN.
Comprovem que compleix la tercera forma normal, ja que està en 2FN i a més cada atribut que no sigui clau candidata depèn només de la clau primària i no d'altres atributs.

FNBC (Boyce-Codd).
Es compleix, ja que està en 3FN i els determinants de totes les dependències en són claus candidates.

4FN.
Es compleix quan està en FNBC i no té dependències amb multivalors.

5FN.
També es compleix, ja que en les taules de relacions no tenim atributs que depenguin de combinacions de claus externes.

4. Elecció del SGBD

4.1 Criteris

En l'elecció del SGBD s'ha optat per fer una comparativa amb els sistemes treballats en les assignatures del grau: Ús de Base de Dades i Disseny de Bases de Dades, així com en altres opcions populars i d'aquesta manera prendre la decisió més encertada.

A l'hora de fer la selecció del SGBD, s'han tingut en els següents aspectes:

- Relacional
- Llenguatge SQL
- Escalable, Transaccional
- Multiusuari
- Permeti utilitzar JSON i/o XML com paràmetres d'entrada de les *stored procedures*[15].

En base als coneixements adquirits en les diferents assignatures i tenint en compte criteris econòmics i tècnics s'ha reduït la tria a quatre productes: PostgreSQL, MySql, Oracle Database i SQL Server.

L'elecció d'aquests productes ha sigut per una banda del coneixement com s'ha esmentat abans, d'una altra banda per ser productes amb versions lliures amb suport tant del fabricant com de la comunitat i en previsió d'una futura escalabilitat.

Les dues primeres opcions són OpenSource, això ens permet disposar d'un programari de codi obert amb ampli suport de la comunitat i les altres dues són de codi propietari amb un suport per part del fabricant.

També es va observar que totes quatre opcions podien ser desplegades mitjançant contenidors Docker i en sistemes tant Windows com Linux.

En conclusió, qualsevol dels quatre productes podria ser vàlid pel projecte i per tant la decisió s'ha basat més en criteris de preferència de les eines d'administració i de desenvolupament que aportava qualsevol de les solucions. Finalment l'opció que s'ha escollit és la de SQL Server en la seva versió *Developer*, amb mires d'una possible migració a SQL Azure o serveis d'Amazon AWS[16]. Les altres opcions avaluades ja les havia fet servir en altres assignatures i vaig pensar que era una bona opció el seleccionar un SGBD en que no hi hagués treballat.

En el disseny de la base de dades s'ha tingut en compte que pugui ser utilitzat per diferents clients i es farà tot l'accés mitjançant procediments emmagatzemats que seran cridats la majoria de vegades amb un paràmetre que contindrà dades estructurades en format JSON.

[15] https://en.wikipedia.org/wiki/Stored_procedure
[16] https://aws.amazon.com/?nc2=h_lg

4.2 Instal·lació SQL Server Developer

Els passos per fer la instal·lació del SGBD que s'ha realitzat han sigut:

Descàrrega de la versió des de la pàgina de descàrregues de Microsoft: https://www.microsoft.com/en-us/sql-server/sql-server-downloads

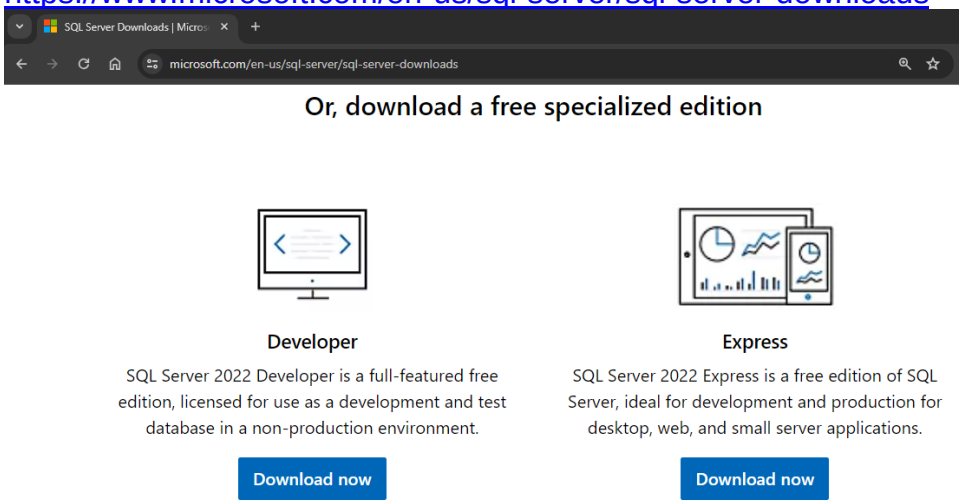


Figura 5. Descàrrega de SQL Server Developer

Un cop s'ha descarregat el programa instal·lador, s'han seguit els passos tal com es mostra en l'annex: Instal·lació SQL Server.pdf.

5. Disseny físic

5.1 Creació esquema

En el SGDB SQL Server, s'utilitza el llenguatge T-SQL (Transact- SQL). Podem definir esquemes, contenidors lògics que ens permetrà emmagatzemar segments de la base de dades i d'aquesta manera en un futur poder assignar permisos. Així, podem assignar l'accés a un tipus determinat d'informació depenent dels privilegis que tingui cadascun dels usuaris.

En el nostre projecte crearem una base de dades, de nom: TFG_UOC_AST.

```
CREATE DATABASE TFG_UOC_AST
```

Tot i que podem indicar més paràmetres com ara espai a ocupar, pàgina de codi, utilitzarem els valors per defecte.

Per facilitat en el projecte crearem totes les taules, funcions, vistes, *stored procedures* en l'esquema *Finance*. Per crear l'esquema ho farem amb la següent sentència:

```
CREATE SCHEMA Finance
```

En el nostre cas podem veure que tenim l'esquema creat mitjançant la sentència següent:

```
SELECT
     *
FROM sys.schemas;
```

Per tant la creació de les taules en l'esquema Finance tindrà aquesta forma:

```
Create table Finance.Account(id int, name varchar(20))
```

En aquest cas es crearia la taula Account, en l'exemple només s'especifica el camp id i el camp name.

5.2 Consideracions

A l'hora de crear les taules s'ha tingut en compte utilitzar la nomenclatura CamelCase, tant pel que fa al nom de les taules com pels camps.

S'ha tingut en consideració a l'hora de crear les taules de tenir camps numèrics identificadors dels registres i que serien les claus principals (PK).

Tot i que SQL Server permet un tipus de camp que s'anomena IDENTITY[17], per assignar camps únics, aquest és un camp que, en principi, no ens permet assignar valor ni editar-lo. També podem assignar valors, mitjançant SEQUENCE, ja que a l'hora de crear els procediments d'inserció de registres era més fàcil l'obtenció del numero assignat, tot i els possibles inconvenients que poden tenir, com ara que s'esborrés per error o bé s'assignés un valor fora de la seqüència.

La forma de crear una seqüència és la següent:

```
CREATE SEQUENCE Finance.AccountSeq
       AS INT
       START WITH 1
       INCREMENT BY 1 ;
```

En l'exemple creem una seqüència de nom AccountSeq (en l'esquema Finance) que comença en el numero 1 i s'incrementa d'1 en 1.

També en totes les taules s'ha afegit un camp que és la data de creació per tenir un millor control de quan es crea un registre.

A continuació es mostra les instruccions per generar una taula (i la seva seqüència), la resta s'informa en l'apartat d'annexos d'aquest document.

```
CREATE SEQUENCE Finance.AccountSeq
       AS INT
       START WITH 1
       INCREMENT BY 1 ;

GO

CREATE TABLE Finance.Account
(
       AccountId INT NOT NULL DEFAULT (NEXT VALUE FOR Finance.AccountSeq),
       Code varchar(30) NOT NULL,
       Name [NVARCHAR](250) NOT NULL,
       Level as LEN(Code),
       DateCreation DATETIME not null CONSTRAINT DF_Account_DateCreation
DEFAULT (GetDate())
       , CONSTRAINT PK_Account PRIMARY KEY (AccountId )
       , CONSTRAINT UK_Account UNIQUE (Code)
);
GO
```

També s'ha fet ús (com es veu en l'exemple, la columna *Level*) de columnes calculades, característica que fa que es pugui estalviar en espai d'emmagatzematge i el que no calgui calcular valors de manera continua. El mateix criteri s'ha utilitzat per crear unes columnes calculades en la taula Period, aquestes columnes calculades són **YearMonthInitial** i **YearMonthFinal**, que són concatenacions de les columnes **YearInitial** i **MonthInitial**, i de **YearFinal** i **MonthFinal** respectivament. Aquestes columnes calculades són utilitzades per trobar el registre de la taula Period que correspon a la data que li passem per paràmetre en la funció fGetPeriodId.

[17] https://en.wikipedia.org/wiki/Identity_column

```
YearMonthInitial as convert(varchar(4), YearInitial) +
format(MonthInitial, '00')

YearMonthFinal as convert(varchar(4), YearFinal) +
format(MonthFinal, '00'),
```

Pel que fa a les restriccions d'integritat sempre que ha sigut possible s'han fet en la mateixa definició de la taula, mitjançant la validació dels valors bé via claus foranes o bé amb *check constraints*[18] o en el cas de la taula Period aquestes validacions s'ha fet via un *trigger*[19], que permet posar valors per defecte i aplicar les restriccions. L'ús de *triggers* s'ha limitat en aquest cas només a fer validacions, tot i que es podria haver utilitzat a l'hora d'inserir registres en que depenent de quines taules executés accions, s'ha preferit fer-ho amb procediments, per tal de dotar de més facilitat de programació així de control de les accions realitzades.

5.3 Emplenat de taules

Inicialment les taules 'mestres' (Period, Journal, TaxType) s'han omplert amb sentències INSERT.

Pel que fa a la taula **Account**, per omplir aquesta, tot i que es pot utilitzar INSERTS, s'ha creat un *stored procedure* que automàticament també crea els registres que formen l'organització del pla comptable en forma jeràrquica segons el seu desglossament en dígits, com es pot veure en la següent figura.

El procés per omplir les taules de moviments: Entry, Note, VATHeader, VATLine i Accumulated es fa amb stored procedures.

43. CLIENTES
- 430. Clientes
  - 4300. Clientes (euros)
  - 4304. Clientes (moneda extranjera)
  - 4309. Clientes, facturas pendientes de formalizar
- 431. Clientes, efectos comerciales a cobrar
  - 4310. Efectos comerciales en cartera
  - 4311. Efectos comerciales descontados
  - 4312. Efectos comerciales en gestión de cobro
  - 4315. Efectos comerciales impagados
- 432. Clientes, operaciones de "factoring"
- 433. Clientes, entidades del grupo
  - 4330. Clientes entidades del grupo (euros)
  - 4331. Efectos comerciales a cobrar, entidades del grupo

Figura 6 - Recull del Pla comptable en mode jeràrquic.

Així pel cas del pla comptable si es crida a la *stored procedure* següent:

```
exec finance.addAccount NULL, '620012', 'Compte 620012'
```

[18] https://ca.wikipedia.org/wiki/Check_constraint
[19] https://sqlearning.com/es/elementos-avanzados/triggers/

Ens crearà els registres corresponents als dígits 5, 4, 3, 2 i 1, si no existeixen, amb una descripció que indica que cal omplir la columna del nom, a banda del registre que es crea.

Quedant així:

Code	Name	Level
620012	Compte 620012	6
62001	Pending complete - 62001	5
6200	Pending complete – 6200	4
620	Pending complete – 620	3
62	Pending complete – 62	2
6	Pending complete – 6	1

5.4 Funcions

El criteri per crear les funcions és el de que sigui objectes reutilitzables en altres apartats del projecte. També per diferenciar-les de les *stored procedures* s'ha afegit el prefix '*f*' i després d'aquest indicador de funció s'ha afegit el prefix '*get'* quan és una funció que obté la informació d'una taula de la base de dades.

La diferència d'utilitzar funcions o *stored procedures* és bàsicament per una qüestió de millor llegibilitat en el codi. En una funció és obligatori retornar un valor, cosa que no és necessari en un procediment, cosa que fa que per certes accions és més còmode es fa servir un procediment.

També, tot i que no s'han usat, una funció pot ser cridada des d'una sentència SQL (SELECT, UPDATE, INSERT, DELETE) i els procediments no.

Per exemple si volguéssim obtenir els apunts comptables d'un compte amb el codi '430025', ho podríem fer de la manera següent:

```
SELECT N.*
FROM finance.Note N
WHERE N.AccountId = Finance.fGetAccountId('430025')
```

Les funcions tenen una estructura més rígida i no es permeten tanta funcionalitat com els procediments, per aquesta raó, no en fem massa ús.

fGetAccountCode	
Descripció	Retorna el camp Code a partir de l'identificadors (AccountId)
Paràmetres entrada	AccountID
Retorn	Code

Operativa del procés	Fa la cerca en la taula Account.

fGetAccountId	
Descripció	Retorna l'identificadors (AccountId), a partir del paràmetre Code
Paràmetres entrada	Code
Retorn	AccountID
Operativa del procés	Fa la cerca en la taula Account pel paràmetre Code retornant l'identificador de la taula

fGetPeriodId	
Descripció	Retorna l'identificadors (PeriodId), a partir del paràmetre date
Paràmetres entrada	Code
Retorn	PeriodID
Operativa del procés	Fa la cerca en la taula Period per la data retornant l'identificador de la taula. Si no el troba retorna -1

fMakeYearMonth	
Descripció	Funció auxiliar que retorna l'any i el mes en format yyyymm a partir de la data. S'utilitza per trobar el Periode en la funció fGetPeriodId
Paràmetres entrada	date
Retorn	
Operativa del procés	Fa un formateig de la data passada retornant l'any i el mes.

fCheckAmountNote	
Descripció	Funció que comprova si la informació del JSON passat com paràmetre per crear un assentament comptable quadra.
Paràmetres entrada	json_data, valors JSON per generar l'assentament comptable.
Paràmetres sortida	Import de diferència, qualsevol valor diferent de zero és la diferència entre el dèbit i l'haver i per tant, no quadra i és un error.
Operativa del procés	Amb la instrucció OPENJSON, obté els valors dels registres corresponents als imports, diferenciant pel tipus.

fGetDateEntrybyNote	
Descripció	Funció que retorna la data de l'assentament a partir de l'identificador de l'apunt comptable.
Paràmetres entrada	NoteId, número de l'apunt comptable.

Paràmetres sortida	Data de l'assentament comptable.
Operativa del procés	Fa una cerca de la data de l'entrada de l'assentament, vinculant les taula Note i Entry pel camp EntryId de cada taula.

5.5 Procediments

La relació de procediments és la següent. Diferenciarem procediments els que tenen el sufix *JSON*, que són procediments on tenim un paràmetre de tipus NVARCHAR(MAX) que és una cadena JSON que ens permetrà tenir múltiples paràmetres segons una estructura, dels procediments de treball que accepten diversos paràmetres.

El criteri per nombrar els procediments ha sigut el de l'acció seguit del nom de la taula, així per posar un exemple el procediment addAccount, el que ens permetrà serà l'afegir (**add**) un registre a la taula Account.

S'ha simplificat el cas de modificació de registres que també, en gairebé totes les taules, s'utilitza el mateix procediment tant per afegir com per modificar registres. On si li passem com paràmetre l'identificador del registre el que es fa és modificar el registre que es cerca.

Pel que fa al procés de baixa de registres no s'han generat procediments com a tal de totes les taules, s'han creat de les taules Journal, Account,... els que s'han creat tenen el prefix '**delete**'.

Gairebé tots els procediments disposen d'un paràmetre de sortida anomenat RSP, que és de tipus *string*, que indica si l'execució ha finalitzat amb èxit ('OK') o si ha fallat, en aquest cas s'indicarà l'error i el tipus d'error.

També cada crida a un procediment farà un registre per tenir una traçabilitat de les crides que es fan a cadascun dels procediments, aquesta traçabilitat quedarà registrada en la taula Log, mitjançant el procediment ***addLog***, aquesta traçabilitat es fa al final del procediment, per tant, hem de tenir-ho en compte ja que si un procediment A crida a un B i aquest a un C, en el log primer es registrarà el C, després el B i finalment l'A.

5.5.1 Procediments. Altes/baixes/modificacions.

addLog	
Descripció	Registrar el procediment que s'executa
Paràmetres entrada	nameProc, paramIN, paramOut
Paràmetres sortida	

Operativa del procés	Inserta un registre en la taula Log.

addAccount	
Descripció	Donar d'alta/modificar un compte al pla comptable.
Paràmetres entrada	AccountId, Code, Name, MultiLevel
Paràmetres sortida	AccountId, per indicar l'id del registre creat. RSP
Operativa del procés	Si no li passem AccountId, comprova si n'existeix un pel paràmetre Code, cridant a la funció fGetAccountId. Donat que el pla comptable, tal com s'ha comentat anteriorment té format d'arbre depenent dels nivells que vulguem obtenir la informació a l'hora de crear un registre ens permetrà crear els registres relacionats (paràmetre Multilevel). Com que no seria correcte posar el mateix nom a tots els registres, s'opta per posar el valor 'Pending complete – ' i el codi com a nom. Així si creem el registre amb el codi 430025 – Client 430025, ens crearà, si no existeixen, el registre amb code='43002' i name='Pending complete – 43002', el 4300 amb code='4300' i name='Pending complete – 4300', ... Aquesta funcionalitat només té lloc quan es crea un registre nou, no en les modificacions.

deleteAccount	
Descripció	Esborrar un registre del pla comptable.
Paràmetres entrada	AccountId
Paràmetres sortida	RSP, per retornar si ha anat bé o no
Operativa del procés	Executa la instrucció i comprova si hi ha registres afectats. S'ha inclòs en una transacció per capturar si hi ha cap missatge d'error per alguna regla d'integritat (és clau forana de la taula Note i Accumulated)

deleteAccountbyCode	
Descripció	Esborrar un registre del pla comptable.
Paràmetres entrada	code
Paràmetres sortida	RSP, per retornar si ha anat bé o no
Operativa del procés	Obté l'accountId cridant a la funció fGetAccountId i si retorna un valor vàlid (superior a 0) crida al procediment deleteAccount

addJournal	
Descripció	Donar d'alta/modificar un registre del diari comptable.
Paràmetres entrada	JournalId, Name
Paràmetres sortida	JournalId, per indicar l'id del registre creat. RSP
Operativa del procés	Si no li passem JournalId, n'obté un a partir del sequence JournalSeq.

deleteJournal	
Descripció	Esborrar un registre del diari comptable.
Paràmetres entrada	JournalId
Paràmetres sortida	RSP, per retornar si ha anat bé o no
Operativa del procés	Executa la instrucció i comprova si hi ha registres afectats. S'ha inclòs en una transacció per capturar si hi ha cap missatge d'error per alguna regla d'integritat (és clau forana de la taula Entry)

addPeriod	
Descripció	Donar d'alta/modificar un registre dels períodes comptables.
Paràmetres entrada	PeriodId, Name
Paràmetres sortida	PeriodId, per indicar l'id del registre creat. RSP
Operativa del procés	Si no li passem JournalId, n'obté un a partir del sequence JournalSeq.

deletePeriod	
Descripció	Esborrar un registre dels períodes comptables.
Paràmetres entrada	PeriodId
Paràmetres sortida	RSP, per retornar si ha anat bé o no
Operativa del procés	Executa la instrucció i comprova si hi ha registres afectats. S'ha inclòs en una transacció per capturar si hi ha cap missatge d'error per alguna regla d'integritat (és clau forana de la taula Entry i Accumulated)

5.5.2 Procediments. Accions.

updateBalance	
Descripció	Actualitzar els acumulats per període
Paràmetres entrada	AccountId, PeriodId, Year, Month, AmountDebit, AmountCredit

Paràmetres sortida	RSP, per indicar si ha anat bé o si pel contrari hi hagut algun tipus d'error.
Operativa del procés	Obté l'identificador de l'acumulat a partir dels paràmetres passats del numero de compte, període comptable, mes i any acumulat i actualitza els valors. Si no troba l'identificador crea un nou registre

updateBalancebyCode	
Descripció	Actualitzar els acumulats per periode
Paràmetres entrada	Code, Date, Debit, Credit
Paràmetres sortida	RSP
Operativa del procés	A partir del paràmetre code, s'obté l'accountId i a partir del paràmetre date, s'obte el període, el mes i l'any per fer la crida al procediment: updateBalance

PeriodIncrementNumEntry	
Descripció	Obtenir un número d'assentament (*Numentry*) a partir del període
Paràmetres entrada	PeriodId
Paràmetres sortida	NumEntry, RSP
Operativa del procés	A partir del paràmetre PeriodId, obtindrà el següent número d'assentament pel període. En cas d'error o de no trobar el PeriodId retornarà en el paràmetre NumEntry el valor de -1
Notes	Aquest procediment es podia haver fet com una funció, però es va decidir per qüestió de bones pràctiques que les funcions no fessin actualitzacions de registres, que només retornessin valors.

addEntry	
Descripció	Crear un assentament comptable
Paràmetres entrada	JournalId, date, Document
Paràmetres sortida	EntryId, RSP
Operativa del procés	Permet fer l'entrada de l'assentament. A partir del paràmetre date obtenim el període. El paràmetre JournalId ha d'existir, i es valida amb una restricció de la taula que seria capturada pel control de transaccions.

addNote	
Descripció	Creart un apunt comptable

Paràmetres entrada	EntryId, AccountId, Amount, Description, CodeNote
Paràmetres sortida	NoteId, RSP
Operativa del procés	Permet fer l'entrada de l'apunt comptable. El paràmetre AccountId ha d'existir, i es valida amb una restricció de la taula que seria capturada pel control de transaccions. Aquest procediment també actualitza els imports de la taula Entry. En el cas de que no s'informi el paràmetre CodeNote amb els possibles valors D o H (dèbit o haver) s'obte aquest segons el signe del paràmetre Amount, si és negatiu, és **H**aver, altre cas és **D**èbit. Aquest procediment també fa la crida al procediment updateBalancebyCode, per actualitzar els acumulats del compte comptable i del seu arbre.

addVAT	
Descripció	Crear una capçalera del registre d'impostos associat a un apunt comptable
Paràmetres entrada	NoteId, Tax, TaxType, InvoiceDate
Paràmetres sortida	VatHeaderId, RSP
Operativa del procés	Permet fer la creació del registre de capçalera dels impostos associat a l'apunt comptable. El paràmetre TaxType ha d'existir, i es valida amb una restricció de la taula que seria capturada pel control de transaccions. En el cas de que no ens passin data de factura (paràmetre invoicedate), s'agafa la de l'assentament a partir del paràmetre NoteId, utilitzant la funció fGetDateEntrybyNote.

addVATLine	
Descripció	Crear el registre d'impostos associat a una capçalera d'impostos
Paràmetres entrada	VatHeaderId, Base, VATPercentage, VATQuote, SurchargePercentage, SurchargeQuote
Paràmetres sortida	VatLineId, RSP
Operativa del procés	Permet fer la creació d'un registre dels impostos associat a la capçalera. Farà tants registres com diferents percentatges de l'impost o del recàrrec. A més actualitza el camp total de la capçalera del registre d'impostos, amb la suma dels camps: base, quota i quota del recàrrec (base, VATQuote i VATSurchargeQuote, respectivament)

addAccountsJSON	
Descripció	Permet donar d'alta i modificar registres del pla comptable a partir d'una cadena JSON en format array

Paràmetres entrada	json_data
Paràmetres sortida	RSP, per indicar si ha anat bé el procés o no.
Operativa del procés	Obté els valors a partir de la cadena en format JSON que sigui un array l'element inicial, fent una crida al procediment addAcountJSON.

addAccountJSON	
Descripció	Permet donar d'alta i modificar registres del pla comptable a partir d'una cadena JSON
Paràmetres entrada	json_data
Paràmetres sortida	RSP, per indicar si ha anat bé el procés o no.
Operativa del procés	Obté els valors a partir de la cadena en format JSON i dona d'alta o modifica els registres del pla comptable cridant al procediment addAcount.

addVATLineJSON	
Descripció	Permet crear línies dels impostos a partir d'una cadena JSON
Paràmetres entrada	json_data, cadena JSON amb les dades, VATHeaderId identificador de la capçalera del registre d'impostos.
Paràmetres sortida	RSP, per indicar si ha anat bé el procés o no.
Operativa del procés	Obté els valors a partir de la cadena en format JSON per tal de crear un registre corresponent cridant al procediment addVatLine

addVATLinesJSON	
Descripció	Procediment que crida al procediment addVATLineJSON per cada element de la cadena JSON del tipus VATLine que correspon a percentatges i/o quotes diferents
Paràmetres entrada	json_data, VATHeaderId, identificador de la capçalera dels impostos
Paràmetres sortida	RSP, per indicar si ha anat bé el procés o no.
Operativa del procés	Obté un objecte amb la informació del percentatge correspoentn a l'impost, a partir de la cadena en format JSON, cridant al procediment addVATLineJSON

addNoteJSON	
Descripció	Procediment per afegir apunts comptables
Paràmetres entrada	json_data, EntryId, identificador de l'assentament comptable
Paràmetres sortida	NoteId, retorna el numero d'assentament creat. RSP, per indicar si ha anat bé el procés o no.

Operativa del procés	Obté els valors a partir de la cadena en format JSON per tal de crear l'assentament comptable, cridant al procediment addNote. Des d'aquest procediment, si està definit així en el paràmetre json_data de la cadena JSON , es crearan els registres corresponents a impostos. Prèviament el que fa aquest procediment és comprovar si existeix el compte comptable fent una crida a la funció fGetAccounId.

addNotesJSON	
Descripció	Procediment que crida al procediment addNoteJSON per cada element de la cadena JSON del tipus **note** que correspon a percentatges i/o quotes diferents
Paràmetres entrada	json_data, EntryId, identificador de l'assentament
Paràmetres sortida	RSP, per indicar si ha anat bé el procés o no.
Operativa del procés	Obté els valors a partir de la cadena en format JSON per tal de crear l'assentament comptable, cridant al procediment addNoteJSON, passant-li la cadena amb el format que necessita el procediment.

addEntryJSON	
Descripció	Permet donar d'alta assentaments a partir d'una cadena JSON
Paràmetres entrada	json_data
Paràmetres sortida	EntryId, retorna el numero d'assentament creat. RSP, per indicar si ha anat bé el procés o no.
Operativa del procés	Obté els valors a partir de la cadena en format JSON per tal de crear l'assentament comptable, cridant al procediment addEntry

addEntryGlobalJSON	
Descripció	Procediment principal per crear els assentaments i tots els registres relacionats a partir de la cadena JSON amb el format definit.
Paràmetres entrada	json_data
Paràmetres sortida	EntryId, retorna el numero d'assentament creat. RSP, per indicar si ha anat bé el procés o no.
Operativa del procés	Obté els valors a partir de la cadena en format JSON per tal de crear l'assentament comptable, cridant al procediment addEntryJSON, si aquest va bé, cridarà al procediment addNotesJSON que processarà cada element de tipus 'note' per crear els apunts comptables i si en disposen del registre d'impostos.

Referent als procediments s'ha de tenir en compte algunes consideracions pel que fa a bones pràctiques, com ara:

- No repetició de codi
- Reutilització de procediments

Pel que fa a la no repetició de codi, de forma normal un procediment fa només una tasca i si n'ha de fer d'altres crida a un altre perquè ho faci.

Tanmateix hi ha procediments en que es repeteix part del codi, de forma inevitable com és la declaració de variables o assignació d'aquestes per poder-se fer servir més endavant en el procediment com les variables d'assignació de valors per fer el log.

En conseqüència, hi ha procediments de nom similar. Un exemple del que es comenta, ho tenim en els procediments, com ara ***addNotesJSON*** i ***addNoteJSON***, el primer processa un array d'objectes JSON cridant al segon amb l'objecte JSON.

5.5.3 Procediments. Consulta

L'apartat de consultes al no haver uns requisits recollits, ja que no era en principi el motiu del desenvolupament del projecte, s'han fet uns quants per poder validar la informació registrat. Malgrat això, les consultes que s'han creat son totalment vàlides per la validació de la informació entrada.

qryTotalInvoicebyCode	
Descripció	Procediment que permet obtenir les factures rebudes per codi i any.
Paràmetres entrada	Opcionals: codi inicial i codi final.
Paràmetres sortida	
Operativa del procés	Executa una instrucció *select* filtrant pels codis passats per paràmetres, en cas de no passar-ne, posa el blanc i 'z' com inicial i final, respectivament.

qryAccumulatedbyCode	
Descripció	Procediment que permet obtenir els acumulats de comptes segons un nivell (dimensió del pla comptable).
Paràmetres entrada	Opcionals: codi inicial i codi final, nivell.
Paràmetres sortida	
Operativa del procés	Executa una instrucció *select* filtrant pels codis passats per paràmetres, en cas de no passar-ne, posa el blanc i

	‘z’ com inicial i final, respectivament. I si no és informat el nivell s’agafa el 6.

qryAccumulatedLevel	
Descripció	Procediment que permet obtenir els acumulats de comptes segons un nivell (dimensió del pla comptable).
Paràmetres entrada	Opcionals: nivell.
Paràmetres sortida	
Operativa del procés	Executa una instrucció *select* filtrant pels codis passats per paràmetres, en cas de no passar el nivell s’agafa el valor 6.

TEST_showAllTables	
Descripció	Procediment que ens permet obtenir un ‘select’ de totes les taules
Paràmetres entrada	
Paràmetres sortida	
Operativa del procés	Executa una instrucció *select* per cada una de les taules. El seu ús per fer proves i validar-les de forma visual.

5.6 Triggers

En aquest projecte no s’ha disseny un sol *trigger*. S’ha optat per fer el màxim d’operacions mitjançant *stored procedures* i minimitzar l’ús dels *triggers* per una qüestió de major control dels possibles errors.

Tot i que s’és coneixedor de la vàlua dels mateixos,ja que ajuden a automatitzar moltes accions, s’ha optat per no utilitzar-los si no era del tot necessari, com és el cas de la taula ***Period***, que amb les restriccions a nivell de taula no n’hi havia suficient.

Period_Valid	
Descripció	Trigger de la taula Period, per tal de validar que la informació que es guardi sigui coherent.
Operativa del procés	Aquest *trigger* el que fa és comprovar que els valors de les columnes de la taula Period, respectin les regles d’integritat del disseny conceptual. A més permet no haver d’informar de tots els valors. Per exemple, si no posem any final (camp YearFinal), l’actualitza amb el valor del camp YearInitial.

5.7 Pla de proves

El pla de proves creat per tal de poder avaluar la funcionalitat de la base de dades segons les especificacions que s’han especificat en els diferents

dissenys, es basa en la comprovació de la correcte execució, així com quan es comprova si ha de donar un error aquest quedi registrat.

Per omplir les taules 'mestres' (Period, Journal, TaxType) s'ha fet mitjançant scripts de forma manual que després s'ha revisat que no doni cap error.

Així tenim una sèrie d'scripts (inclosos en l'annex) que s'han d'executar en ordre, que entre altres coses el que fan és crear i comprovar:

- Creació de la base de dades i l'esquema.
- Generació de les taules. Revisem que no existeixen errors a la generació de les taules. Confirmem que son adequades tant les claus primàries com les foranes.
- Restriccions i validacions. Proves d'inserir registres, tot comprovant que compleixen les restriccions que hi ha a les taules amb restriccions, com claus duplicades o que faltin valors en camps que són obligatoris, valors no permesos, etc... Així comprovem que es produeixen els errors esperats, tal com s'especifica en les regles d'integritat.
- Confirmació que les restriccions de clau forana no permeten eliminar dades relacionades. I que si es creen registres sense correspondència amb la taula forana.
- Realització de consultes bàsiques, per tal d'obtenir dades i amb relacions complexes entre taules.

En l'script '03-Insertar registres.sql', es poden anar executant instruccions per validar que les dades es guarden correctament o bé que donen error. Hem de tenir en compte que estem utilitzant el software Microsoft SQL Server Management Studio, que ens permet executar sentències individuals o en bloc, seleccionant el text, tal com es pot veure en la captura de pantalla següent:

```
-- tipus registre d'iva
INSERT INTO Finance.TaxType(Tax, Code, Name )
VALUES
('S', 'G', 'General'),
('S', 'I', 'Importació'),
('S', 'C', 'Comunitari'),
('R', 'G', 'General')
GO
-- ha de donar un error si vull insertar un registre repetit:
-- Error:Infracción de la restricción UNIQUE KEY 'UK_TaxType'.
-- No se puede insertar una clave duplicada en el objeto 'Finance.TaxType'. El valor de la clave duplicada es (S, G).
INSERT INTO Finance.TaxType(Tax, Code, Name )
VALUES
('S', 'G', 'General')
GO
```

Figura 7 - Execució individual d'instruccions SQL

Exemples d'execució de proves i resultats.

En aquest afegim períodes comptables d'anys naturals i modifiquem la descripció d'un període existent, creat anteriorment.

Podem observar que la primera crida a 'EXEC Finance.addPeriod', se li passa el paràmetre inicial el valor NULL, per tant no ens retornarà l'identificador i només ens passa el nom (variable @name), per tant, crerà

el registre amb valors per defecte i que el trigger definit s'encarregarà de validar i omplir.
En la segona crida, ja se li passa l'any (2023), que és el paràmetre @yearinitial.
En la tercera, se li passa a més de la variable @PeriodId per recollir el valor (utilitzem la paraula clau OUPUT), a més del nom, així com la resta de paràmetres: yearInitial, yearFinal, montInitial i monthFinal, en aquest cas creem un registre d'any no natural.

```
-- Periodes comptables
DECLARE @PeriodId INT = NULL
DECLARE @RSP varchar(MAX)
DECLARE @name nvarchar(30)

SET @name  = 'EXERCICI ' + convert( varchar, year(getDate()));
EXEC Finance.addPeriod NULL, @name
EXEC Finance.addPeriod NULL, 'Exercici 2023', 2023

-- periode no natural Octubre'25 fins setembre'26
-- recollim el id per després cridar a modificar el nom
EXEC finance.addPeriod @PeriodId OUTPUT, 'Exercici 2025', 2025, 2026, 10,9
-- per comprovar els registres
SELECT * FROM Finance.Period
IF NOT @PeriodId IS NULL
    BEGIN
    -- modificació del registre, amb paràmetres per nom
    EXEC Finance.addPeriod @PeriodId = @PeriodId OUTPUT, @name = 'Period 2025-26', @RSP = @RSP OUTPUT
END
-- per comprovar els registres
SELECT * FROM Finance.Period
SELECT * FROM Finance.Log
GO
```

Figura 8 - Inserció de registres en taula Period

I podem veure els resultats. Observarem que la instrucció s'executa dues vegades, per veure els resultats després de la crida al procediment de modificació del nom. Així com els registres creats en la taula de log dels procediments cridats.

Resultados | Mensajes

	PeriodId	Name	YearInitial	YearFinal	MonthInitial	MonthFinal	NumEntry	YearMonthInitial	YearMonthFinal
1	16	EXERCICI 2024	2024	2024	1	12	0	202401	202412
2	17	Exercici 2023	2023	2023	1	12	0	202301	202312
3	18	Exercici 2025	2025	2026	10	9	0	202510	202609

	PeriodId	Name	YearInitial	YearFinal	MonthInitial	MonthFinal	NumEntry	YearMonthInitial	YearMonthFinal
1	16	EXERCICI 2024	2024	2024	1	12	0	202401	202412
2	17	Exercici 2023	2023	2023	1	12	0	202301	202312
3	18	Period 2025-26	2025	2026	10	9	0	202510	202609

	LogId	nameProcedure	parameterInput	parameterOutput
1	17	addPeriod	[{"name":"EXERCICI 2024"}]	OK
2	18	addPeriod	[{"name":"Exercici 2023","YearInitial":2023}]	OK
3	19	addPeriod	[{"name":"Exercici 2025","YearInitial":2025,"YearFinal":2026,"MonthInitial":10,"MonthFinal":9}]	OK
4	20	addPeriod	[{"PeriodId":18,"name":"Period 2025-26"}]	OK

Figura 9 - Resultat execució procediment addPeriod

5.7 Addicions i millores del projecte

Un dels problemes més importants que m'he trobat a l'hora de fer proves és la de mirar d'automatitzar-les per tal de verificar el seu funcionament. Si bé, podia executar els procediments i després revisar els resultats, aquesta tasca ha sigut sense dubte la més pesada del projecte.

Malgrat això, en la fase de fer proves del desenvolupament es va estar buscant eines per automatitzar aquestes proves. S'analitza un producte *opensource*[20]: https://tsqlt.org/. La impressió inicial del producte semblava que era el que realment es buscava, però no estava previst en la planificació la de fer un procés tan complet de procediments de prova i per això es va utilitzar. Altres opcions implicaven haver d'utilitzar Visual Studio i construir projectes amb C#, tal com s'indica en aquesta informació que vaig trobar en l'enllaç: https://www.sqlshack.com/es/pruebas-de-unidad-con-sql-server-data-tools/ o a https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-data-tools/jj851200(v=vs.103).

Ambdues opcions implicaven crear procediments de proves, per tant es tenia clar que aquests s'havien de fer i no havien d'interposar-se en el desenvolupament del disseny de la base de dades del projecte.

Cap al final del projecte mentre es preparava l'entrega final i a partir d'aquesta idea i amb la necessitat de fer proves sobre els procediments que es creaven i com part del projecte de desenvolupament de la base de dades, es va crear una eina pròpia de test.

Aquesta eina consisteix en tenir una base de dades de test amb procediments que s'executen de forma automàtica. Per fer-la s'ha creat una base de dades que servirà per fer bateries de proves amb més o menys automatitzacions. Si bé, és cert, que només el disseny d'aquesta base de dades per fer tests i la seva implementació seria per fer un nou projecte, és un bon punt d'inici per futures ampliacions del projecte.

La base de dades té com nom: TFG_UOC_AST_TEST on hi ha definits una sèrie de procediments per testejar la base de dades. Aquests procediments el seu valor de retorn és que ha anat bé o error (0 o 1). Què és el valor que retorna una *stored procedure*, un valor numèric.

Partint del resultat que ha de retornar un procediment i per tal de tenir un control sobre el procés de proves el que s'ha fet és dissenyar un procediment que executi les proves una a una i tenint en compte un ordre per si hi ha relació entre proves. Per tant per saber si un procediment segons el que retorna és correcte, ho fem segons la taula següent:

Sufix	Retorn
_OK	0
_Error	1

Per tant si un procediment té el sufix _OK, perquè sigui correcte ha de retornar un 0. En canvi si un procediment ha de provocar un error (i el capturem), ha de retornar un 1, que voldrà dir que ha donat error, i aquest és el resultat esperat.

[20] https://www.redhat.com/es/topics/open-source/what-is-open-source

Per fer els proves tenim un procediment (`TESTALLProcedures`) que és qui fa la crida als procediments que estan configurats per a que el seu resultat sigui correcte o doni error. Aquest procediment a la vegada crida al procediment `PreparaTest` que és l'encarregat d'inicialitzar les dades, i al procediment `OmpleTest` que és on omplim la taula amb els procediments definits que s'executaran.

En el procediment `PreparaTest` esborrem el contingut de la taula per després en el procediment `OmpleTest` omplir-la, que vindria a ser el procediment on indiquem quins procediments volem executar.

```
CREATE TABLE storedTest (
    storedId int identity(1,1),
    stored NVARCHAR(200),
    [order] INT NOT NULL,
    storedDesc NVARCHAR(200),
    CONSTRAINT PK_storedTest PRIMARY KEY (storedId )
);
```

Figura 10 - Taula d'execució de tests

En el procediment `OmpleTest` indiquem quins procediments volem executar i en quin ordre tal com es veu en la següent figura.

```
CREATE OR ALTER PROCEDURE OmpleTest
AS
BEGIN
    -- procediments a testejar
    INSERT into storedTest (stored, [order], storedDesc)
    VALUES
    ('addJournal_OK', 0, 'creem Journal amb journalid=0 i nom General'),
    ('addJournal_00_OK', 1, 'Canvia el nom a Diari General'),
    ('addJournal_00a_OK', 2, 'Comprova que el nom canviat sigui Diari General'),
    ('addJournal_00a_Error', 3, 'Dona error ja que el nom canviat no és XXDiari General'),
    ('addJournal_02_Error', 20, 'Crear el journal de nom Diari de Test'),
    ('deleteJournal_01_OK', 30, 'esborrem el journalid=0'),
    ('fGetPeriodId_Error', 40, 'no existeix cap periode'),
    ('fGetPeriodId_OK', 50, 'creem el periode any actual i comprovem que existeix'),
    ('fGetPeriodId_02_Error', 60, 'cerquem data 2025'),
    ('fGetPeriodId_02_OK', 70, 'Es crea periode 2025 i es comprova si existeix'),
    ('addEntryGlobal_02_Error', 80, 'no hi ha el journalid =0'),
    ('addJournal_OK', 90, 'creem Journal amb journalid=0, esborrat abans'),
    ('addEntryGlobal_02_Error',100, 'no hi ha el compte 410001'),
    ('addAccount_OK', 110, 'creem account amb accountcode = 400001'),
    ('addAccount_01_OK', 111, 'creem account amb accountcode = 472000'),
    ('addAccount_02_OK', 112, 'creem account amb accountcode = 600000'),
    ('addEntryGlobal_02_OK',120, 'Creem un assentament amb journalid=0 i compte 400001'),
    ('deleteJournal_01_Error', 150,'No podem borrar un journal que està en un entry')

END;
GO
```

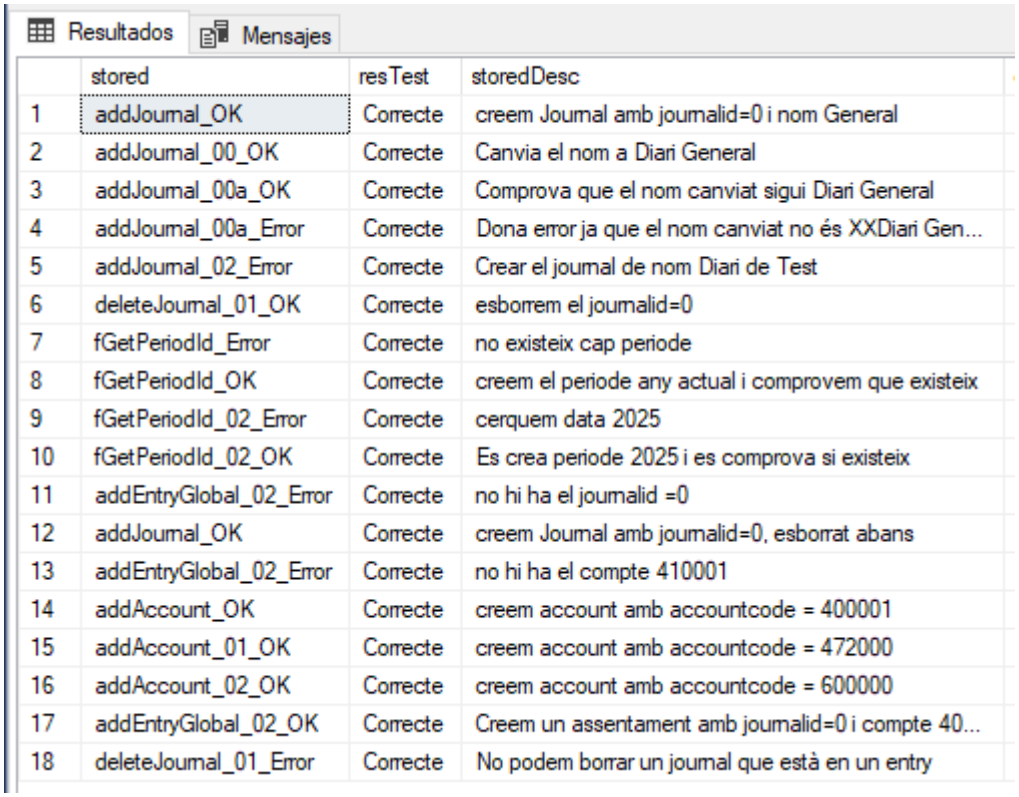
Figura 11 - Procediments de test

Per executar la bateria de tests:

```
USE TFG_UOC_AST_TEST
GO
EXEC TestALLProcedures
GO
```

Aquest procediment el que fa és recórrer tots els registres de la taula 'storedTest' i els va executant un a un.
Un cop els ha processat ens mostrarà el resultat que ha obtingut cada test així com un resultat final.

Si el resultat és correcte en la pestanya de resultats ens han d'aparèixer els tests executats i el seu resultat.



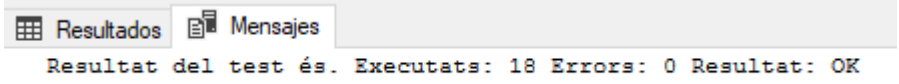
Resultados | Mensajes

	stored	resTest	storedDesc
1	addJournal_OK	Correcte	creem Journal amb journalid=0 i nom General
2	addJournal_00_OK	Correcte	Canvia el nom a Diari General
3	addJournal_00a_OK	Correcte	Comprova que el nom canviat sigui Diari General
4	addJournal_00a_Error	Correcte	Dona error ja que el nom canviat no és XXDiari Gen...
5	addJournal_02_Error	Correcte	Crear el journal de nom Diari de Test
6	deleteJournal_01_OK	Correcte	esborrem el journalid=0
7	fGetPeriodId_Error	Correcte	no existeix cap periode
8	fGetPeriodId_OK	Correcte	creem el periode any actual i comprovem que existeix
9	fGetPeriodId_02_Error	Correcte	cerquem data 2025
10	fGetPeriodId_02_OK	Correcte	Es crea periode 2025 i es comprova si existeix
11	addEntryGlobal_02_Error	Correcte	no hi ha el journalid =0
12	addJournal_OK	Correcte	creem Journal amb journalid=0, esborrat abans
13	addEntryGlobal_02_Error	Correcte	no hi ha el compte 410001
14	addAccount_OK	Correcte	creem account amb accountcode = 400001
15	addAccount_01_OK	Correcte	creem account amb accountcode = 472000
16	addAccount_02_OK	Correcte	creem account amb accountcode = 600000
17	addEntryGlobal_02_OK	Correcte	Creem un assentament amb journalid=0 i compte 40...
18	deleteJournal_01_Error	Correcte	No podem borrar un journal que està en un entry

Figura 12 - Resultat dels tests

D'aquesta manera podem provar el nostre disseny de base de dades sense haver-ho de fer de forma semi-manual i amb validació manual de si ha anat correcte o no.

Podem veure un resum del resultat de les proves realitzades en la següent figura.



Resultados | Mensajes

```
Resultat del test és. Executats: 18 Errors: 0 Resultat: OK
```

Figura 13 - Resum del resultat dels tests

Tal com s'ha comentat és un inici d'un projecte addicional al projecte que es tracta en aquesta memòria, ja que si bé el disseny i del desenvolupament de la base de dades és important tant o més ho és poder provar aquest desenvolupament.

5.8 Definició format de paràmetre JSON

Una de les parts més importants d'aquest projecte, si no la més important és la creació dels assentaments comptables a partir d'una informació estructurada que es definida en un paràmetre en format JSON.

Aquest paràmetre JSON pot ser informat des de múltiples clients diferents, d'aquesta manera no hi ha una dependència del llenguatge en el que s'estigui fent el desenvolupament, per tant, es poden utilitzar diferents llenguatges.

Un dels avantatges del format utilitzat és el que no necessita que siguin informats molts paràmetres com seria en un procediment, sinó que amb un sol paràmetre amb l'estructura adequada permet subministrar al procediment la informació necessària per poder crear l'assentament comptable i els registres associats, així com l'acumulació dels imports.

L'estructura és extensible de tal manera que el seu manteniment és molt més fàcil de portar a terme.

Aquesta estructura tindria uns valors obligatoris, més unes restriccions, que de no informar-se els procediments auxiliars ja validen o bé les mateixes restriccions existents en la taula també faria la tasca de validació de que la informació sigui correcte.

Valors obligatoris:

- **entry**: aquest seria l'objecte principal i que engloba tota la información relacionada de l'assentament comptable.
- **journal**: correspondria al diari comptable, si no s'informa o no es fa correctament no es generarien els registres al no complir una regla d'integritat referencial al ser clau de la taula Entry que té una clau forana cap a la taula Journal.
- **note**: és un array d'objectes *note*, que a la vegada té diversos elements. Una restricció és que la suma dels elements note per tipus (que només pot ser D o H, dèbit o haver) ha de coincidir. Per entrendre-ho millor, la suma de l'import de tots els elements de tipus D ha de coincidir amb la suma de l'import de tots els elements de tipus H. En els objectes *note*, a la vegada tenim:
    - **accountId** o **accountCode**, identificador del compte comptable o codi del compte comptable, en el cas de que ens informin accountId, ja no és necessari accountCode. Si

ens informen només accountCode, s'utilitza per buscar el valor d'accountId.
    - **import**: import de l'apunt comptable.

En el procediment que permet crear els assentaments comptables a partir d'una cadena en format JSON, en el procés de generar l'apunt comptable, si no li informem de l'entrada type (D o H), per defecte s'agafa el criteri que si és inferior a zero és un haver (H), i si no és dèbit (D), per això aquest valor no és obligatori.

Un objecte *note*, pot tenir o no objectes VAT, és a dir un apunt pot tenir relació amb un registre d'impostos. Si és aquest el camp aleshores aquest registre d'impostos haurà de tenir de forma obligatòria informat:

- **type**, on s'indicarà si és R o S, repercutit o suportat.
- **VATLine**: array d'objectes on tindrem les bases, percentatges de l'impost i quota. Hi haurà com a mínim un objecte de tipus VATLine per cada VAT.
    - **base**: base de l'impost
    - **percentage**: tant per cent a aplicar a la base
    - **quote**: resultat d'aplicar el percentatge a la base, és interessant informar-la ja que de vegades per arrodoniment pot passar que no desquadri per un cèntim.

Exemple d'assentament comptable i la seva relació amb estructures de dades JSON.

Assentament de la factura 'FRA1' de vendes al client **430025**, de data **25/07/2024** d'un import de **2.310** €, amb 2 IVA del tipus **Repercutit**, ambdós de **1.000** € de base i un **21**% i un **10**% de percentatge d'IVA i unes quotes de **210** € i **100** €, respectivament.

Assentament	Concepte	Valor
	Data	25/07/2024
	Diari	122
Apunt		
1	Compte	430025 (Client 25)
	Tipus	D – Dèbit
	Descripció	N/F Fra1
	Import	2.310
	IVA – Tipus	R – Repercutit
	Base1	1.000
	Percentatge1	21
	Base2	1.000
	Percentatge2	10
2	Compte	477000 (IVA Repercutit)

	Tipus	H – Haver
	Import	310
3	Compte	700000 (Vendes)
	Tipus	H – Haver
	Import	2.000

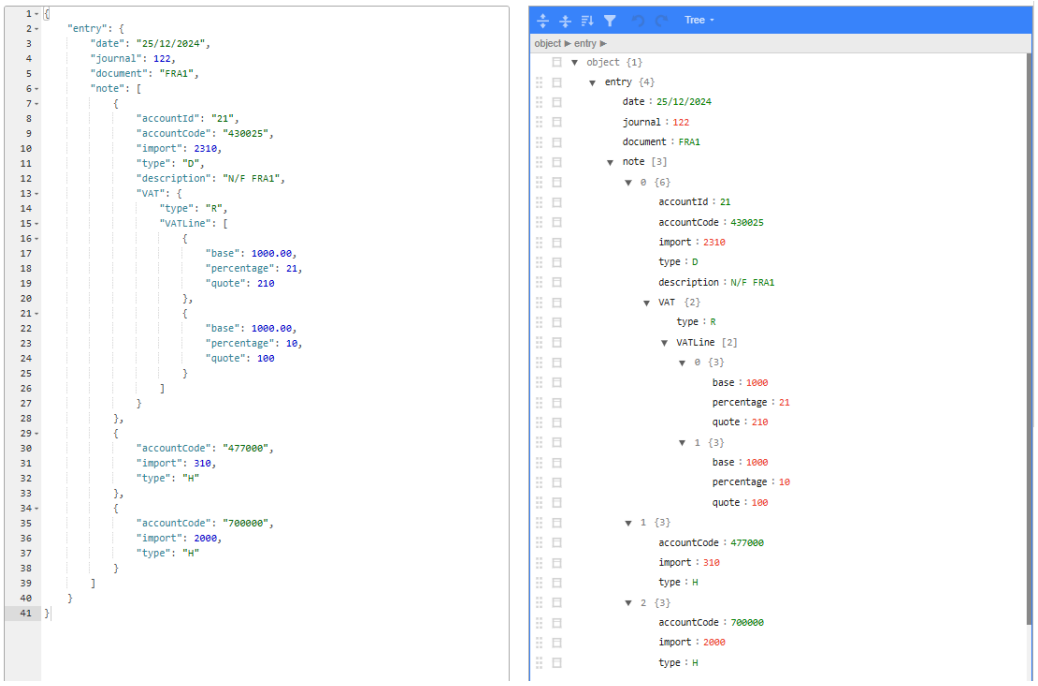


Figura 14 - Correspondència JSON – Objectes

Podem observar que els objectes JSON de tipus ‘note’ en la seva traducció a objecte són array, així podríem tenir múltiples elements de tipus ‘note’ que conformarien els apunts comptables.
El mateix seria pels objectes JSON de tipus ‘VATLine’ que la seva equivalència és d’un array.

Definint de forma més simple l’assentament sense indicar dates, només d’apunts comptables, ens quedaria de la següent manera.

Dèbit	Compte	Haver
2.310	430025 – Clients	
	477000 – IVA Repercutit	310
	700000 - Vendes	2.000

Evidentment tal com hem comentat abans hi ha uns camps que són obligatoris informar com ara la data i el tipus de diari, tot i que es podria arribar a una ‘convenció’ de que tinguéssim un diari per defecte o d’altres valors informats per defecte, a fi i efecte de fer més àgil i ràpida la introducció d’informació.

Aquest assentament correspondria a una estructura força completa d’una factura amb 2 IVES (21% i 10%) i que generaria uns acumulats als comptes 430025, 470000 i 700000 i a tots els seus immediats superiors segons l’arbre de definició del pla comptable

Compte	Dèbit	Haver
430025	2.310	
43002	2.310	
4300	2.310	
430	2.310	
43	2.310	
4	2.310	
477000		310
47700		310
4770		310
477		310
47		310
4		310
700000		2.000
70000		2.000
7000		2.000
700		2.000
70		2.000
7		2.000

En el següent exemple podem veure la cadena JSON d'un tiquet real de supermercat amb 4 percentatges d'IVA.


MERCADONA, S.A.
A-46103834

C/ VIA ROVIANO 2
43893 ALTAFULLA
TELEFON: 977598990
11/05/2024 14:33 OP: 374049
FACTURA SIMPLIFICADA: 4119-014-326295

TOTAL (€) 50,09
TARGETA BANCÀRIA 50,09

IVA	BASE IMPOSABLE (€)	QUOTA (€)
0%	19,94	0,00
5%	1,24	0,06
10%	22,23	2,22
21%	3,64	0,76
TOTAL	47,05	3,04

TARG. BANCARIA: **** **** **** 4987
N.C: 055946362 AUT: R96997
AID: A0000000041010 ARC: 3030
Verificat per dispositiu

Import: 50,09 € MASTERCARD

Figura 15 - Exemple Tiquet 4 IVES


```
{
  "entry": {
    "date": "11/05/2024",
    "journal": 0,
    "document": "4119-014-326295",
    "note": [{
      "accountCode": "400001",
      "import": 50.09,
      "type": "H",
      "description":   "S/F   4119-014-
326295",
      "VAT": {
        "type": "S",
        "VATLine": [{
          "base": 19.94,
          "percentage": 0,
          "quote": 0
        }, {
          "base": 1.24,
          "percentage": 5,
          "quote": 0.06
        }, {
          "base": 22.23,
          "percentage": 10,
          "quote": 2.22
        }, {
          "base": 3.64,
          "percentage": 21,
          "quote": 0.76
        }]
      }
    }, {
      "accountCode": "472000",
      "import": 3.04,
      "description": "IVA S/F 4119-014-
326295"
```

6. Seguiment del projecte

6.1 PAC1 – Pla de treball

Inicialment aquest PAC consistia en la generació del Pla de Treball. Amb la planificació prevista, recollida dels requeriments.

Per fer la planificació es va decidir fer-la en hores, ja que era més fàcil de quantificar ja que els dies que corresponien al cap de setmana la dedicació era molt major i per tant el càlcul del temps més acurat que no pas fer la planificació en dies. L'eina per fer el diagrama de gantt ha sigut Microsoft Project 2019, llicència per estudiants.

Així, s'havia de fer una estimació per tal d'aconseguir les fites tal com es proposava i deixant un marge de temps extra per a possibles imprevistos.

6.2 PAC2

En aquesta PAC s'han acomplert les tasques que s'havien definit segons el Pla de Treball. Aquestes tasques eren la del disseny conceptual del projecte, identificant les principals entitats que formarien part del sistema. Es va fer el disseny en UML, mitjançant l'eina Draw.io que s'ha inclòs en l'apartat 2.3.

A més, s'han tingut en compte les propostes fetes pel professor sobre la PAC1/Pla de Treball que s'han reflectit en el document que va ser entregat en les dates previstes.

S'han treballat els models conceptual i lògic, a banda de treballar en la gestió dels riscos del projecte.

També es va fer la selecció del SGBD, així com la seva instal·lació (captures de pantalla en els annexo), on la selecció va ser la de SQL Server tot i que podia ser qualsevol altre de les avaluades: MySql, Oracle o PostgreSQL, la selecció va ser sobretot per adquirir nous coneixements, ja que els altres 3 ja els havia treballat en diferents assignatures.

A partir d'aquesta PAC2, es va començar a donar forma a la memòria

6.3 PAC3

En aquesta PAC, s'ha seguit la planificació establerta en el Pla de treball i s'ha anat treballant en el disseny físic. Considero que ha sigut en aquesta entrega on ja es tenia els disseny conceptual i lògic ben definit on s'ha fet el màxim de desenvolupament del projecte que la complexitat ha sigut més la de passar del model lògic al físic amb les particularitats del llenguatge SQL.

M'ha agradat redescobrir les columnes calculades per no haver de tenir columnes que s'actualitzin via *trigger* o haver de tenir camps extra.

6.4 Entrega final

En aquesta fase final, ha sigut quan s'han creat la majoria de procediments de creació de procediments a partir de cadena JSON, així com els procediments de consulta, inicialització de dades, proves de coherència.

Potser aquesta part del projecte és la més feixuga per la feina de recopilació de documentació, creació de la presentació, gravació en vídeo, revisió dels procediments creats...

En la fase de l'entrega final ha sigut quan s'han realitzat la major part dels procediments per provar el desenvolupament, així com les consultes.

Totes les fases i tasques han tingut la seva importància i és en aquesta fase final, la que s'ha d'acabar d'elaborar el document que correspon a la memòria, on hi ha la feina de revisar el que s'ha escrit en les altres fases, reescriure si hi havia alguna explicació que no quedava del tot clara.

7. Conclusions

7.1 Introducció

Dels objectius plantejats inicialment, crec que s'han assolit tots, ja que en el Pla de Treball, ja es va descartar desenvolupar la part de la configuració dels automatismes que formaria part d'un altre projecte, essent un projecte de futur, que complementaria aquest.

S'ha desenvolupat segons els requeriments inicials i per aquesta raó estic satisfet de la decisió presa que era centrar-se en la base de dades de registre d'assentaments comptables i altres registres relacionats, com seria el registre que correspondria a les factures.

En el desenvolupament del projecte s'ha fet palès que és necessita molt de temps per tal de fer les proves d'inserció/modificació i esborrat de registres de la base de dades.

El fet d'utilitzar procediments per fer les proves de creació de registres i validacions m'ha donat la tranquil·litat de que s'omplien correctament totes les taules relacionades.

Aquesta no hauria de ser una feina manual i que es puguin fer aquestes proves d'una manera més automatitzada, per evitar errors seria una de les coses que m'hagués agradat poder treballar més. Cap al final del desenvolupament vaig trobar una eina per fer tests unitaris mitjançant l'eina Visual Studio i amb l'eina que em vaig 'construir', tal com he explicat en l'apartat del Pla de proves, crec que es podrien automatitzar moltes de les proves que s'haurien de fer. Trec la conclusió de que amb les proves unitàries ens assegurem del correcte funcionament dels procediments.

Podria dir que l'ús d'un paràmetre JSON en els procediments més importants és el que fa que un dels objectius inicials del projecte, ser més àgil i escalable, s'ha aconseguit en escreix.

Valoro de forma molt significativa el seguiment fet pel professor consultor i les revisions proposades, per exemple, quan se'm va demanar justificar el perquè utilitzava *sequence* en comptes d'*identity* pels camps incrementals que formaven la clau primària. Aquestes recomanacions que se m'han fet a cada entrega, han permès millorar el producte final.

7.2 Planificació de projecte.

He de comentar que la planificació ha sigut força encertada, ja que en el Pla de Treball vaig estimar un temps addicional per a imprevistos, i aquest temps l'he fet servir tan per documentar-me millor a l'hora de treballar amb els procediments com pel que fa al tractament de les dades en format JSON.

Es va escollir la metodologia en cascada tenint en compte el tipus de projecte que s'estava desenvolupament i per tant va ser encertada, amb tot, sempre s'ha pogut revisar els dissenys, sobretot en l'apartat de disseny físic per tal de tenir en consideració aspectes propis del SGBD, com ara la utilització de 'sequence'[21] en comptes de fer servir camps incrementals, per identificar de manera única un registre, tal com comentava anteriorment.

L'explicació de l'elecció del *sequence* es va fer en base a un article que vaig trobar (https://blog.sqlauthority.com/2018/01/24/sql-server-identity-jumping-1000-identity_cache/ mentre mirava d'implementar els procediments d'inserció de registres i que respectessin la integritat referencial. Treballar amb *sequence* em va comportar una mica més feina, però ho vaig considerar una bona opció ja que també em va permetre la possibilitat d'assignar un valor al camp de forma directe. Per posar un exemple, assignar el valor 0 per un registre de la taula *Journal* i d'aquesta manera podria no fer necessari la seva inclusió en la definició de l'estructura.

També l'anar confeccionant la memòria en el moment que estava realitzant la tasca m'ha permès representar millor en el que estava treballant, i per tant al fer cas del suggeriment fet pel professor en l'entrega del Pla de Treball m'ha servit per poder portar aquesta memòria al dia.

7.3 Futur del projecte.

Tenint en compte el que he après a l'hora de treballar amb els procediments i amb el format JSON, considero que el sistema creat té moltes possibilitats, ja que s'ha pensat sempre en que sigui, sobretot, molt escalable i amb possibilitat d'ampliació amb més informació extra-comptable, com podria ser la comptabilitat analítica o la cartera de cobraments/pagaments.

Una altra opció de futur seria la de tenir en compte si és una factura afectada per IRPF i per tant seria una estructura similar a la que es genera ara pels impostos, i per tant ampliar-la significaria fer una crida a un procediment més per cada apunt comptable que en tingui associat.

També tal com s'ha comentat en la introducció d'aquest capítol, i que seria un projecte complet en sí, seria el de la configuració per poder automatitzar l'entrada d'assentaments i que complementaria als procediments creats en aquest projecte. Aquest considero que seria el projecte de futur.

Altres millores per un futur; està clar: dotar de més automatismes. Com per exemple, que un compte de client o proveïdor pogués tenir associat un compte on generar l'apunt de la venda o la compra. El mateix pel tipus

[21] https://learn.microsoft.com/en-us/sql/t-sql/statements/create-sequence-transact-sql

d'IVA. D'aquesta manera seria encara molt més àgil l'entrada de la informació. Significaria també que les dades transmeses serien menys ja que l'estructura JSON seria més reduïda. O encara més reduïda, si parlem de factures, on només informant poques dades, el sistema ens fes tots els càlculs per obtenir la part dels impostos i les contrapartides.

Una altra millora seria la de poder fer factures en diferents monedes i al generar l'assentament amb els apunts poder aplicar un canvi de moneda.

És en aquest punt on la metodologia *agile* potser seria més adequada, ja que partiríem d'un producte mínim viable, i aquestes millores podrien ser noves funcionalitats que s'anirien incorporant al producte. Ja que la metodologia en cascada si bé és bona per aquest projecte, per noves funcionalitats no seria la més adient.

També un punt pendent podria ser tot el que fa referència a la seguretat, amb creació d'usuaris, tot i que com tot el desenvolupament s'ha basat en procediments es podria limitar de forma ràpida per perfils d'usuari el que no es pogués inserta directament a les taules mitjançant instruccions *INSERT* o *UPDATE*.

Una altra bona proposta seria la de no haver de prefixar alguns valors a les taules com ara que la taula ***TaxType***, el camp ***Tax*** no hagi de ser obligatòriament S o R (Suportat o Repercutit). Això es podria resoldre creant una taula ***Tax*** i que el camp identificador sigui clau forana de la taula ***TaxType*** pel camp ***tax***.

També m'ha fet adonar que hi ha molt per explorar, com per exemple que un procediment no pugui ser cridat directament, és a dir, fent el símil amb un mètode d'una classe, aquest seria privat i només es podria executar des dins de la mateixa classe. S'ha de suposar que aquest aspecte es podria tenir en compte assignant permisos a depenent dels priviligis d'un usuaris es poguessin executar uns procediments o bé uns altres. En el nostre projecte, gairebé es podria limitar als d'altes, baixes i modificacions de les taules mestres i al procediment ***addEntryGlobalJSON***, ja que aquest crida a d'altre procediments de forma interna.

En resum, aquest treball m'ha fet veure que si es volen fer les coses correctament, és del tot necessari marcar-se uns objectius realistes i seguir una bona planificació.

8. Glossari

- Waterfall. Nomenclatura en anglès per definir la metodologia en cascada.

- Agile. Nomenclatura en anglès per definir la metodologia àgil. Ideal per projectes que tenen necessitat d'adaptar-se fàcil i de flexibilitat.

- UML: abreviatura en anglès d'*Unified Modeling Language*. Llenguatge unificat de modelització.

- IVA: abreviatura de Impost sobre el Valor Afegit, impost indirecte que s'aplica al consum.

- RE: abreviatura del recàrrec d'equivalència. Règim especial d'IVA que és aplicat a comerciant minoristes de productes que no transformen.

- ERP: abreviatura en anglès d'E*nterprise Resource Planning*

- SGBD, abreviatura de Sistema Gestor de Base de Dades.

- JSON, abreviatura en anglès de JavaScript Object Notation, es tracta d'un format de text que emmagatzema informació estructurada, utilitzada habitualment per intercanvi de dades entre un servidor i un client. Representa les dades en parells clau-valor.

- XML, abreviatura en anglès de Extensible Markup Language. El seu ús és semblant al format JSON. Representa les dades en un patró d'arbre.

- AWS, abreviatura en anglès de Amazon Web Services.

- Azure, plataforma al núvol de Microsoft.

- Backup, nomenclatura en anglès que fa referència a les còpies de seguretat.

- Docker, sistema de contenidors que permet emmagatzemar tot el necessari per que una aplicació es pugui executar.

- SMSS abreviatura de Microsoft SQL Server Management Studio.

- On-premise, o en local, fa referència al tipus d'instal·lació d'una solució de programari. Aquesta instal·lació es realitza en el servidor i en la infraestructura de l'empresa.

9. Bibliografia

José Ramón Rodríguez (2018). “La gestió de projectes. Conceptes bàsics” Editorial UOC

Casas Roma, Jordi (2018). “Introducció al disseny de bases de dades” Editorial UOC.

Casas Roma, Jordi; Cuartero Olivera, Josep (2020). “Disseny conceptual de bases de dades” Editorial UOC.

Burgués Illa, Xavier; Cuartero Olivera, Josep (2020). “Disseny lògic de bases de dades” Editorial UOC.

Cabré i Segarra, Blai et al. (2020). “Disseny físic de bases de dades” Editorial UOC.

Albós Raya, Amadeu (2019). “Introducció als sistemes d’informació a les organitzacions” Editorial UOC.

Guitart Hormigo, Isabel (2011) “Sistema d'informació empresarial”. Editorial UOC

Project Management Institute(2017) “A GUIDE TO THE PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK® Guide) – Sixth Edición”

Waterfall [Data de consulta: 6 d’abril de 2024]
https://en.wikipedia.org/wiki/Waterfall_model

¿Cuál es la metodología más adecuada para tu proyecto? [Data de consulta: 6 d’abril de 2024]
https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html

Agile [Data de consulta: 6 d’abril de 2024]
https://en.wikipedia.org/wiki/Agile_software_development

PC Componentes [Data de consulta: 14 d’abril de 2024]
https://www.pccomponentes.com/

JSON Schema, declarative language that allows you to annotate and validate JSON documents, [Data de consulta: 1 de maig de 2024]. https://json-schema.org/

Documentación técnica de SQL Server [Data de consulta: 1 de maig de 2024]. https://learn.microsoft.com/es-es/sql/sql-server

JSON Editor, [Data de consulta: 1 de juny de 2024].
https://techiedelight.com/tools/jsonEditor#

Tutorial: Crear y ejecutar una prueba unitaria de SQL Server, [Data de consulta: 1 de juny de 2024].
https://learn.microsoft.com/es-es/sql/ssdt/walkthrough-creating-and-running-a-sql-server-unit-test?view=sql-server-ver16

CREATE TRIGGER (Transact-SQL) [Data de consulta: 1 de juny de 2024]:
https://learn.microsoft.com/es-es/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver16

10. Annexos

Descripció	Fitxer
• Manual d'instal·lació	Instal·lació SQL Server.pdf
• Planificació del projecte	Planificació del projecte.pdf
• Script de creació de la base de dades i esquema	*00 - Base de dades.sql*
• Script de creació de taules	*01 - Creació de taules.sql*
• Script de creació de triggers	*02 - Creació de triggers.sql*
• Script amb stored procedures i funcions	*03 - Stored Procedures i funcions.sql*
• Script amb stored procedures de consulta	*04 - Stored Procedures Consulta.sql*
• Script d'inserció de registres mestres i proves manuals	*05 - Insertar registres.sql*
• Script de creació de la base de dades de test i procediments auxiliars per fer les proves sobre la base de dades desenvolupada	T0 - Base de Dades Test Unitaris.sql
• Script de creació dels tests unitaris	T1 - Test unitaris.sql
• Script de definició del procediment que executa els tests unitaris i execució dels test unitaris	T2 - Executa TEST.sql