

FluidERP

Disseny i proposta d'implementació d'una solució ERP integral per a Instrumentos Científicos S.A. (ICSA)

Alvar Cuevas i Fajardo
Grau d'Enginyeria Informàtica

Julian Perez Aragon
María Isabel Guitart Hormigo

Data Lliurament: Juny de 2021



Aquesta obra està subjecta a una llicència de [Reconeixement 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by/3.0/es/)

FITXA DEL TREBALL FINAL

Títol del treball:	<i>FluidERP: Disseny i proposta d'implementació d'una solució ERP integral per a Instrumentos Científicos S.A. (ICSA)</i>
Nom de l'autor:	<i>Àlvar Cuevas i Fajardo</i>
Nom del consultor/a:	<i>Julian Perez Aragon</i>
Nom del PRA:	<i>María Isabel Guitart Hormigo</i>
Data de lliurament (mm/aaaa):	<i>06/2021</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Sistemes d'informació integrats (ERP)</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Organització, ERP, Processos</i>

Resum del Treball:

Aquest TFG és una proposta de disseny, implementació i posada en funcionament d'un nou ERP per a una empresa que dona solucions integrals en el món del laboratori. L'empresa del cas estudiat és real, i la proposta realitzada ha sigut aprovada per a la seua implementació en els propers mesos.

El camp comercial en que es mou l'empresa té unes necessitats molt específiques per a un ERP. Encara que la compravenda de material de laboratori i instrumentació és important en el volum de negocis, també desenvolupa programari a mesura i ofereix una sèrie de servicis no inventariables com consultoria per a l'assoliment d'acreditacions i certificacions. L'empresa treballa al mercat espanyol i també a la resta d'Europa i el món.

El projecte comença en la fase d'oportunitat a través d'unes mancances continuades al programari que s'està utilitzant actualment. Es fa un estudi sobre la viabilitat del projecte i es compilen les necessitats de totes les parts interessades, per a tindre una decisió consensuada sobre el projecte.

A partir d'ací, es continua amb les fases necessàries per a dur endavant el projecte. Dins de l'abans del projecte està el disseny dels components i funcionalitats a implementar, les proves d'usuari, el pla d'implementació amb la gestió de riscos i el pla de posada en marxa, formació i gestió del canvi.

Com a conclusions del projecte s'ha obtingut una gran quantitat d'informació de l'empresa, així com una sèrie de pautes a tindre en compte en futurs plans d'actualització del programari.

Abstract:

This TFG is a proposal for the design, implementation and commissioning of a new ERP for a company that provides comprehensive solutions in the world of the laboratory. The company of the case studied is real, and the proposal made has been approved for implementation in the coming months.

The business field in which the company operates has very specific needs for an ERP. Although the purchase and sale of laboratory equipment and instrumentation is important in the turnover, it also develops tailor-made software and offers a series of non-inventory services such as consulting for the achievement of accreditations and certifications. The company works in the Spanish market and also in the rest of Europe and the world.

The project begins in the opportunity phase through ongoing shortcomings in the software currently in use. A feasibility study is carried out and the needs of all stakeholders are compiled in order to have a consensual decision on the project.

From here, we continue with the necessary phases to carry out the project. Within the pre-project is the design of the components and functionalities to be implemented, the user tests, the implementation plan with risk management and the commissioning, training and change management plan.

As a result of the project, a large amount of information has been obtained from the company, as well as a series of guidelines to be considered in future software update plans.

Índex

1. Introducció.....	1
1.1 Context i justificació del Treball.....	1
1.2 Objectius del Treball.....	2
1.3 Enfocament i mètode seguit.....	2
1.4 Planificació del Treball.....	4
1.5 Breu sumari de productes obtinguts.....	5
1.6 Breu descripció dels altres capítols de la memòria.....	5
2. Antecedents.....	7
2.1 Situacions precedents.....	7
2.2 Situació actual.....	8
2.3 Alternatives proposades.....	9
2.4 Càlcul del cost del projecte.....	10
2.5 Presa de decisió.....	11
3. Limitacions sobrevingudes al projecte.....	12
4. Disseny de l'aplicació.....	14
4.1 Consideracions prèvies.....	14
4.1.1 Plataformes i llenguatges.....	14
4.1.2 Framework.....	15
4.1.3 Desplegament i contenidors.....	16
4.1.4 SGDB.....	18
4.1.4.1 Proves de rendiment.....	19
4.1.4.2 Selecció SGDB.....	21
4.2 Disseny general.....	21
4.3 Disseny del component dades.....	22
4.3.1 Estructura de la base de dades.....	22
4.3.2 Plantilles.....	25
4.3.3 Etiquetat i regles d'etiquetat.....	25
4.3.4 Primitives d'accés a dades.....	25
4.3.4.1 Creació de dades.....	26
4.3.4.2 Accés a dades.....	26
4.3.4.3 Modificació de dades.....	27
4.3.5 Implementació de les funcions del component dades.....	27
4.4 Disseny del component servidor.....	28
4.4.1 Mòduls.....	28
4.4.2 Mòdul plantilles.....	30
4.4.3 Mòdul entitat genèrica.....	30
4.4.4 Mòdul permisos.....	30
4.4.5 Mòdul regles de filtrat.....	30
4.4.6 Mòdul de documentació.....	31
4.4.7 Mòdul d'informes.....	31
4.4.8 Mòdul de vendes.....	32
4.4.9 Mòdul de compres.....	32
4.4.10 Mòdul de compravenda.....	32
4.4.11 Mòdul de proposta.....	33
4.4.12 Mòdul de facturació.....	33
4.5 Disseny del component client.....	33

4.5.1 Consideracions prèvies.....	33
4.5.2 Pàgina d'accés.....	34
4.5.3 Pàgina d'inici.....	35
4.5.4 Menú de configuració.....	35
4.5.5 Menú d'etiquetes.....	35
4.5.6 Vista en taula d'elements.....	36
4.5.7 Vista individual d'elements.....	37
4.5.8 Selecció d'elements d'una llista tancada.....	38
5. Proves amb usuaris clau.....	39
5.1 Procediment.....	39
5.2 Secretaria.....	41
5.3 Logística.....	42
5.4 Comercial.....	42
5.5 Direcció.....	43
5.6 Conclusions de les proves.....	44
6. Pla d'implementació.....	45
6.1 Recursos implicats.....	45
6.2 Abast de la implementació.....	45
6.3 Riscos en la implementació.....	46
6.3 Calendari d'implementació.....	47
7. Posada en marxa i gestió del canvi.....	49
7.1 Personal implicat.....	49
7.2 Pla de formació.....	49
7.3 Paral·lel i canvi gradual.....	50
7.4 Pas a funcionament en producció.....	51
7.5 Gestió del canvi.....	51
8. Conclusions.....	53
8.1 Objectius assolits.....	53
8.2 Coneixements útils per a nous projectes.....	55
8.3 Continuació del projecte.....	56
9. Glossari.....	57
10. Bibliografia.....	60

Índex de figures

Figura 1: Metodologia de treball.....	3
Figura 2: Fases del projecte.....	4
Figura 3: Cronograma del projecte.....	4
Figura 4: Evolució de la imatge corporativa de ICSA en 35 anys.....	7
Figura 5: <i>Captura</i> de l'ERP que s'utilitza actualment a l'empresa.....	8
Figura 6: Cost d'oportunitat de la proposta.....	11
Figura 7: Desenvolupament en cascada i desenvolupament àgil.....	12
Figura 8: Evolució de la metodologia de desplegament, fins a contenidors.....	17
Figura 9: Pros i contres de la utilització de bases de dades estructurades i no relacionals.....	18
Figura 10: Resultats en segons de les proves d'SGBD.....	20
Figura 11: Interconnexions dels components.....	21
Figura 12: Mòduls del component servidor.....	29
Figura 13: Proposta inicial de la interfície d'accés.....	39
Figura 14: Proposta inicial de pàgina d'inici i vista en taula.....	40
Figura 15: Proposta inicial de vista individual.....	41

Figura 16: Resultat de les proves amb usuaris.....	44
Figura 17: Riscos detectats a la implementació del projecte.....	46
Figura 18: Plantilla de severitat de riscos.....	46
Figura 19: Planificació de la dedicació per mesos.....	47
Figura 20: Personal de programació implicat per mes.....	48
Figura 21: Planificació de posada en marxa.....	50
Figura 22: Transversalitat de la gestió del canvi.....	51

Taula d'exemples de codi

<i>JSON</i> 1: Element.....	23
<i>JSON</i> 2: Atribut.....	23
<i>JSON</i> 3: Format.....	24
<i>JSON</i> 4: Enllaç.....	24
<i>JSON</i> 5: <i>Find</i>	26
<i>JSON</i> 6: <i>Mod</i>	27

1. Introducció

1.1 Context i justificació del Treball

Instrumentos Científicos S.A. (ICSA) és una empresa fundada a l'any 1984 per Rogelio Franco per a comercialitzar equipament de laboratori. En els seus 37 anys de vida ha anat creixent fins als 30 treballadors i ampliant el seu catàleg de productes i servicis fins a poder complir de forma general qualsevol necessitat d'un laboratori modern, tant en l'àmbit analític com en el clínic. En l'actualitat, a banda de l'origen d'instrumentació, l'empresa comercialitza i ofereix qualsevol tipus de fungible, material o servici que necessite un laboratori, inclòs maquinari i programari informàtic. Actualment ICSA té un equip de desenvolupament de programari propi.

Per la gran quantitat de serveis que dona ICSA, és complicat definir fluxos de compra-venda concrets que siguin útils per a tots els processos de l'empresa. Així mateix, les seues característiques tan de compra-venda, desenvolupaments propis, servicis i gestió d'existències de maquinària, fungibles i altres elements fan que els desenvolupaments genèrics d'ERP o CRM no siguin útils sense implementacions específiques que tinguin en compte aquestes peculiaritats, amb el cost i problemes de manteniment que això provoca.

En el moment de redacció d'aquesta memòria, ICSA compta amb una ferramenta ERP pròpia, desenvolupament intern realitzat a començaments d'aquest segle. Encara que compleix de forma general les necessitats organitzatives i de facturació actuals, s'ha quedat completament antiquat en procediments i funcionament, llastrant l'evolució futura de l'empresa. Des de direcció fa anys que s'està buscant una alternativa i s'ha plantejat comprar una solució externa. Després de tantejar el mercat actual s'ha arribat a la conclusió que cap aplicació genèrica compleix els requisits necessaris que ja estan implementats en la ferramenta actual, i per tant cal com a mínim fer una implementació específica. Després d'estudiar els costos d'aquesta implementació, s'ha trobat que el cost total és comparable amb fer un desenvolupament complet intern i s'ha decidit estudiar aquesta via.

Un dels productes que comercialitza ICSA és una solució informàtica anomenada NevisLIMS. Es tracta d'una ferramenta de gestió de Laboratori (LIMS) que s'ha desenvolupat íntegrament dins de la empresa, i es presenta als laboratoris com una solució per a gestionar totes les tasques rutinàries i d'auditoria de qualsevol tipus de laboratori. Així mateix, facilita al laboratori el compliment de les normatives i acreditacions necessàries. NevisLIMS porta ja 6 anys al mercat, i actualment s'està implementant des de zero una nova versió que comprèn tota la experiència acumulada del projecte, amb les noves tàctiques de programació, gestió de bases de dades i procediments de laboratori.

Per tot això, s'ha proposat utilitzar tot aquest coneixement per a continuar amb el desenvolupament intern de l'ERP d'ICSA, i obtenir una ferramenta actualitzada i potent, millorant els processos actuals de l'empresa i que es pugui mantenir actualitzada a mig termini.

Al capítol 2. Antecedents d'aquesta memòria es troben els antecedents i el raonament realitzat tant pel departament d'informàtica de l'empresa com per direcció d'aquesta per a decidir realitzar el nou desenvolupament que significarà un nou ERP per a ICSA.

1.2 Objectius del Treball

Els objectius a tractar en aquest treball de fi de grau, alineats amb el material de l'assignatura [1], són els següents:

- Documentar tots els processos que realitza l'ERP actual, així com la seua utilització actual, per assegurar que queden dins del disseny de la nova ferramenta en el cas de ser utilitzats.
- Documentar tots els processos de compra-venda, facturació i tresoreria de ICSA per a localitzar processos que no estigueren contemplats en l'ERP actual.
- Disseny de la part interna del nou aplicatiu ERP: processos, emmagatzemament i consulta de dades. Integracions verticals i horitzontals. Compliment amb els processos antics i nous localitzats.
- Disseny bàsic de les interfícies d'usuari a utilitzar per usuaris i administradors del nou aplicatiu. Enllaç de les interfícies amb els processos.
- Proves d'interfícies i processos amb els usuaris clau de l'aplicatiu.
- Modificacions en els dissenys després de les proves amb usuaris clau.
- Definició del pla de treball per a implementar els dissenys realitzats.
- Documentació inicial de la implementació i pla de posada en funcionament.

1.3 Enfocament i mètode seguit

L'enfocament triat per a la solució ERP de ICSA ha sigut la d'unificar part de l'actual desenvolupament intern de l'empresa per a comercialitzar un nou LIMS, per a ajudar a la realització d'una ferramenta ERP i modern.

Gràcies a això, es preveu que es pugui reutilitzar tot el disseny i programació de les bases de dades i primitives d'accés i modificació de dades, auditoria de dades i control d'accésos d'aquest projecte per al futur desenvolupament de la ferramenta LIMS. També es podran reutilitzar algunes interfícies genèriques.

Per altra banda, com el ERP utilitzat actualment també és un desenvolupament intern, es manté tota la informació de processos i funcionament per a no perdre cap funcionalitat amb el canvi.

La metodologia a seguir divideix la realització de la nova ferramenta en sis fases diferenciades, representades a la figura 1:

1. **Fase de captació d'informació:** es documentaran tots els processos actuals de funcionament intern d'ICSA que siguen susceptibles de ser inclosos en l'ERP, tant els que ja estan implementats a la solució actual

com els que s'identifiquen nous. Amb aquesta documentació es prendrà la decisió de quins implementar i quins deixar fora.

2. **Fase de disseny:** Es realitzarà i documentarà el disseny tant intern com extern del nou aplicatiu, seguint les decisions preses a la fase anterior. En especial es definiran amb detall les primitives d'accés i modificació de dades, procediments i interfícies d'usuari.
3. **Fase de prova:** Es realitzaran maquetes el més detallades possible de les interfícies d'usuari, seguint els procediments documentats, i es presentaran a una selecció d'usuaris clau. Es documentarà les seues reaccions, temps d'aprenentatge i temps de finalització de les tasques proposades.
4. **Fase de redisseny i documentació:** Amb les dades aportades per la fase de prova, es faran les modificacions al disseny de les parts que no siguen satisfactòries amb els objectius d'utilitat i eficiència de l'aplicatiu. Es reunificarà tota la documentació realitzada en les fases anteriors i es planificarà la implementació del programari final, així com les fases de formació i posada en marxa.
5. **Fase d'implementació:** Es farà la implementació efectiva de l'aplicació, seguint la documentació de la fase anterior. Aquesta fase estarà formada per un seguit de sub-fases recurrents d'implementació i prova amb usuaris.
6. **Fase de posada en marxa:** Implementació dels plans de formació i posada en marxa de l'aplicatiu.

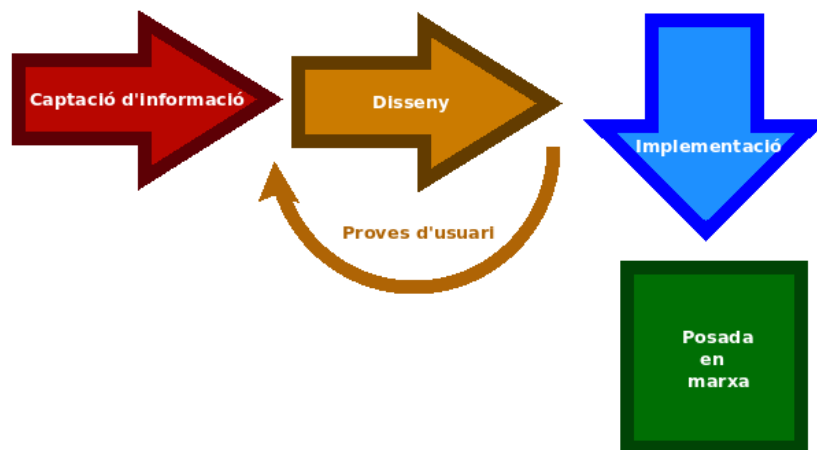


Figura 1: Metodologia de treball

Cal tindre en compte que, per als objectius d'aquest treball, només es completaran les fases 1 al 4. Les fases 5 i 6 només es definiran a aquesta memòria en la seua part teòrica, sense arribar a realitzar els treballs proposats, els quals es realitzaran després de la presentació de l'actual projecte.

1.4 Planificació del Treball

La planificació temporal del projecte i la inclusió en les PACs és la següent:

PAC2:

- Fase 1, captació d'informació: 1 de març a 20 de març.

PAC3:

- Fase 2, disseny: 21 de març a 19 d'abril.
- Fase 3, prova: 20 d'abril a 10 de maig.

PAC4:

- Fase 4, redisseny i documentació: 11 de maig a l'1 de juny.

A banda d'aquestes dades que són de planificació del treball de fi de grau, s'ha de tindre en compte les dades de finalització de cadascuna de les fases del nou LIMS de ICESA, ja que l'avanç d'aquest projecte depèn de la reutilització dels resultats del LIMS. Vist gràficament, la planificació és pot trobar a la figura 2, i el seu cronograma a la figura 3.

Fase	Procés	Data d'Inici	Duració	Data fi
Fase 1	Captació de processos originals	01/03/2021	7	08/03/2021
	Captació de processos nous	09/03/2021	7	16/03/2021
	Valoració processos	17/03/2021	3	20/03/2021
	Documentació de processos	05/03/2021	15	20/03/2021
Fase 2	Estructura base de dades	21/03/2021	9	30/03/2021
	Primitives d'accés i mod. dades	31/03/2021	9	09/04/2021
	Interfícies d'usuari	10/04/2021	9	19/04/2021
	Documentació disseny	26/03/2021	24	19/04/2021
Fase 3	Realització de maquetes	20/04/2021	10	30/04/2021
	Prova usuaris administradors	01/05/2021	4	05/05/2021
	Prova usuaris clau	06/05/2021	4	10/05/2021
	Documentació i conclusions	26/04/2021	14	10/05/2021
Fase 4	Redisseny	11/05/2021	5	16/05/2021
	Planificació implementació	17/05/2021	7	24/05/2021
	Planificació formació i arrancada	25/05/2021	7	01/06/2021
	Documentació i final del projecte	14/05/2021	19	02/06/2021

Figura 2: Fases del projecte

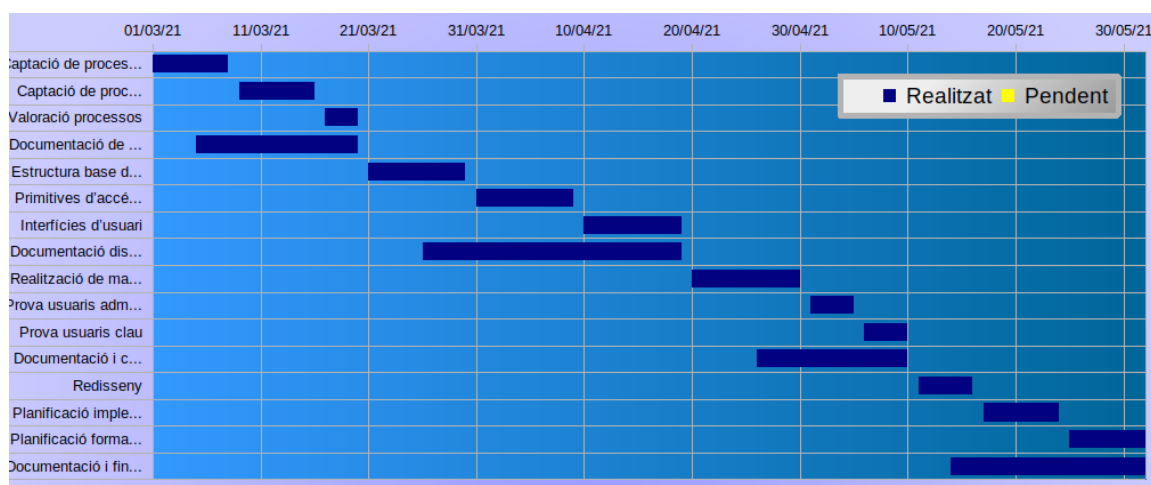


Figura 3: Cronograma del projecte

1.5 Breu sumari de productes obtinguts

Els productes obtinguts al final del projecte són:

1. Document de documentació de processos ICOSA.
2. Document de disseny del nou ERP.
3. Document de conclusions proves d'usuari i redisseny del nou ERP.
4. Document de planificació d'implementació, formació i posada en funcionament.
5. Memòria del projecte.

Cal tindre en compte que la documentació completa és propietat d'ICOSA, i per tant només es publicarà de forma oberta el seu resum a través d'aquesta memòria i la defensa d'aquesta.

1.6 Breu descripció dels altres capítols de la memòria

Capítol 1: Introducció (pàgina 1)

Busca donar una aproximació general dels objectius d'aquest projecte així com l'enfocament triat per a la seua evolució i la planificació que s'utilitzarà per a la seua realització, tant per part d'aquesta memòria com de la resta de processos previstos per a la futura implementació i posada en funcionament.

Capítol 2: Antecedents (pàgina 7)

Mostra la situació precedent i evolució d'aquesta dins de l'empresa a través de totes les solucions de gestió que s'han anat adaptant a través dels seus 37 anys d'història, fins arribar a la situació actual. Per a descriure els problemes de la situació actual, s'ha utilitzat el mètode d'entrevistes personals per definir-los. Per últim, es mostra les diverses alternatives proposades a l'empresa i amb tota aquesta informació es pren una decisió raonada sobre aquest projecte.

Capítol 3: Limitacions sobrevingudes al projecte (Pàgina 12)

Una explicació de com ha evolucionat el projecte durant la redacció d'aquesta memòria, i la justificació de la metodologia de desenvolupament utilitzada.

Capítol 4: Disseny (pàgina 14)

Es fa una descripció de cadascun dels components en que es proposa formular la nova aplicació, així com els mòduls que corresponen a cada component. En aquesta part de la memòria veurem la composició de les estructures de dades bàsiques, així com l'emmagatzemament de dades triat i la forma d'interconnectar això amb les interfícies d'usuari. També tenim a aquest capítol una descripció de les interfícies més importants a implementar i el seu funcionament de cara a l'usuari final.

Capítol 5: Proves amb usuaris clau (pàgina 39)

Per avaluar algunes de les interfícies i processos a modificar a la nova aplicació, i abans de fer la implementació, s'ha fet una sèrie de proves amb usuaris clau que els utilitzaran de forma extensiva i contínua, utilitzant maquetes i vis-tes parcials de les interfícies finals. En aquest capítol trobem la metodologia uti-

litzada, la descripció dels usuaris clau, les conclusions de les proves i les modificacions afegides al disseny gràcies a aquestes.

Capítol 6: Pla d'implementació (pàgina 45)

Es fa una proposta de pla d'implementació complet, amb els recursos necessaris tant tècnics com de personal, els temps assignats a cada recurs per a dur endavant la programació i implementació de l'aplicació i una planificació completa d'aquesta tenint en compte la limitació de recursos i les necessitats de funcionament de l'empresa.

Capítol 7: Posada en marxa (pàgina 49)

En aquest capítol es mostra el personal que serà necessari per a la posada en marxa de la nova aplicació així com els nous procediments a seguir, a través de un model de tres passos: formació → paral·lel → posada en real. També s'explica la planificació de cadascun d'aquests passos i la seua evolució.

Capítol 8: Conclusions del projecte (pàgina 53)

Descripció de la situació final esperada al final del projecte, continuació d'aquest en futur i consideracions finals a tindre en compte en el cas de realitzar altres projectes per part de l'empresa.

Capítol 9: Glossari (pàgina 57)

Definició d'alguns conceptes tècnics utilitzats a la memòria.

Capítol 10: Bibliografia (pàgina 60)

Bibliografia consultada per a redactar aquesta memòria i la resta del projecte.

2. Antecedents

2.1 Situacions precedents

La necessitat d'obtenir un ERP especialitzat per al funcionament d'ICSA existeix des de la seua fundació a 1984. ICSA va ser una empresa pionera en la utilització de la informàtica, i ha tingut com a mínim un treballador en plantilla per a fer desenvolupament específics en aquest tema pràcticament durant tota la seua història .



Figura 4: Evolució de la imatge corporativa de ICSA en 35 anys

Amb el canvi de segle, les necessitats logístiques van augmentar, i es va veure que calia fer un estudi personalitzat i intensiu de les necessitats d'aquell moment i les que després vindrien. Posteriorment a aquest estudi es va decidir contractar més personal programador per a dur endavant un projecte d'implementació d'un nou ERP per a l'empresa. A 2001 es va contractar un tercer programador que, conjuntament amb el personal original, va fer la majoria del desenvolupament amb Pascal i altres ferramentes de la època. Aquest nou desenvolupament va ser molt costós tant temporalment com en recursos materials, i no es va tindre una ferramenta útil fins a l'any 2008.

A 2009 es va incorporar a l'equip de programadors altra persona, degut a que per una banda el manteniment de la ferramenta original estava bloquejant a l'equip informàtic de portar correctament el dia a dia de la infraestructura de computadors de l'empresa, i per altra es volia monetitzar el desenvolupament realitzat els anys anteriors.

Després d'un estudi de mercat, es va decidir que no es podia entrar amb un ERP, però si que es podia utilitzar el coneixement extens de l'empresa en el món del laboratori i la consultoria en laboratoris agroalimentaris per a reconverir l'aplicació en un LIMS: una ferramenta de gestió de laboratoris. Amb aquesta meta es va treballar per l'equip de desenvolupament fins que es va arribar a la conclusió que tant l'arquitectura com les ferramentes de l'ERP no eren les adequades per a un LIMS.

Finalment el LIMS es va desenvolupar de forma independent i es va començar a comercialitzar l'any 2014. Actualment és utilitzat en un centenar de laboratoris tant agroalimentaris com químics i clínics tant a la península ibèrica com a altres països com Angola, o Qatar.

A l'any 2018 es va proposar utilitzar la base del LIMS per a refer el ERP de ICESA, però el projecte va acabar en fracàs: Encara que a 2019 es va posar en funcionament, es va abandonar als pocs mesos ja que tant la base de dades com la seua estructura no estava ben orientada a les necessitats de ICESA, i això causava greus problemes tant a secretaria com a logística de l'empresa. Des d'aquell moment es va tornar a l'ERP original posat en funcionament a 2008, amb totes les seues limitacions i deficiències adquirides amb el pas del temps.

2.2 Situació actual

Com hem vist al punt anterior, la situació actual és la utilització com a ERP d'una ferramenta feta a través d'un desenvolupament intern començat a 2001 i posat en funcionament a 2008, i que es pot veure a la figura 5. Encara que s'han anat fent modificacions per a tindre en compte l'evolució de l'empresa i de la legislació corresponent, moltes d'aquestes estan basades en pegats fora de l'estructura original. Per altra banda, moltes de les tàctiques i tecnologies modernes aplicables a la gestió d'empresa no són possibles d'explotar a través de la ferramenta actual.

Id. Intención	Empleado	Persona Juridica	Total Factura	Fecha de creación	Total Base	Contrato
4 200000	4650008	SERVICIO SALUD CASTILLA ...	-1.024,87	13/10/2020 12:35:59	-847,00	9 200048
2 200206	4650008	SERVICIO SALUD CASTILLA ...	847,00	13/10/2020 12:55:26	700,00	9 200050
2 200207	4650002	CIRUGIA OCULAR DE MADRI...	421,81	19/10/2020 9:43:37	348,60	9 200056
2 200208	4650004	EMPRESA DE TRANSFORMA...	605,00	20/10/2020 13:33:53	500,00	9 200101
2 200209	4650004	MICROGAIA BIOTECH, S.L.	816,75	20/10/2020 13:54:31	675,00	9 200102
4 200001	4650025	SERVICIO MURCIANO DE SA...	-3.042,67	21/10/2020 14:07:42	-2.514,60	9 200111
2 200210	4650025	SERVICIO MURCIANO DE SA...	3.183,12	21/10/2020 14:12:44	2.630,68	9 200111
4 200002	4650025	SERVICIO MURCIANO DE SA...	-1.173,34	21/10/2020 16:33:44	-969,70	9 200113
2 200211	4650025	SERVICIO MURCIANO DE SA...	1.169,71	21/10/2020 16:39:01	966,70	9 200113
2 200212	4650003	AGRUPACION MEDICA BALE...	1.394,08	23/10/2020 12:05:29	1.152,13	9 192141
2 200213	4650025	AKRALAB S.L.	703,78	23/10/2020 12:07:16	581,64	9 192083
2 200214	4650025	ALTHAIA, Xanxa Assistencial...	933,39	23/10/2020 12:09:40	771,40	9 200024
2 200215	4650025	ALTHAIA, Xanxa Assistencial...	511,83	23/10/2020 12:28:59	423,00	9 200025
2 200216	4650002	AMADOR GARCIA SANCHO	385,51	23/10/2020 12:38:40	318,60	9 200021
2 200217	4650025	CENTRO ANAT PATOLÓGICA...	137,75	23/10/2020 12:40:49	113,84	9 200112
2 200218	4650025	CENTRO ANAT PATOLÓGICA...	235,22	23/10/2020 12:44:06	194,40	9 200052
2 200219	4650025	CENTRO ANAT PATOLÓGICA...	356,79	23/10/2020 12:45:50	296,52	9 192164
2 200220	4650003	CLINICA DERMATOLÓGICA D...	484,00	23/10/2020 12:49:21	400,00	9 192153
2 200221	4650025	CONSORCI SANITARI DEL M...	2.308,68	23/10/2020 12:53:37	1.908,00	9 200030
2 200222	4650025	CONSORCI SANITARI DEL M...	2.308,68	23/10/2020 12:55:46	1.908,00	9 200031
2 200223	4650025	CONSORCI SANITARI DEL M...	1.099,89	23/10/2020 12:57:19	909,00	9 200032
2 200224	4650003	CORPORACIO SANITARIA P...	368,03	23/10/2020 13:01:04	304,16	9 192153
2 200225	4650003	CORPORACIO SANITARIA P...	87,12	23/10/2020 13:02:59	72,00	9 200010
2 200226	4650003	CORPORACIO SANITARIA P...	320,17	23/10/2020 13:03:58	264,60	9 200071
2 200227	4650003	CORPORACIO SANITARIA P...	151,01	23/10/2020 13:05:21	124,80	9 200094
2 200228	4650003	Centro Diagnostico Calderón...	283,14	23/10/2020 13:07:31	234,00	9 200003
2 200229	4650003	Centro Diagnostico Calderón...	366,44	23/10/2020 13:09:34	302,84	9 200084
2 200230	4650003	Clinica Medicina Integral Dia...	389,93	23/10/2020 13:12:09	322,26	9 200017
2 200231	4650003	Consoenci Sanitari Integral	794,64	23/10/2020 13:13:53	654,73	9 192157

Figura 5: Captura de l'ERP que s'utilitza actualment a l'empresa

Per a conèixer els problemes irresolubles actuals, s'ha entrevistat als caps dels tres departaments més implicats en la utilització de l'ERP: Per administració que gestiona facturació i pagaments Inma Garcia; per logística amb compres, estoc i enviaments German Balsano; per Comercial amb pressupostos i albarans Jose Carlos Herrero i per últim a direcció amb Lola Franco. Aquest procediment d'identificació de problemes s'ha fet seguint les indicacions de Grant Norris, James R. Hurley, Kenneth M. Hartley et al [2].

Després d'aquestes entrevistes, els problemes més importants amb els que tracten i han de resoldre per a millorar la seua eficiència són:

Administració (Inma García):

- Lentitud i falta de flexibilitat en la generació de factures.
- Complicació en creuar dades, en especial entre factures i pressupostos.
- Impossibilitat de gestionar tresoreria i pagaments de factures.
- Complicacions amb la integració de la ferramenta de comptabilitat.
- Falta d'indicadors útils com situació d'impagaments o factures vençudes.
- Complicació en la visualització de fases de pagament o pagaments previstos.

Logística (German Balsano):

- Bloquejos constants quan s'han de donar d'alta una gran quantitat d'elements individuals.
- Complicació de creuar dades de compres amb estoc i magatzem.
- Impossibilitat de bloqueig d'estoc per a vendes concretes.
- Falta de ferramentes per establir estocs mínims i desitjables.

Comercial (Jose Carlos Herrero):

- Falta d'integració entre l'ERP (contractes i factures) i el CRM (pressupostos i propostes).
- Complicació per extreure dades de facturació per departaments o persones que han realitzat la venda.

Direcció (Lola García):

- Total falta d'indicadors dins de l'aplicació.
- Complicació per extreure dades (a través de consultes directes de la base de dades) per estudiar-les en aplicacions externes.
- Falta d'utilitats de Business Intelligence (BI) per extreure tendències o reutilitzar dades d'anys anteriors.
- Incapacitat d'encreuament de dades dins de l'aplicació.

2.3 Alternatives proposades

Les alternatives plantejades a realitzar un desenvolupament intern nou han sigut: continuar amb la ferramenta actual com està, fer un esforç d'actualització de la ferramenta actual i adquirir d'un proveïdor un ERP comercial. Aquestes han sigut les conclusions de cadascuna de les opcions, després de la seua discussió amb personal afectat:

- **Continuar** ERP actual sense modificacions: des de Direcció de l'empresa s'ha exigit realitzar un canvi, de manera que aquesta opció queda descartada.
- **Actualització** de l'ERP actual: Aquest està programat en tecnologies obsoletes i un llenguatge de programació que la meitat de l'equip actual de desenvolupament no domina. A banda, les necessitats estudiades no són compatibles amb les estructures de dades i emmagatzemament

actual, de manera que caldria fer una modificació extensiva de tota l'aplicació.

- **Adquisició** d'ERP comercial: des del començament és l'opció favorita de l'equip informàtic de l'empresa. S'han estudiat les següents alternatives comercials.
 - Sage 200¹ (abans Sage Murano): Ferramenta en núvol de gestió completa empresarial. Coneguda per l'equip de desenvolupament de l'empresa des de fa anys per haver integrat a alguns clients el seu LIMS amb Sage. Una opció molt completa i moderna. El cost de la implementació equivalent a la proposada en aquesta memòria s'ha valorat en 60.000€.
 - SAP ERP²: L'opció més completa i flexible de les alternatives revisades. Es tracta d'una ferramenta àmpliament utilitzada en qualsevol tipus d'empresa, i que té mòduls per a fer pràcticament de tot. És la solució més cara de les estudiades, amb un cost equivalent d'uns 100.000€.
 - Oracle NetSuite:³ Una altra solució de ERP en núvol, menys flexible que les anteriors, però també més barata. La implementació equivalent serien uns 40.000€.
 - Odoo ERP⁴: Es tracta d'un ERP comercial open source. La implementació específica la feria una empresa externa, però al ser *open source* ens permetrà continuar-la internament en el futur. El cost de realitzar una implementació equivalent està valorat en uns 30.000€.

Totes les valoracions econòmiques de les alternatives s'han fet a través d'una entrevista amb comercials d'empreses que ofereixen aquests productes, i responen a fer una implementació completa de les necessitats de l'empresa.

2.4 Càlcul del cost del projecte

Per avaluar les opcions descrites i poder comparar amb l'adquisició d'altres ERP comercials, s'ha fet un càlcul aproximat del cost d'oportunitat que correspondrà al projecte.

Per fer això, s'ha avaluat de forma inicial el temps de dedicació necessari en programadors/hora de cadascun dels components proposats (i explicats en el capítol 4. Disseny de l'aplicació), i això s'ha multiplicat pel cost d'oportunitat d'un treballador/dia definit per direcció de l'empresa, que correspon a 400€.

Cal tindre en compte que aquest **cost d'oportunitat** no correspon al cost real que tindrà el desenvolupament per a l'empresa, sinó el màxim que aquesta podria deixar d'ingressar per tindre a treballadors dedicats a aquest projecte i no a altres que es podrien monetitzar directament. Tampoc compta amb cap cost de formació necessari o costos indirectes produïts per les tecnologies utilitzades.

1 <https://www.sage.com/es-es/productos/sage-200cloud/>

2 <https://www.sap.com/products/enterprise-management-erp.html>

3 <https://www.netsuite.com/>

4 <https://www.odoo.com/>

	Hores/persona
Preparació	60
BD	120
Servidor	160
Client	180
Implementació	140
Posada en marxa	45
Dedicació	705
Cost d'oportunitat total	47.000,00 €

Figura 6: Cost d'oportunitat de la proposta

Els càlculs totals del cost d'oportunitat es pot trobar a la figura 6, amb el total calculat de hores/persona de 705 hores, que amb un aprofitament de 6/hores dia previst i el cost de persona/dia definit per ICESA donaria un cost aproximat de 47.000€.

2.5 Presa de decisió

Amb les dades definides, es presenta a Direcció de l'empresa i als departaments implicats el resum de les opcions amb el cost, tant monetari com competitiu, de cada solució per a prendre la decisió final. Tant Direcció com els departaments implicats s'alineen amb l'opció de **realitzar un desenvolupament intern**, en contra de l'opinió del departament d'informàtica que proposa la compra d'un ERP comercial.

Finalment es pren la decisió de portar cap endavant aquest projecte de ERP desenvolupat internament a ICESA, amb la condició d'intentar fer alguns dels components reutilitzables per a futurs desenvolupaments interns com per exemple el nou LIMS, i la monetització d'aquests.

3. Limitacions sobrevingudes al projecte

Abans d'entrar a detallar el disseny de l'aplicació, es necessari fer una sèrie de puntualitzacions sobre els objectius finals d'aquest projecte i la seua evolució durant el temps de redacció d'aquesta memòria.

En un inici es va proposar com un disseny complet d'una ferramenta ERP per a l'ús intern d'ICSA, seguint un paradigma de desenvolupament en cascada, amb fases molt definides i allargades en el temps. Aquest és el millor model per a fer aplicacions tancades i sobre les quals es té un coneixement ampli, com és el cas de l'ERP que ha de ser utilitzat a l'empresa.

Quan es va detectar la magnitud del projecte a la fase de disseny, es va proposar a direcció de ICSA de forma completa per aconseguir el vist-i-plau i l'assignació corresponent de recursos. En eixe moment es va veure que aquesta aproximació no només no era la més encertada sinó que era simplement impossible. La situació actual d'alguns dels departaments de ICSA no permeten que es mantinga l'ERP actual durant anys fins que el desenvolupament estiga acabat, i la disponibilitat de recursos tampoc permet llançar projectes a molt llarg termini.

A banda tampoc és possible, per les càrregues de treball actual dels treballadors, fer fases de posada en marxa llargues, ni maratons per a documentar tots els processos utilitzats dins de l'empresa. A això s'afegeix la futura jubilació d'una de les persones amb més coneixement del departament d'administració, el qual cal assimilar en un curt període de temps.

Desenvolupament en cascada



Desenvolupament àgil

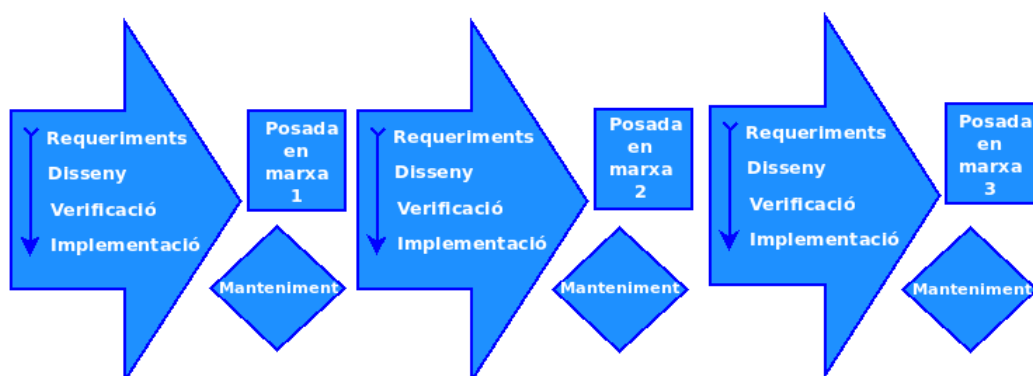


Figura 7: Desenvolupament en cascada i desenvolupament àgil

És per això que es va decidir que la millor solució per a resoldre aquest problema és canviar el paradigma de desenvolupament per un àgil, tal com defineix Robert C. Martin [3] i apareix resumit a la figura 7, i que permet fer el projecte per fases on a la finalització de cada fase obtenim un producte acabat que ja es puga utilitzar i formar als usuaris per a la seua utilització.

En resum: durant el temps de redactat d'aquesta memòria han canviat tant els recursos, com els objectius i la metodologia de desenvolupament del projecte, i per tant el contingut del document s'ha modificat per a que reflectisca la situació i funcionament del projecte a juny de 2021.

4. Disseny de l'aplicació

4.1 Consideracions prèvies

Tal com hem vist al l'apartat 1.3 Enfocament i mètode seguit i a 1.2 Objectius del Treball, es busca obtenir una aplicació completament funcional per a les necessitats actuals i futures en el context ERP de la empresa, així com reutilitzar els coneixements actuals de l'equip de treball i obtenir components que puguin ser reutilitzats per a futures comercialitzacions.

Així mateix, a les proves prèvies i antecedents exposats al capítol 2. Antecedents, vam trobar la complicació de mantindre grans quantitats de dades coherents i poder gestionar-les en temps real, sense tindre recursos poc aprofitats o sobredimensionar els recursos necessaris per a poder mantindre l'eficiència en els casos de major demanda.

També, tal com s'ha vist en els antecedents a aquest projecte, la migració d'aplicacions per motiu del manteniment de les ferramentes utilitzades és una despesa important que cal tindre en compte, i per tant utilitzar al màxim possible ferramentes gestionades externament per a no augmentar el cost de manteniment de sistemes.

Per això, i seguint aquesta dinàmica, exposem les següents necessitats:

- Utilitzar de forma general els **coneixements actuals** de l'equip de treball.
- Utilitzar **tecnologies modernes**, actualitzades a la situació actual, i que puguin ser fàcilment mantingudes durant un mínim de 10 anys.
- Els components han de ser **reutilitzables** al màxim.
- **Externalitzar** al màxim la gestió d'elements comuns, a través de servicis IaaS i PaaS per a minimitzar la infraestructura interna.

4.1.1 Plataformes i llenguatges

La primera decisió de disseny és establir la plataforma destí del projecte. Per les necessitats de reutilització i accés, l'única alternativa viable és que la nova aplicació se execute baix el paradigma de la computació en núvol, utilitzant com a client l'explorador web de l'usuari.

Des del punt de vista de programació, també cal decidir els llenguatges que es van a utilitzar. En aquest punt és prou clar que tot ha de ser utilitzant els coneixuts i utilitzats per l'equip de treball actual, que serà qui implemente finalment aquest disseny, i que correspon a Python per a la part en back-end, i Javascript per als front-end. En especial, s'ha triat Python 3.9 pel seu manteniment allargat i ECMAScript 2016 pel seu suport universal a tots els exploradors web moderns (David Flanagan, [6]).

4.1.2 Framework

El següent pas és definir els *frameworks* que s'utilitzaran. En aquest cas si que s'ha donat un poc més de llibertat que no amb els llenguatges, donada la situació de canvi de tecnologia obligada per la renovació de model. Els últims 10 anys s'ha utilitzat a ICSA com a *framework* bàsic per a tots els projectes de desenvolupament Django 1.4, amb Python 2; aquesta versió de Django ha quedat en els últims anys sense suport, ja que el manteniment de Python 2 es va perdre en gener de 2020. L'actualització necessària a Django 3 (basat en Python 3) ens dona la oportunitat per a re-estudiar les necessitats del projecte i avaluar altres opcions.

En especial, es van avaluar les següents tres alternatives com a *framework* per a l'actual projecte:

Django 3⁵: L'alternativa obvia, degut als desenvolupaments anteriors. Es tracta d'un *framework* molt utilitzat que, entre altres coses facilita l'accés a les dades a través d'un sistema integrat. Dels *frameworks* web per a python és probablement és més complet. Com a desavantatge, la seua complexitat el fa poc flexible, i en alguns cassos és molt pesat per a serveis lleugers.

Django proporciona el seu propi ORM per a gestionar estructures de dades directament emmagatzemades en disc, a través de diverses opcions per a connectar bases de dades, així com un sistema complet per a gestionar models, que facilita el desenvolupament d'entorns web. També permet realitzar accions basades en esdeveniments diferents als accessos.

Flask⁶: També un *framework* que facilita els desenvolupaments basats en web, però a diferència de Django es completament modular i minimalista. Permet la generació de vistes i connexió a bases de dades, però no al nivell que permet Django.

Per altra banda, el seu minimalisme i simplicitat permet crear aplicacions més complexes i diferents sense tindre que treballar contra limitacions arbitràries o APIs sense massa documentació (Miguel Grinberg, [4])

Com a principal contra, Flask no permet de forma directa realitzar programació asíncrona o basada en esdeveniments.

Web2py⁷: L'últim *framework* python valorat està basat en Django i Ruby on Rails (RoR), i el seu objectiu és ser àmpliament portable entre plataformes. Encara que té els beneficis ja vists a Flask, sense perdre moltes de les funcionalitats de Django, actualment la seua comunitat i documentació és molt reduïda i per això es descarta per aquest desenvolupament.

5 <https://www.djangoproject.com/>

6 <https://flask.palletsprojects.com/>

7 <http://www.web2py.com/>

Amb totes aquestes consideracions, finalment es decideix utilitzar per al projecte com a framework Python Flask, ja que la necessitat de flexibilitat per a poder implementar una estructura de dades especialitzada, amb una base de dades no estàndard i la capacitat de modularitat i separació entre components ha pesat més que la facilitat de codificació.

Amb Flask podrem implementar cada component i mòdul per separat, així com totes les APIs i integracions necessàries per a que totes les parts puguin comunicar-se entre elles i intercanviar dades (Miguel Grinberg, [4]).

4.1.3 Desplegament i contenidors

Com hem vist anteriorment, un dels objectius és intentar reduir al màxim els costos tant en infraestructura com en recursos no utilitzats. A banda, l'aplicació final ha de ser capaç d'escalar d'una forma el més automàtica possible per assumir pics de demanda i reduir recursos en els temps de parada, cosa que es pot aconseguir a través de la tecnologia de contenidors per al desplegament. Les alternatives estudiades han sigut:

Kubernetes⁸: Es tracta d'un sistema que permet la automatització del desplegament, ajust d'escala i control d'aplicacions en contenidors. En un origen va ser dissenyat i implementat per Google, i actualment es tracta de programari lliure de codi obert.

Kubernetes defineix una sèrie de primitives que, conjuntament, permeten el desplegament, manteniment i escalat de l'aplicació continguda. Aquest components són completament extensibles gràcies a la API de kubernetes, que s'utilitza directament pels components interns així com les extensions i contenidors executats. A través de l'API, cada component de l'aplicació en curs es converteix en un pod, que funciona de forma independent a la resta de l'aplicació.

De forma general, la utilització de kubernetes facilitarà tres dels objectius d'aquest projecte: la modularitat, la capacitat d'escalat i el fàcil desplegament i moviment de l'aplicació entre servidors tenint en compte la demanda de recursos necessaris en cada moment.

AWS Fargate⁹: No és realment una alternativa a kubernetes, sinó una addició a aquestes per a facilitar la seua gestió. AWS Fargate permet implantar i desplegar aplicacions kubernetes sense la necessitat d'aprovisionar i administrar cap tipus de servidor físic ni virtual. Per cada aplicació desplegada a Fargate, aquest assigna els recursos i servidors físics necessaris depenent de la càrrega actual i esperada en cada moment.

A banda, la capacitat de Fargate d'executar cada pod de kubernetes en un espai de nucli separat dóna seguretat i estabilitat a tot el sistema.

8 <https://kubernetes.io/>

9 <https://aws.amazon.com/fargate/>

AWS Fargate facilita en bona mesura l'escalabilitat de l'aplicació, i redueix la quantitat de recursos necessaris en moments en que aquesta no els necessita, un dels requeriments bàsics d'aquest projecte.

Docker¹⁰: Es tracta d'una tecnologia que permet gestionar aplicacions en contenidors, com per exemple els utilitzats en kubernetes, com si foren màquines virtuals completament independent entre elles. Gràcies a això es guanya en modularitat, còpies de seguretat i control de versions per la seua capacitat de fer imatges de un moment concret de cada contenidor i restaurar-les i facilita la implementació ràpida ja que els desplegaments de prova són pràcticament automàtics.

Docker seria una alternativa a l'ús de servicis com AWS Fargate per a gestionar els contenidors de forma directa en servidors reals o màquines virtuals. Els beneficis són el control total sobre la plataforma i la no dependència de cap empresa o programari complet, ja que és lliure i independent. Com a contres tenim per una banda que es complica la gestió i manteniment de servidors, i l'escalat i ús de recursos passa a ser completament manual. A la figura 8 tenim l'evolució de la metodologia de desplegaments¹¹.

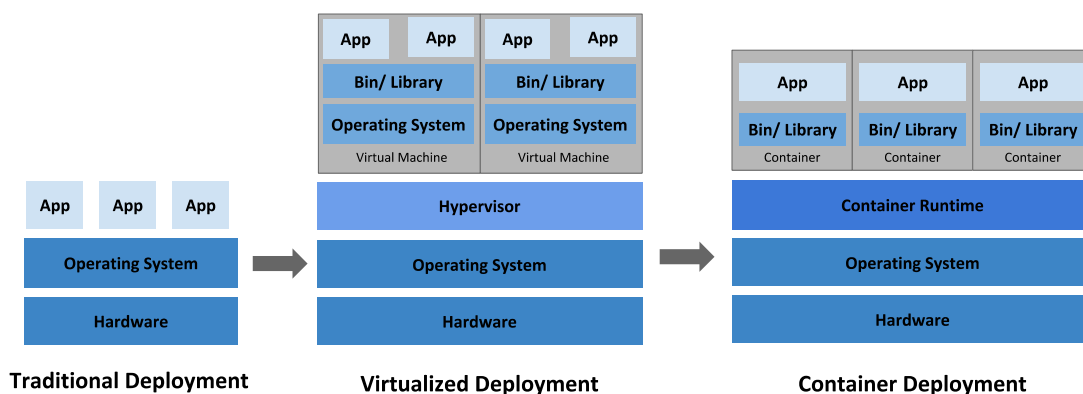


Figura 8: Evolució de la metodologia de desplegament, fins a contenidors.

Considerant les alternatives estudiades, es decideix desplegar contenidors amb kubernetes sobre la plataforma AWS Fargate ja que és la solució que més ens apropa als objectius originals de l'aplicació.

Amb aquesta solució no només aconseguim reduir de forma important la gestió de servidors, sinó també facilitar l'escalat futur de l'aplicació i reduir de forma considerable els costos en recursos no utilitzats en moments de baixa demanda. A banda, augmenta de forma important la possible reutilització de alguns dels components del sistema.

10 <https://www.docker.com/>

11 Imatge procedent de kubernetes.io, baix llicència CC BY 4.0.

4.1.4 SGBD

Un altre punt important a decidir és el sistema de gestió de bases de dades. Com hem vist anteriorment la flexibilitat és una de les característiques principals del projecte, tant per la necessitat de la reutilització del codi com per les pròpies necessitats de l'aplicació final.

Per això, s'ha estudiat utilitzar els següents tres paradigmes que podrien utilitzar com a back-end de les dades de l'aplicació:

- **SQL** (dades fortament estructurades): Es tracta del paradigma més utilitzat i que permet mantindre una coherència forta, així com referències directes entre els elements i capacitat de executar accions directes en el SGBD. Molt eficient si s'utilitzen els recursos necessaris.
- **Base de dades orientada a objectes**: Amb una coherència més baixa que els sistemes SQL, però basat en el concepte d'herència. Permet més flexibilitat sense perdre del tot la rigidesa de sistema, a costa de pitjor eficiència a l'accés de dades.
- **Base de dades orientada a documents**: No existeix cap coherència de dades. S'obté la màxima flexibilitat i també màxima eficiència a costa de perdre referències i encreuament de dades.

A la figura 9 tenim una taula de pros i contres de treballar amb bases de dades relacionals i no relacionals (Ernest Abadal Falgueras, Lluís Codina Bonilla, [5]).

Per les peculiaritats del projecte comentades anteriorment, i experiències anteriors exposades al capítol 2. Antecedents, s'escolleix treballar amb bases de dades orientades a documents, i en especial les que treballen amb documents JSON.

	Pros	Contres
Bases de dades relacionals (SQL)	<ul style="list-style-type: none"> • Assegura la integritat de les dades • Esquema normalitzat • Canvis en cascada • Llenguatge universal de consultes • Àmpliament utilitzat i molt eficient 	<ul style="list-style-type: none"> • No és flexible • No treballa bé amb dades diferents al text • No es pot manipular el contingut de les entrades de forma individual
Bases de dades no relacionals (noSQL)	<ul style="list-style-type: none"> • Flexibilitat total en l'estructura de dades • Consultes completament configurables • Tractament individualitzat d'atributs • Escalat trivial per a futures ampliacions de funcionament 	<ul style="list-style-type: none"> • Sense un estàndard d'utilització, cada SGBD funciona de forma diferent. • Menys provat i menys desenvolupaments que SQL. • La integritat de les dades depèn completament de la programació. • No compta amb gestió d'esdeveniments ni canvis en temps real.

Figura 9: Pros i contres de la utilització de bases de dades estructurades i no relacionals

Dins d'aquesta categoria, s'han estudiat els següents sistemes de bases de dades:

MongoDB¹²: Es tracta d'un programari lliure de gestió de bases de dades orientades a documents. Utilitza el seu propi format anomenat BSON que és una ampliació de JSON original per a introduir de forma fàcil dades binàries. MongoDB és una plataforma àmpliament provada en molts projectes, eficaç i sòlida. Com hem vist en l'apartat d'antecedents, l'equip de desenvolupament de ICSA té àmplia experiència utilitzant MongoDB ja que les memòries cau de dades del sistema Nevis estan implementades utilitzant-lo.

MongoDB es pot implantar amb gestió total o directa, o utilitzar serveis com per exemple MongoDB Atlas, que són servidors ja gestionats externament als núvols públics d'Amazon, Google o Microsoft i permeten abstraure tota la infraestructura necessària per a utilitzar MongoDB sense la necessitat de gestionar-lo o mantenir-lo.

Amazon DocumentDB¹³: És una implementació alternativa del format de MongoDB proposada i comercialitzada per Amazon, majorment compatible amb totes les ferramentes de MongoDB. Aquesta solució només es pot utilitzar de forma completament gestionada per Amazon, de manera que abstrau l'accés de dades de la gestió i manteniment de servidors.

Encara que és una solució propietària, la capacitat de ser compatible amb MongoDB permet en un futur la fàcil migració entre els dos servicis, i promet escalar i tindre un rendiment superior que el de MongoDB.

Apache CouchDB¹⁴: Una altra solució no-SQL de base de dades orientada a documents. Utilitza JSON per a emmagatzemar les dades de forma pareguda a les solucions anteriors, i té darrere una gran quantitat de projectes grans assegurant la seua solidesa i futur.

La manipulació de dades a CouchDB es fa a base de cridades HTTP, que concorda complement amb la idea de comunicació de l'aplicació ERP proposada. Encara que el seu rendiment siga inferior a les aplicacions anteriorment estudiades, cal tindre-ho en compte per la seua fàcil inclusió en l'arquitectura dissenyada i la seua facilitat d'ampliació en un futur.

Per a prendre una decisió es tindrà en compte el rendiment en cas d'estrès de cadascuna de les opcions proposades, així com la proximitat de cada solució als objectius del projecte i l'aplicació final.

4.1.4.1 Proves de rendiment

Es realitzen proves de rendiment dels tres motors de base de dades proposats. Per la seues característiques les proves no es poden fer en les mateixes condi-

12 <https://www.mongodb.com/>

13 <https://aws.amazon.com/documentdb/>

14 <https://couchdb.apache.org/>

cions, ja que MongoDB i CouchDB són configurades per l'usuari i en canvi DocumentDB és un servici completament gestionat.

Per a intentar fer les proves de rendiment de la forma més agnòstica possible a qualsevol de les opcions, s'han fet sobre màquines virtuals equivalents a les oferides per la solució de Amazon DocumentDB, en especial de les anomenades *t3.medium* a Amazon EC2 que donen 2 nuclis de computació en paral·lel, 4GiB de memòria i 5Gb de connexió amb la xarxa.

Les proves han sigut 3 diferents: inserció de dades, modificació de dades i consulta de dades. Un script genera l'estructura corresponent i equivalent per a cada SGBD, de manera que les dades introduïdes són equivalents entre els tres sistemes. Aquest programa de prova, escrit en python, utilitza les llibreries natives de cada plataforma per connectar i fer les peticions/consultes, seguint les equivalències en cada cas. Cadascuna de les proves es compten com el temps en segons de fer 1.000 vegades cada acció aleatoritzada, i s'executa des d'una màquina virtual en la mateixa zona del núvol de Amazon on estan les bases de dades.

Per a la inserció de dades, s'ha preparat un JSON equivalent a una plantilla de nevis d'un resultat model, amb els valors de cadascun dels seus atributs aleatoritzats; en el cas dels atributs de tipus data, es generen de forma automàtica sobre els possibles del calendari de 2021.

Per a la modificació de dades, es modifica aleatòriament un dels atributs de qualsevol dels 1.000 elements introduïts anteriorment.

Per a la consulta de dades, es generen dos dates aleatòries corresponents a l'any 2021 i s'executa la cerca d'entre eixes dos dates d'un atribut aleatori dels de data de l'estructura generada. Com la resta de proves, es repeteix 1.000 vegades.

		Inserció x 1000 (s)	Modificació x 1000 (s)	Consulta x 1000 (s)
MongoDB	P1	34	38	121
	P2	32	34	122
	P3	33	34	118
	MITJANA	33	35	120
DocumentDB	P1	36	37	118
	P2	31	30	151
	P3	35	31	122
	MITJANA	34	33	130
CouchDB	P1	101	98	998
	P2	98	95	1002
	P3	94	98	997
	MITJANA	98	97	999

Figura 10: Resultats en segons de les proves d'SGBD

Els resultats de les proves, en segons, es troben a la figura 10, on tant MongoDB com DocumentDB tenen resultats molt pareguts, i CouchDB queda un ordre de magnitud darrere, en especial en les cerques de dades.

4.1.4.2 Selecció SGDB

Amb aquestes dades a la mà, i tota la resta d'informació, es decideix apostar per utilitzar com a SGDB per al projecte la solució d'Amazon DocumentDB, que per una banda dona un rendiment equivalent al més eficient dels provats, i per l'altra facilita la posada en funcionament gràcies a la seua aproximació com a DBaaS, completament gestionat de forma externa. Això facilita també la generació de còpies de seguretat i la resposta i recuperació en cas de desastre.

4.2 Disseny general

Com hem vist a la secció 4.1 Consideracions prèvies, l'objectiu és obtenir una ferramenta útil i alguns dels seus components siguen completament reutilitzables. Per aconseguir això, s'ha dissenyat una estructura basada en 3 components diferenciats:

- **Dades**
- **Servidor**
- **Client**

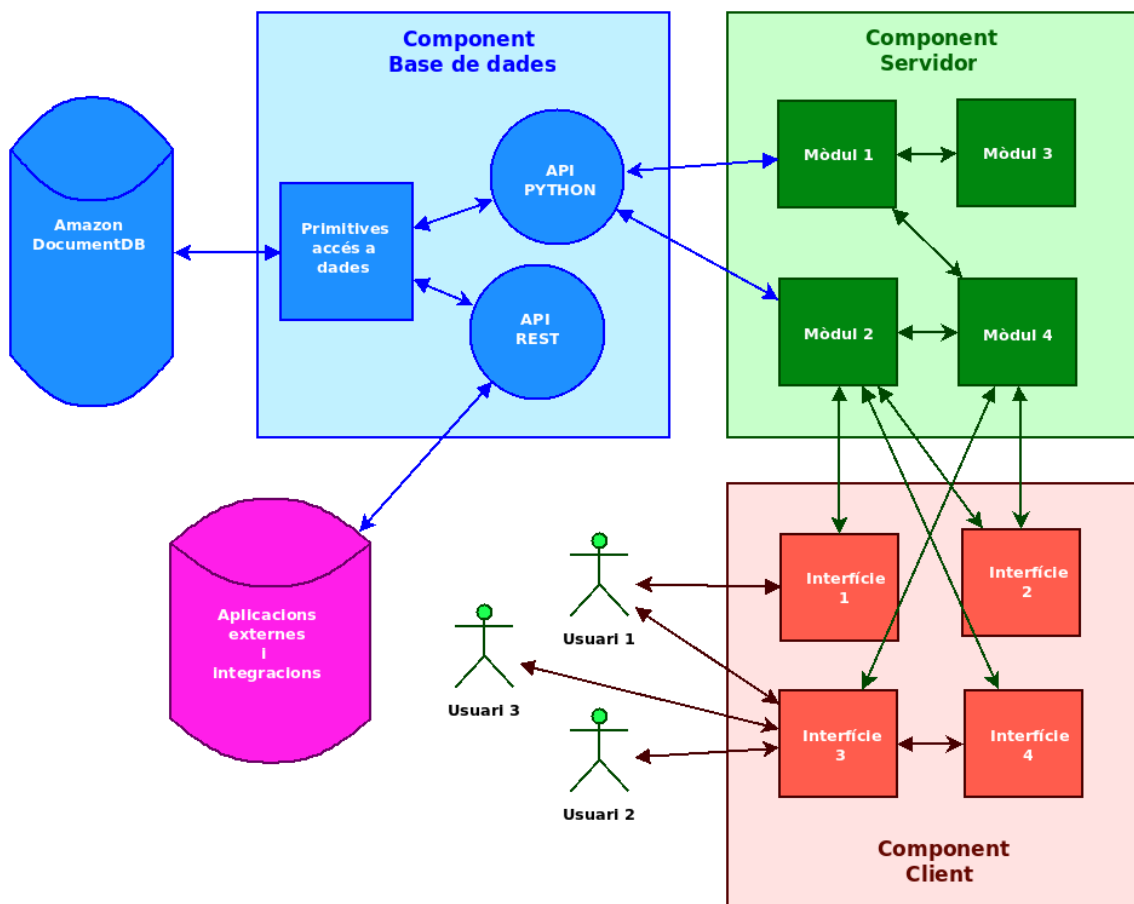


Figura 11: Interconnexions dels components

Cadascú d'aquests components és una aplicació en si mateix, que encara que mantinguem interdependències necessàries (el servidor no funcionarà sense accés a dades, i el client no funcionarà sense accés al servidor) es dissenyaran i programaran de forma diferenciada. Aquestes interconnexions estan definides

a la figura 11. En especial, les característiques bàsiques de cada component són:

Dades: Correspon a l'estructura bàsica, primitives de accés a dades i servicis necessaris per a utilitzar aquestes primitives i fer-les públiques per a la utilització a la resta de components. En especial, per aquest projecte es faran públiques aquestes primitives a través de dos formes diferents:

- Un API Python intern, utilitzable com un mòdul a carregar.
- Un API REST accessible a través d'Internet, per a facilitar les integracions amb altres aplicacions.

L'accés a dades anirà limitat a través del sistema d'usuaris i grups propi de l'accés a dades.

Servidor: Part interna de l'aplicació, que té els *backends* necessaris per a realitzar la gran majoria d'accions necessàries. Aquesta part serà implementada amb Python i Flask, com a contenidor per a kubernetes. Cada mòdul de la part del servidor correspondrà a un micro-servici dins del contenidor que servirà una sèrie d'URLs que corresponen a totes les necessitats d'accés i processament de dades del mòdul. Accedirà a les dades a través de l'API Python del component de dades.

L'accés a dades anirà limitat per un sistema d'usuaris i grups, que no té perquè coincidir amb el d'accés a dades.

Client: Implementarà totes les interfícies d'usuari necessàries, i utilitzarà cridades REST cap a les URL del component servidor per a les consultes o modificacions d'aquestes. Està compost per un micro-servici flask que pot anar dins del mateix contenidor del component servidor, o altre diferenciat.

L'accés a dades serà utilitzant el mateix sistema d'usuaris i grups del component servidor. Estarà íntegrament programat en html+css+javascript.

4.3 Disseny del component dades

4.3.1 Estructura de la base de dades

Com ja hem vist anteriorment, la base de dades serà orientada a documents. Cada ítem amb entitat pròpia dins del sistema, a partir d'ara «element», estarà format per una sèrie d'etiquetes anomenades «atributs», dins d'un JSON.

Cada tipus d'element tindrà la seua col·lecció separada dins de la base de dades, de manera que dos elements de dos tipus no estiguen mai en la mateixa col·lecció.

El document amb ID 0 de cada col·lecció no correspondrà a cap element, sinó a informació general del tipus d'element.

L'estructura bàsica d'un element és la següent:

```
{
  "type": "",
  "attr": [],
  "labels": [],
  "timestamp": "",
  "id": 1,
  "eid": "",
}
```

JSON 1: Element

Amb el següent contingut:

- type: Tipus d'element que correspon el document.
- attr: Llista de atributs que conté l'element. El format està explicat més endavant.
- labels: Llista d'etiquetes en les que està l'element. El format està explicat més endavant.
- timestamp: Data i hora de creació de l'element, en format ISO 8601.
- id: Identificador auto-numèric de l'element.
- eid: Cadena de caràcters que correspon a un identificador únic per a tot el sistema de l'element. Mai pot haver dos elements en el sistema amb el mateix uid. Conté el tipus d'element, el seu id i el seu timestamp de generació.

Per la seua banda, cada ítem de atribut serà un diccionari amb l'estructura bàsica següent:

```
{
  "name": "",
  "value": "" ó [],
  "format": {},
  "audit": [],
  "timestamp": "",
  "user": ""
}
```

JSON 2: Atribut

Amb el següent contingut:

- name: cadena de caràcters amb el nom estàndard de l'element.
- value: cadena de caràcters, o llista en el cas d'enllaç, amb el contingut de l'atribut.

- format: diccionari amb informació del format de l'atribut. El contingut està explicat més endavant.
- audit: llista de ítems atributs, amb tots els valors que ha tingut l'atribut, inclòs l'actual, en forma d'auditoria de canvis.
- timestamp: Data i hora de creació de l'atribut, en format ISO 8601. Cal tindre en compte que la modificació correspon a la generació d'un nou atribut i la introducció a la llista de audit.
- user: Nom de l'operador que ha creat o modificat l'atribut.

Per la seua banda, el contingut de format és el següent:

```
{
  "type": "",
  "i18n": {},
  "extra": ""
}
```

JSON 3: Format

Amb el contingut:

- type: Tipus de atribut. Més endavant hi ha una llista extensiva d'atributs.
- i18n: diccionari amb les traduccions configurades del nom de l'atribut. La clau és el codi 639-1 i el contingut la forma del nom traduïda.
- extra: qualsevol altra informació que siga necessària per a definir el format de l'atribut.

Cal tindre en compte que, en el cas que l'atribut siga de tipus enllaç, el seu format canvia lleugerament. El valor no serà una cadena de caràcters sinó una llista de diccionaris amb el següent format:

```
{
  "entity": "",
  "eid": "",
  "id": 1
}
```

JSON 4: Enllaç

Amb el següent contingut:

- entity: Nom del tipus de entitat a la que s'està enllaçant.
- eid: codi eid de l'element enllaçat.
- ed: codi id de l'element enllaçat.

En el cas dels enllaços, també s'amplia l'element format, que afegirà al seu diccionari la clau «entity» amb el valor del nom de l'entitat que s'està enllaçant.

4.3.2 Plantilles

Existeix un tipus de dada bàsic en el sistema, que és «template» i correspon a les plantilles a través de les quals es definiran els atributs per defecte dels elements del sistema, així com les seues característiques.

L'objectiu és que el sistema de plantilles done una capacitat bàsica per a la generació de nous elements. Des del punt de vista de l'implementador, afegir un nou element al sistema es basa en obtenir la seua estructura i enviar-la a l'API de creació del component de base de dades.

L'estructura de dades del document plantilles serà idèntica a la resta d'elements del sistema. L'estructura interna de la plantilla estarà a un atribut del document plantilla anomenat «*attributesJson*» i contindrà el mateix format que la estructura de dades bàsica de tots els elements emmagatzemats, que correspon també a el format de l'API d'afegir o modificar elements.

4.3.3 Etiquetat i regles d'etiquetat

Com hem vist anteriorment, una de les bases del sistema és la seua flexibilitat i per tant que els usuaris puguen definir els seus llistats de forma ràpida i eficient. Per això, altre tipus de dada bàsica del sistema són les regles de filtrat.

Des del punt de vista d'estructures de dades, les regles de filtrat corresponen a elements normals del sistema, amb els seus documents estàndard i atributs.

L'element regla de filtrat conté un atribut amb un JSON que defineix la regla, així com altre atribut amb les etiquetes associades per aquesta regla. Les regles de filtrat es basen en la cerca de valors sobre els atributs dels elements a filtrar, que poden ser directes de l'element que s'està filtrant, o indirectes d'elements accessibles des del primer a través d'enllaços. No existeix limitació sobre el número d'atributs utilitzats per la regla, ni sobre la quantitat d'enllaços recorreguts per accedir a aquests.

L'objectiu del sistema de regles de filtrat és afegir una cua d'elements pendents d'etiquetar, on entren tots els elements nous i modificats del sistema. Seguint l'ordre de la cua i de forma asíncrona, cada regla del mateix tipus de l'element es verificarà sobre aquest, i en cas de complir la regla se l'afegirà a l'element la etiqueta o etiquetes corresponents. Així mateix, si una regla deixa de complir pel canvi a l'element, l'etiqueta o etiquetes corresponents seran retirades.

Algunes primitives de dades afegiran elements en la cua d'etiquetat, la qual no està exposada i funciona de forma completament independent.

4.3.4 Primitives d'accés a dades

Com hem vist anteriorment, el component de dades és independent de qualsevol altre del sistema, i s'accedirà a aquest a través dels APIs exposats. Els dos APIs bàsics seran els accessibles a través del mòdul de Python i REST, i contenen les següents primitives d'accés, creació i modificació de dades. Tots els paràmetres són enviats a través de un JSON amb els formats indicats.

4.3.4.1 Creació de dades

- **new**: És la primitiva bàsica per a generar elements. Com a paràmetre té una llista de JSONs complets d'elements concrets, i els introduirà a la base de dades. Útil per a introduir les primeres plantilles, o copiar elements. Cal tindre en compte que el contingut de l'id, eid i audit de cada atribut subministrats com a paràmetre s'ignoraran. Com a retorn s'obindrà una llista amb els documents dels elements generats, en el mateix ordre que a la entrada, o un codi d'error en cas de no poder-se donar d'alta.

4.3.4.2 Accés a dades

- **find**: correspon a la directiva d'accés a dades. Com a paràmetre es passa un JSON amb el següent format:

```
{
  "entity": "",
  "filters": [],
  "from": 0,
  "max": 0,
  "loadlinks": [],
}
```

JSON 5: Find

Amb el següent contingut:

- **entity**: El tipus d'element que es vol cercar.
- **filters**: Llista d'atributs que es volen cercar.
- **from**: Enter n que indica que s'ignoraran els primers n elements trobats.
- **max**: Enter n que indica que només es retornaran un màxim de n elements trobats.
- **Loadlinks**: Llista amb el nom dels atributs de tipus enllaç que es volen carregar. En el cas de carregar enllaços dins e enllaços, s'utilitzarà l'operador «.» per indicar que es tracta d'un atribut de l'enllaç.

Els filtres corresponen a un diccionari amb les següents claus:

- **attr**: nom de l'atribut a cercar. En el cas que es vullga cercar dins de un atribut de tipus enllaç, es poden separar a través de l'operador «.».
- **type**: tipus de filtre. Pot tindre els valors «=», «!=», «contains», «exists», «notexists», «full», «empty» i «regexp».
- **value**: contingut a buscar a l'atribut. Només obligatori en el cas que type siga «=», «!=», «contains» o «regexp».

4.3.4.3 Modificació de dades

- **mod**: Correspon a la primitiva per a modificar els documents d'elements ja donats d'alta a la base de dades. Recordem que, per les necessitats del projecte, cap element es pot esborrar completament i qualsevol canvi queda registrat a la auditoria de dades. Com a paràmetre es pot enviar l'estructura completa de l'element amb els canvis aplicats, o com a mínim aquest esquelet de l'estructura:

```
{
  "type": "",
  "attr": [],
  "id": 1,
}
```

JSON 6: Mod

On attr correspon a la llista d'atributs amb canvis, amb el format ja vist d'atribut.

Cal tindre en compte que, en el cas de faltar un atribut a l'estructura enviada, aquest no canviarà a la base de dades però en cas d'afegir un atribut que no existira abans, aquest s'afegirà. Els canvis sobre auditoria en els atributs s'ignoraran en tots els casos, ja que aquesta es reconstruirà automàticament en cada canvi.

El procés tornarà la llista de documents dels elements modificats, en el mateix ordre d'entrada, o un codi d'error en el cas de no haver-se pogut modificar o que passe el timeout de bloqueig.

- **FindMod**: Conjunció de la primitiva find i la mod. Modificarà tots els elements que retorne el resultat de la cerca.

4.3.5 Implementació de les funcions del component dades

Com hem vist anteriorment, el component dades serà implementat com un servei independent de la resta de l'aplicació, com un micro-servei dins del contenidor i accessible a través de l'API python a tota la resta de servicis del contenidor, així com externament a través de l'API REST.

Existeixen dos estructures extra internes: un diccionari amb els elements actualment bloquejats per modificació, i una llista d'elements a la espera d'etiquetat.

El procés d'etiquetat és sempre per element i es reavaluen totes les etiquetes possibles per aquest en un ordre FIFO, i de forma asíncrona a la resta de processos.

El procés de canvi en cascada (triggers) es fa sempre de forma síncrona.

A banda de les funcions exposades de l'API, el funcionament intern una vegada rebudes les dades funciona de la següent manera:

- Nou element: neteja de llistes d'auditoria en els atributs, addició del document de l'element dins de la base de dades amb generació de nou id, assignació de eid, afegir nou element a la cola d'etiquetat.
- Modificació d'element: Si l'element està al diccionari de bloquejats, s'espera a que deixi d'estar-lo. Durant el procés, l'element s'afegeix al diccionari d'elements bloquejats per modificació. Per cada atribut i element modificat, es comprova si la modificació és real, en cas afirmatiu es copia la situació actual a la llista de auditoria i es modifiquen els valors pels nous introduïts. Procés de canvis en cascada sobre la modificació de l'element. Afegir nou element a la cola d'etiquetat.

4.4 Disseny del component servidor

4.4.1 Mòduls

El component servidor, tal com hem vist anteriorment, està format per un conjunt de mòduls que funcionen com a micro-servicis que, encara que són coherents entre ells, executen les accions de forma independent.

Cadascun d'aquest mòduls tenen una URL bàsica única a través de la qual el component client accedeix a les dades per a mostrar els continguts. A banda, cada servici concret del mòdul té un nom que, afegit a la URL bàsica, és l'accés a aquest via REST.

Tots els accessos a dades dels mòduls es realitzen a través de l'API Python del component de dades.

L'estructura bàsica de tots els mòduls és la següent:

- Codificació de servicis en **Python**: és on està el codi flask per al qual, davant d'una petició de servici per part del component client, se li torne la informació sol·licitada o es realitzen les modificacions demanades.
- Codi HTML+**Javascript** específic del mòdul: en alguns casos, cal modificar el funcionament del component client amb funcionalitats extra o modificades. Aquest codi va injectat directament en el client, o es tracta de vistes específiques que s'accedeixen des del client. A la càrrega de cada vista que corresponga a un mòdul concret, el component client farà una cridada a la URL del mòdul que conté aquest codi extra per a la seua utilització. Cal tindre en compte que aquest codi pot modificar substancialment el funcionament general del component client, i fins i tot sobrecarregar les seues funcions declarades.

- Codificació d'**esdeveniments** en Python: En ocasions hi ha accions que s'executen degut a esdeveniments externs com pot ser una alarma o un procés que es fa només en un dia concret. Aquesta codificació ha d'estar separada de la que correspon als servicis executats manualment. També s'introdueixen ací els *triggers* o qualsevol esdeveniment que s'execute quan un element del mòdul corresponent canvia.

Exemples d'esdeveniments són una acció que s'execute tots els dies a les 2h o la còpia d'un atribut d'un element modificat a l'atribut d'altre element enllaçat en unes condicions concretes.

Els mòduls que entren al disseny d'aquest projecte, són els següents:

- **Plantilles** (correspon a entitat plantilla)
- **Entitat genèrica**
- **Permisos** (correspon a entitat rol i grup)
- **Regles de filtrat** (entitats regles i etiquetes)
- **Documents**
- **Informes**
- **Vendes** (correspon a entitats contracte, comanda de client, i entrega)
- **Compres** (correspon a entitats comanda, proveïdor i Recepció)
- **Element de compravenda** (correspon a entitats element producte i venda programada).
- **Proposta** (correspon a entitats proposta i catàleg)
- **Facturació** (correspon a entitat factura i diari)

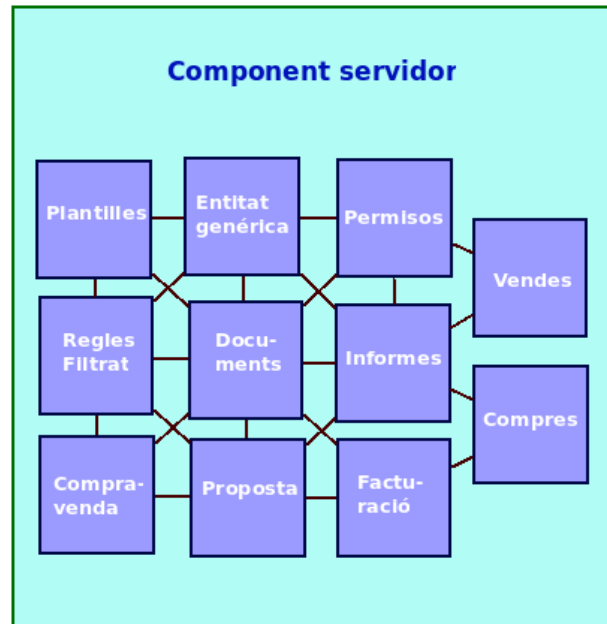


Figura 12: Mòduls del component servidor

No és l'objectiu d'aquesta memòria fer una descripció detallada del funcionament de tots els mòduls, però si que es mostra a continuació una descripció i funcionament general de cadascú d'ells per ajudar a comprendre el funcionament general d'aquest component.

4.4.2 Mòdul plantilles

Implementa les vistes necessàries per a la gestió de plantilles. La plantilla és un tipus de dades que, a banda dels atributs necessaris, conté un atribut anomenat «attributesJson», de tipus JSON, que dins porta l'estructura bàsica per generar un element del tipus corresponent. L'estructura és idèntica a la utilitzada a la base de dades, tenint en compte que els valors introduïts a la plantilla corresponen a valors per defecte.

Les accions inicials possibles a plantilla són: duplicar.

Els estats possibles de plantilla són: Vigent i Obsolet. Tots es poden obtenir des del seu llistat.

4.4.3 Mòdul entitat genèrica

Mòdul base que s'utilitzarà per ampliar al fer altres mòduls. Proveirà les vistes necessàries per a mostrar llistats de etiquetes, fer filtres als llistats, mostrar la vista individual de cada element i editar els elements.

Les accions possibles són duplicar, afegir atribut i exportar elements.

Els estats són vigent i obsolet.

4.4.4 Mòdul permisos

Aquest mòdul correspon a les entitats Rols i Grups.

Rols és un tipus de dada que controla per cadascuna de les entitats els permisos que té un usuari per interactuar amb ella. Per defecte, els permisos de cadascuna de les entitats són «visualitzar», «editar», «cancel·lar» i «autoritzar».

Grups és una entitat que té dos objectius diferenciats: per una banda limita les etiquetes que un usuari qualsevol pot visualitzar en el menú d'etiquetes de la part esquerra de la pàgina inicial, i per l'altra limita en la vista en taula la informació a la que no té grup o com a mínim té un grup en comú a l'usuari, de forma independent a l'entitat.

A través de rols i grups es limita l'accés a la informació d'usuaris concrets, i la interacció d'aquests usuaris amb la informació; cada usuari té assignat sempre un únic rol, i pot tindre un o més grups.

4.4.5 Mòdul regles de filtrat

Les regles de filtrat és la forma per posar contingut a una etiqueta per mostrar-la després en el menú d'etiquetes de la pàgina d'inici, i la forma d'accedir a la informació amb vista en taula.

Cada regla de filtrat està composta per un JSON amb una cerca concreta a la base de dades, configurada a través d'un interfície i emmagatzemada en el

mateix format que les consultes a l'estructura de dades, i una o vàries etiquetes.

En el registre o modificació de dades, cada dada modificada és revisada contra totes les regles de filtrat que són del seu mateix tipus d'element; en el cas que l'element no complira anteriorment la regla i la complisca després del canvi, serà marcat amb l'etiqueta o etiquetes indicades. Si l'element ja estiguera etiquetat i deixara de complir la regla, perdrà en la seua estructura de dades la marca d'eixa etiqueta.

L'objectiu de les regles de filtrat és obtindre llistats dinàmics en temps real ja preparats, que poden ser carregats pels usuaris sense tindre que executar la cerca en cada càrrega, pel seu cost temporal i de recursos.

4.4.6 Mòdul de documentació

Conté la entitat de documentació, que a banda de les accions genèriques gestiona el control de versions dels documents en el sistema.

Mai poden haver dos documents amb la mateixa codificació i mateix número de versió. Els documents poden tindre adjunts, que poden ser pujats de forma automàtica (per exemple, un informe donarà d'alta un nou document amb un adjunt que correspon al informe generat), o manual per un usuari. En el cas de canviar un adjunt d'un document, es generarà un nou element document idèntic a l'anterior amb un nou nombre de versió, i l'original passarà a obsolet.

Les accions extra de documents són nova versió, que permet des de la vista en taula pujar un nou adjunt al document, i recuperar versió que passa a obsolet la última versió del document seleccionat i recupera com a vigent la triada.

4.4.7 Mòdul d'informes

Aquest mòdul proveeix totes les funcions necessàries per a generar qualsevol tipus d'informe amb dades del sistema. Està compost de dos parts:

- Informes interns: Tot el procés es realitza dins de l'aplicació, i l'objectiu final és un PDF o un XLS basat en una plantilla html.

En aquest cas s'ha estudiat si utilitzar per als documents PDF la mateixa tàctica que en altres desenvolupaments interns de l'empresa, que és generar el html des de l'aplicació i utilitzar una ferramenta externa (*wkhtmltopdf*¹⁵) per a convertir aquest html en PDF, o utilitzar el mòdul Flask-WeasyPrint per a generar de forma directa aquest arxiu.

Degut a les necessitats de signatura electrònica, tant de PDF com de factura electrònica legal, i també les obligacions de certs clients que els exigeixen les seues pròpies administracions formats específics, no hi ha altra opció que utilitzar ferramentes externes per a processar els informes generats. Vist això, es decideix continuar utilitzant *wkhtmltopdf* per

15 <https://wkhtmltopdf.org/>

la seua potència i flexibilitat, en lloc del mecanisme intern de Flask que igualment caldrà que ampliar amb les ferramentes externes.

- API per utilitzar un reporter extern: s'implementarà una API simple basada en taules (format estructurat de dades) per a poder utilitzar amb reporters externs com per exemple Crystal Reports.

4.4.8 Mòdul de vendes

Aquest mòdul és el que gestionarà les vendes i contractes realitzats a través de l'aplicació. Ha d'anar integrat amb la funcionalitat del CRM intern de ICSA per al seguiment de les propostes realitzades.

Conté les entitats:

- Contracte: correspon a un contracte a llarg termini, compta amb les accions «Nova comanda» i «Informe».
- Comanda de client: correspon a una venda concreta, amb accions compta amb les accions «Entrega», «Crear productes contracte», «Assignar productes a contracte», «Crear servicis contracte», «Assignar servicis a contracte», «Informe».
- Entrega: Correspon a un albarà d'entrega, diferenciat al de la comanda i al de compra. Les accions són «Revertir» i «Facturar».

4.4.9 Mòdul de compres

Gestió de compres i pagaments de matèries primeres, fungibles i inventariables utilitzats a la empresa o venuts a terceres persones. A través del mòdul de compres es pot obtenir estoc, inventari i adquisicions.

El mòdul de compres ha d'estar així mateix integrat amb comptabilitat (aplicació externa), per a gestionar per una banda les compres contra vendes a client, i per altra les adquisicions internes de la empresa.

A través de compres també es gestionarà les pèrdues d'estoc per motius diversos, com per exemple deterioraments o caducitats.

4.4.10 Mòdul de compravenda

És l'enllaç entre l'estoc i la venda a client en el cas de les compres que van a estoc, o les vendes que es fan baix comanda.

A través d'aquest mòdul i les seues accions i procediments es permet per una banda realitzar compres automàtiques assignades a una venda en el cas que es realitze sense estoc, i per l'altra banda assignar estoc ja adquirit a vendes a client noves.

Cal tindre en compte que aquest mòdul també gestionarà els trencaments d'estoc (compres automàtiques per reducció de l'estoc baix del límit indicat), així com els *robatoris* d'estocs assignats ja a una venda (quan un estoc assignat

nat a una venda realitzada s'utilitza finalment per altra venda més urgent o imminent).

4.4.11 Mòdul de proposta

Forma part del procediment de CRM, i per això cal que estiga integrat en la resta d'aplicacions utilitzades per l'empresa. Una proposta és una oferta en ferm realitzada a un client, que es pot convertir en qualsevol moment en un contracte.

Les propostes no seran realitzades directament a l'ERP perquè venen de processos anteriors de vendes, però han d'estar registrades, a través d'integració, per a mantindre la traçabilitat de tot el procés de venda.

Fins que una proposta es converteix a contracte, no afecta als processos de estoc ni se li genera cap venda associada.

4.4.12 Mòdul de facturació

Les factures generades són el final del procés de venda de l'ERP, on es calculen tots els assentaments de comptabilitat i es genera l'informe de factura i el seu document corresponent per a mantindre en el sistema.

Els assentaments de comptabilitat han d'estar integrats amb l'aplicació de comptabilitat de l'empresa, a banda de quedar registrat a l'ERP, i una vegada traspassats ja no es podran modificar.

Sobre les factures s'han de poder fer rectificacions i abonaments de factures completes, que generaran també nous assentaments de comptabilitat que seran traspassats igual que noves factures.

Dins del mòdul de facturació aniran els models necessaris, que seran generats i signats de la forma corresponent pel mòdul d'informes.

4.5 Disseny del component client

4.5.1 Consideracions prèvies

Com hem vist anteriorment, la separació del component client del component servidor es dona per la necessitat d'unificar al màxim el funcionament de totes les interfícies del sistema. Tal com s'ha explicat a l'apartat 1.1 Context i justificació del Treball, i 1.3 Enfocament i mètode seguit, aquest disseny està molt basat en les decisions preses amb projectes anteriors, on la diversitat d'interfícies ha complicat en moltes ocasions, i limitat en altres, el desenvolupament final de la ferramenta.

Aquest component està programat íntegrament amb HTML+CSS+JavaScript, i les úniques parts en Python són les necessàries per servir aquests arxius, ja

siguen estàtics i fixes del component, o dinàmics que provenen de qualsevol dels mòduls del component servidor.

Les interfícies bàsiques del component a anem a descriure a aquesta memòria són:

- **Pàgina d'accés:** correspon als interfícies necessaris per a poder accedir al sistema, en una instància concreta.
- **Pàgina d'inici:** correspon a tota la estructura bàsica una vegada dins de l'aplicació, i a partir de la qual es construeixen la resta de interfícies.
- **Menú de configuració:** mecanisme per a que els usuaris canvien certes parts del sistema.
- **Menú d'etiquetes:** forma d'accés als llistats precarregats de dades, definits pels propis usuaris.
- **Vista en tabla d'elements:** forma de mostrar els llistats en el sistema.
- **Vista individual d'elements:** accés al detall d'un o diversos elements dins del sistema.
- **Selecció d'elements d'una llista tancada:** interfície per a fer una selecció de entre una llista d'elements definida pel sistema o per l'usuari, a través de filtres o cerques.

Igual que a l'apartat de mòduls, no és objectiu d'aquesta memòria el detallar la visualització i funcionament detallat d'aquestes interfícies, però sí que s'inclou una descripció breu del seu objectiu i funcionament per a millorar l'enteniment de l'aplicació pel lector.

4.5.2 Pàgina d'accés

Es tracta del començament de la interacció de l'usuari amb la interfície, i li permet accedir al sistema si ja té un mètode acceptat d'accés.

Els mètodes d'accés a implementar seran el de nom d'usuari i contrasenya, i també via certificat digital instal·lat en l'explorador web.

Internament, aquest interfície també gestiona la capacitat de crear un nou compte associat a una invitació generada per un administrador, on l'usuari ha d'omplir totes les seues dades, inclòs la contrasenya escollida.

La interfície també compta amb un sistema simple per a recuperar la contrasenya perduda, on a través d'indicar el correu electrònic amb el que va ser registrat, obté per aquest mitjà un enllaç a la pàgina de la invitació amb les seues dades i la capacitat de posar una nova contrasenya.

En el cas on un usuari tinga configurat l'accés per certificat digital, només l'administrador podrà enviar-li una nova invitació per a accedir a través de contrasenya.

A la figura 13 que es troba a la pàgina 39, al capítol de proves d'usuari, podem trobar un primer moc-up de com s'ha de mostrar aquesta interfície.

4.5.3 Pàgina d'inici

Es tracta de la part més important del component de client, ja que és la que uneix la resta de components i els interconnecta per a que la interfície siga coherent. S'accedeix a ella una vegada identificat l'usuari a la pàgina d'accés i també cada vegada que es recarrega de forma completa la pàgina a l'explorador web.

A la figura 14 que es troba a la pàgina 40 tenim un primer moc-up d'aquesta interfície, utilitzat per a les proves amb usuaris, on trobem a la esquerra el menú d'etiquetes, a la part central un llistat obert en la modalitat simple i a la part de dalt el menú de configuració.

Aquesta interfície ha d'implementar la funció per a fer peticions a les URLs corresponents dels mòduls del component servidor, així com la capacitat de mostrar els resultats en pantalla. També és la que gestiona la petició de permisos d'usuari des del punt de vista de la interfície, encara que el control d'accés a dades es fa en la part del component servidor.

Els usuaris administradors tenen el menú de configuració disponible en aquesta interfície, i bàsicament és la que fa de punt d'unió amb tota la resta d'interfícies del sistema.

4.5.4 Menú de configuració

Correspon a la part superior de la pàgina d'inici. Mostra la informació de l'operador connectat, li permet modificar aquesta i, en el cas que l'usuari tinga els permisos corresponents, també dona accés a la configuració.

Al moc-up de la figura 14 es pot veure dibuixat a la part superior dreta.

4.5.5 Menú d'etiquetes

Correspon a la part esquerra de la pàgina d'inici, i és on van les etiquetes a través de les quals es poden accedir als llistats de nevis, precarregats a través de les regles de filtrat.

El menú d'etiquetes és configurable per cada grup d'usuaris, de manera que aquests només podran veure les etiquetes configurades al seu grup. En el cas que un usuari tinga més d'un grup, li apareixeran les etiquetes en diverses agrupacions dins d'aquest menú d'etiquetes.

El menú és la forma d'accés als llistats de la vista en taula, i la forma normal d'accedir a la informació corresponent en cadascun dels elements del sistema. Quan es selecciona una etiqueta de la llista, la part central de la pàgina actual canviarà pel llistat corresponent a tots els elements etiquetats amb l'etiqueta seleccionada.

Al moc-up de la figura 14 es pot veure dibuixat a la part esquerra.

4.5.6 Vista en taula d'elements

La vista en taula d'elements és la forma més comú de visualitzar i cercar informació a l'aplicació. S'accedeix de forma normal a ella des de qualsevol de les etiquetes del menú d'etiquetes, i mostrarà una línia per cada element que està dins de la etiqueta seleccionada.

El nombre de files que es poden visualitzar a cada pàgina de la taula és configurable entre un mínim de 20 i un màxim d'infinít. En el cas que hi haja més línies disponibles que el màxim definit, aquestes es dividiran en pàgines amb el màxim de línies possibles, i apareixerà un menú de navegació a la part superior dreta que permet anar a la primera pàgina, última pàgina, pàgina anterior i pàgina següent.

La càrrega de les línies es asíncrona i seqüencial. Al carregar un llistat es carregaran sempre les primeres 100 línies (si existeixen) i els identificadors de la resta de línies disponibles (fins a un màxim de 500 pàgines). Posteriorment s'aniran carregant de forma asíncrona les línies que falten a través de peticions amb els seus identificadors, de 100 en 100. En el cas d'utilitzar paginació, només es carregaran les línies en el cas que es sol·liciten les pàgines, en paquets de 100 en 100.

Les columnes de la vista en taula són configurables per usuari, i perduren entre sessions. Les columnes possibles són els atributs de qualsevol de les plantilles del tipus d'element. No es poden extreure columnes d'atributs que no estan a cap plantilla. Per a seleccionar les columnes, s'utilitza la interfície de selecció d'elements d'una llista tancada, que correspon a totes les possibles. Per defecte, les columnes són les marcades com a «*vista en taula*» al format de l'atribut de la primera plantilla sistema del tipus corresponent.

Adicionalment, es poden seleccionar columnes que provenen d'atributs d'elements enllaçats pels de la llista.

Les taules es poden utilitzar de dos formes:

- Taula simple: totes les files de la taula corresponen a elements de l'etiqueta seleccionada.
- Taula doble: la taula està partida verticalment en dos, a la primera part es mostra de la mateixa forma que si fora una taula simple, i a la part de la dreta es carreguen els elements configurats en un atribut enllaç múltiple definit de l'element seleccionat de la part esquerra. El contingut carrega en el moment de seleccionar un element a la taula original. Si a la taula de l'esquerra se selecciona més d'un element, a la de la dreta apareixeran tots els enllaços de la conjunció dels seleccionats.

Al moc-up de la figura 14 podem veure en la part central una primera aproximació a una taula simple, amb les columnes configurades i en visualització directa, i amb paginació de 20 línies per pàgina.

La taula conté també un menú superior amb les accions i canvis d'estat possibles. Aquestes es carreguen depenent del tipus d'element a través del JavaScript específic de cada mòdul de servidor. En cas d'utilitzar la taula doble, cada taula tindrà el seu menú propi d'accions i canvis d'estat.

Des de la taula, es pot seleccionar un o diversos elements (línies) i obrir la vista individual d'aquests; en el cas d'utilitzar la taula doble, la vista individual s'obrirà només en la part que corresponga de la taula.

L'ordenació de la taula per defecte serà sempre en ordre d'antiguitat invers, on l'element més nou serà el primer en aparèixer. Aquesta ordenació es pot canviar a través de qualsevol de les columnes mostrades, punxant en la seua capçalera; s'ordenarà per eixe atribut tant de forma directa com inversa, sempre en ordre alfabètic per a textos, i numèric per atribut de tipus nombre.

La vista en taula també conté un filtre simple, que busca dins del contingut d'aquesta i només en els atributs de les columnes mostrades el text introduït. Així mateix, existeix la possibilitat de carregar a la taula un filtre avançat, que permet cercar per atributs específics o combinació d'aquests.

Per últim, en alguns casos por haver-hi agrupació específica en algunes de les taules mostrades. Aquesta agrupació anirà configurada directament per tipus d'entitat i canviarà l'ordenació en que la taula serà mostrada per defecte.

4.5.7 Vista individual d'elements

La vista individual d'elements és la forma normal de registre de nous elements o la modificació d'aquests.

A la vista individual es mostra en un nombre configurable de columnes per usuari cadascun dels atributs que existeixen a l'entitat. En el cas que l'atribut siga editable i l'usuari tinga permisos per editar l'entitat, aquest es podrà modificar.

Les modificacions realitzades a la vista individual no s'envien al component servidor ni són emmagatzemades en cap cas a no ser que es preme de forma explícita el botó de guardar que es mostra en tot moment en la part superior i inferior de la interfície.

Si la taula que ha donat pas a la vista individual tenia més d'un element seleccionar, s'obriran tots els elements corresponents en forma de fitxes d'arxiu, on l'usuari pot canviar entre elles a través d'uns botons de navegació a la part superior i inferior.

De forma addicional, els administradors poden configurar en les plantilles els atributs ocults per defecte; en aquest cas, aquests atributs només es visualitzaran o modificaran si el rol de l'usuari ho permet i a més a més preme de forma explícita un botó per a mostrar-los a la interfície.

Podem veure una primera aproximació d'aquesta interfície, amb dues columnes, i paginació en forma de fitxes d'arxiu, en la maqueta de la figura 15 a la pàgina 41 del capítol de proves d'usuari.

4.5.8 Selecció d'elements d'una llista tancada.

Aquesta és l'última interfície de les bàsiques del sistema, i permet seleccionar un element d'una llista extensa i no predefinida. Per exemple, és útil quan dins d'un element cal posar enllaços a altre element del sistema, ja siga dels continguts dins d'una etiqueta concreta o amb un filtre predefinit.

Aquesta interfície, en format de pop-up flotant, mostra en la seua part superior una llista en taula amb columnes reduïdes dels elements que es poden seleccionar, així com la casella per a seleccionar-los. A la seua part inferior apareix una segona llista amb els elements ja seleccionats, on es permet eliminar-los o canviar el seu ordre arrastrant per als casos on l'ordre de la selecció és important.

En alguns casos, aquest interfície també permet seleccionar la quantitat d'elements seleccionat de cada tipus dels presentats.

Cal tindre en compte que aquesta interfície pot anar filtrada o obtindre les dades de moltes fonts diferents, depenent del tipus d'element al que apunta, el element d'on prové l'enllaç, o de forma específica de cada atribut.

Així mateix, aquesta interfície pot deixar només seleccionar un element d'entre tots els possibles, fer una selecció múltiple d'elements i fins i tot en alguns casos permetre seleccionar una quantitat determinada de cada element triat.

5. Proves amb usuaris clau

5.1 Procediment

Per avaluar el funcionament dels nous procediments i interfícies d'usuari que s'utilitzaran en l'ERP, abans de la seua implementació, s'han realitzat una sèrie de proves d'aquests amb alguns usuaris claus de tots els departaments implicats de l'empresa.

L'objectiu d'aquestes és verificar si les interfícies són clares i simples per a fer les tasques bàsiques, i obtenir feedback dels usuaris abans de començar amb la implementació.

El procediment ha sigut la realització de maquetes dels interfícies, tant dibuixades en paper com a través de l'aplicació MockFlow, i se les ha mostrat de forma individual als usuaris, sense cap explicació, per a que proposen com haurien de continuar per a fer algunes tasques bàsiques del sistema (Marcos Antonio Catalán Vega, [7]).

En especial, les interfícies representades en maqueta han sigut les següents:

- **Accés al sistema:** identificació de l'usuari per accedir al sistema. A la figura 13 podem veure la maqueta mostrada als usuaris.
- **Pàgina d'inici:** indicadors i objectius generals, menú d'etiquetes i panell de configuració d'usuari. A la figura 14, amb la vista en taula.
- **Vista en taula:** vista tabulada de una sèrie d'elements del sistema. A la figura 14, integrada amb la pàgina d'inici.
- **Vista individual:** vista en fitxa del contingut d'un element. A la figura 15, un exemple dels mostrats.



WWW.ICSA.ES MÉDICA ANÁLITICA LIMS QUIENES SOMOS CONTACTOS

LIMS Servicio Técnico
MATERIAL DE LABORATORIO
DATA
Analítica
Laboratorios de Análisis
Consultoría 17025
PATRONES
Motilidad digestiva Consultoría
ANATOMÍA PATOLÓGICA
ISO 9001
Médica
Laboratorios Clínicos
Neurofisiología ISO 15189
UROLOGÍA

Acceso de usuario

Usuario:

Contraseña:

[Acceder con certificado electrónico](#)

[¿Has olvidado la contraseña?](#)

Figura 13: Proposta inicial de la interfície d'accés

Els procediments que s'han testat són els següents:

1. **Accés al sistema.**

2. Canvi de **contrasenya** de l'usuari.
3. Creació d'un **nou compte** client en el sistema.
4. **Localització** d'un compte client en el sistema pel seu nom.
5. Localització de la data d'**accions** sobre un compte client.

A l'usuari amb el que s'està fent la prova se li presenten les interfícies en el mateix ordre que li apareixeran en el sistema, i se li demana com faria els procediments descrits, en l'ordre indicat. Se registra el temps que tarda l'usuari en fer el procés, o la seua incapacitat en el cas de que no li siga possible.

El director de la prova pot contestar a totes les preguntes que fan els usuaris, excepte dir directament com continuar. No pot donar pistes ni parlar durant la prova a no ser que siga la resposta a una pregunta directa.

The screenshot shows a CRM application interface. On the left, there is a sidebar with a tree view containing categories like 'CRM', 'Oportunidad seguimie', 'CRM Cuentas', and 'CRM Propuestas'. The main content area is titled 'Todas las cuentas' and features a search bar and a table of accounts. The table has columns for 'Nombre', 'Razón Social', 'Tipo De Entidad', and 'Descripción'. Below the table, it indicates '0 elementos seleccionados de 4270'. At the top right, the user's name 'Álvar Cuevas i Fajardo Administración' is visible.

Nombre	Razón Social	Tipo De Entidad	Descripción
AC-21-0050	PATCONSULT. LAB.	Laboratorio	PATCONSULT. LAB.
AC-21-0049	Lubaq Corporate, S.L	Laboratorio	Lubaq Corporate, S.L
AC-21-0048	ASIng Servicios de Ingeniería, S.L.	Industria	ASIng Servicios de Ingeniería, S.L.
AC-21-0047	Yosmar Carolina Pérez González	Laboratorio	Yosmar Carolina Pérez González
AC-21-0046	(IBIMA) Instituto de investigación Biomédicas de Málaga	Universidad	(IBIMA) Instituto de investigación Biomé
AC-21-0045	Inveseed Technologies, SL	Industria	Inveseed Technologies, SL
AC-21-0044	Leica Microsistemas S.L.U.	Facilitador	Leica Microsistemas S.L.U.
AC-21-0043	Centro médico Voth S.L.	Laboratorio	Centro médico Voth S.L.
AC-21-0042	BIOTECNAL	Laboratorio	BIOTECNAL
AC-21-0041	Laboratorio de Sanidad Animal de la diputación de Vizcaya	Laboratorio	Laboratorio de Sanidad Animal de la dip
AC-21-0040	Plataforma Logística Sanitaria de Huelva	Hospital	Plataforma Logística Sanitaria de Huelv
AC-21-0039	IES MEDICAL	Laboratorio	IES MEDICAL
AC-21-0038	Dilemma Solutions, S.L	Laboratorio	Dilemma Solutions, S.L
AC-21-0037	Agrofresh	Industria	Agrofresh
AC-21-0036	Institut Català de la Salut	Instituto	Institut Català de la Salut
AC-21-0035	HOSPITAL CLINICO UNIVERSITARIO DE VALENCIA	Hospital	HOSPITAL CLINICO UNIVERSITARIO I
AC-21-0034	CASA MAS, SL	Industria	CASA MAS, SL
AC-21-0033	Nilsson Laboratorios, SL	Laboratorio	Nilsson Laboratorios, SL
AC-21-0032	Air Validation	Distribuidor	Air Validation
AC-21-0031	Damián Vaquero S.L	Industria	Damián Vaquero S.L

Figura 14: Proposta inicial de pàgina d'inici i vista en taula

Per últim, abans d'acabar la prova es pregunta a l'usuari si té algun comentari o proposta de millora sobre les interfícies que ha vist. La prova està preparada per a que pugui ser feta en 15 minuts, reduint al màxim la distracció dels treballadors sobre la seua feina en horari laboral. En el cas de passar d'aquest temps, es marca que l'usuari no ha sigut capaç d'acabar, i es dona per terminada la prova.

S'ha realitzat aquest test a un total de 6 persones, dividides en 2 del departament de secretaria, 1 del departament de logística, 2 del departament comercial i 1 del departament de direcció. Els resultats es troben a continuació.

|

Razón social *	<input type="text" value="Leica Microsistemas S.L.U."/>	Descripción *	<input type="text" value="Leica Microsistemas S.L.U."/>
Área	<input type="text"/>	CIF	<input type="text" value="B58521147"/>
Teléfono	<input type="text"/>	Provincia	<input type="text" value="Barcelona"/>
Grupo	<input type="text"/>	Autonomía	<input type="text" value="CATALUÑA"/>
Tipo de entidad *	<input type="text" value="Facilitador"/>	Consultor	<input type="text"/>
Comercial	<input type="text"/>	Clasificación avanzada *	<input type="text" value="Anatomía Patológica BBDD ICSA FACILITADORES"/>
Clasificación	<input type="text"/>		

Acciones

Tipo	Fecha	Operador	Campaña	Oportunidad	Propuesta	Observaciones
Propuesta	2021-05-25	Herrero, Jose Carlos			PR-21-0562	Viene de nuev
Propuesta	2021-05-25	Herrero, Jose Carlos			PR-21-0561	Viene de nuev

Figura 15: Proposta inicial de vista individual

5.2 Secretaria

Prova 1:

- Usuari: Inma Garcia.
- Edat: 45 anys.
- Data: 11 de maig de 2021.
- Rol a l'empresa: Facturació i tresoreria.
- Temps total de la prova: 12 minuts.
- Temps dels procediments proposats:
 - 1: 1 minut.
 - 2: 3 minuts.
 - 3: 2 minuts.
 - 4: 3 minuts
 - 5: 3 minuts.
- Comentaris de l'usuari:
 - Proposa la modificació de la informació d'usuari cap a un lloc més visible i clar.
 - Proposa la modificació de la vista en taula per a poder tindre les dades de les taules d'un element directament a la vista original.
 - Proposa que les taules siguin directament editables.

Prova 2:

- Usuari: Lola González.

- Edat: 67 anys.
- Data: 11 de maig de 2021.
- Rol a l'empresa: Secretària.
- Temps total de la prova: Sobrepasat temps màxim (15 minuts).
- Temps dels procediments proposats:
 - 1: 1 minut.
 - 2: 5 minuts.
 - 3: 5 minuts.
 - 4: 3 minuts
 - 5: temps màxim sobrepasat.
- Comentaris de l'usuari:
 - Indica que té moltes dificultats per entendre com funciona sense més indicacions.
 - Redunda en la necessitat de treballar directament sobre els indicadors.
 - A partir de la tercera prova apareix frustrada i poc comunicativa.
 - Acaba el temps sense que arribe a resoldre l'última prova.

5.3 Logística

Prova 3:

- Usuari: German Bolsano.
- Edat: 32 anys.
- Data: 12 de maig de 2021.
- Rol a l'empresa: Encarregat de magatzem.
- Temps total de la prova: 8 minuts.
- Temps dels procediments proposats:
 - 1: 1 minut.
 - 2: 2 minuts.
 - 3: 1 minuts.
 - 4: 2 minuts
 - 5: 2 minuts
- Comentaris de l'usuari:
 - Proposa la modificació de la vista en taula per a poder tindre les dades de les taules d'un element directament a la vista original. Mateixa proposta que l'usuari 1.
 - Comenta que ha treballat amb diversos sistemes d'estoc, i que tots són molt pareguts.

5.4 Comercial

Prova 4:

- Usuari: Jose Carlos Herrero.
- Edat: 41 anys.
- Data: 13 de maig de 2021.
- Rol a l'empresa: Comercial consultoria i analítica.
- Temps total de la prova: 11 minuts.
- Temps dels procediments proposats:

- 1: 1 minut.
- 2: 3 minuts.
- 3: 2 minuts.
- 4: 2 minuts
- 5: 3 minuts
- Comentaris de l'usuari:
 - Es queixa de que tinga que canviar de ferramenta, quan el CRM actual ja el coneix.
 - Una vegada acabada la prova, agraeix que les interfícies siguen molt paregudes al CRM actual, i demana aplicar filtres més complexes en les taules dins dels elements.

Prova 5:

- Usuari: Federico Sánchez.
- Edat: 46 anys.
- Data: 13 de maig de 2021.
- Rol a l'empresa: Comercial mèdica.
- Temps total de la prova: 10 minuts.
- Temps dels procediments proposats:
 - 1: 1 minut.
 - 2: 2 minuts.
 - 3: 2 minuts.
 - 4: 3 minuts
 - 5: 2 minuts
- Comentaris de l'usuari:
 - Li pareix tot molt assumible.
 - Agraeix que es tinga en compte als comercials, que normalment queden fora de les propostes de programari o canvis de procediments.

5.5 Direcció

Prova 6:

- Usuari: Maria Dolores Franco.
- Edat: 44 anys.
- Data: 18 de maig de 2021.
- Rol a l'empresa: Gerent.
- Temps total de la prova: 10 minuts.
- Temps dels procediments proposats:
 - 1: 1 minut.
 - 2: 2 minuts.
 - 3: 3 minuts.
 - 4: 3 minuts
 - 5: 1 minuts
- Comentaris de l'usuari:
 - Indica que el més important per a ella són els indicadors i que a la prova no es veuen.
 - Considera les interfícies mostrades clares i amb un funcionament prou estàndard com per a que siguen enteses sense formació especial.

5.6 Conclusions de les proves

El resultat de les proves és paregut a l'esperat, amb la següent taula de temps:

	Usuari 1	Usuari 2	Usuari 3	Usuari 4	Usuari 5	Usuari 6	Mitjana
Procediment 1	1	1	1	1	1	1	1,00
Procediment 2	3	5	2	3	2	2	2,83
Procediment 3	2	5	1	2	2	3	2,50
Procediment 4	3	3	2	2	3	3	2,67
Procediment 5	3	-	2	3	2	1	2,20
Total	12	> 15	8	11	10	10	10,20

Figura 16: Resultat de les proves amb usuaris

Les conclusions generals són, primer de tot, que la dificultat d'ús de l'aplicació creixerà amb l'edat de la persona que l'utilitzi i que haver utilitzat eines paregudes baixa molt el temps d'aprenentatge i facilita el seu ús.

Després de les proves d'ús, es decideix aplicar els següents **canvis en el disseny inicial**:

- Canviar la interfície de la vista en taula per a que pugui mostrar dos taules alhora. La segona taula es configura en cada vista i correspon a una de les taules internes de l'element seleccionat a la primera taula.
- Modificar la informació d'usuari per a que sigui accessible directament i no a través d'un menú ocult.
- Introduir l'opció en algunes taules específiques per a que algunes dades puguin ser modificades directament.

Així mateix, es pren la decisió de fer un **reforç especial en el pla de formació** per a tot el personal que no utilitzara directament l'ERP anterior, i també per als majors de 55 anys.

Per últim, pareix que per alguns usuaris els indicadors, que en un principi només queden esbossats en aquesta primera implementació, són molt importants i cal evolucionar-los. Per la limitació temporal de la implementació es continuen mantenint majoritàriament fora, però **cal continuar a futur el projecte** amb aquesta línia d'investigació i implementació.

6. Pla d'implementació

Amb el disseny preparat, es proposa un pla d'implementació inicial, basat en els requisits actuals de l'empresa i els recursos que pot cedir ja siga de forma completa o parcial per a la implementació i programació de l'ERP.

6.1 Recursos implicats

Els recursos implicats en aquest projecte són per una part recursos tècnics i per altra recursos humans.

Dins dels **recursos tècnics**, i a banda del material de treball normal dels programadors de l'empresa, es comptarà amb 2 màquines virtuals dins d'Amazon EC2, del tipus *c5.4xlarge* amb 16 nuclis de procés en paral·lel, 32 GB de RAM i 10 Gb de connexió a la xarxa per a provar directament el funcionament del codi en situacions de normalitat. Així mateix, també es comptarà amb una màquina *db.r5.large* amb 2 nuclis de computació i 16 GiB de RAM per a la base de dades gestionada Amazon DocumentDB.

Dins dels **recursos humans** necessaris per al projecte, es comptarà amb 3 dels programadors que dedicaran parcialment les seues hores setmanals per aquest projecte, de la següent manera:

- Programador 1: 60% del seu temps per al projecte.
- Programador 2: 60% del seu temps per al projecte.
- Programador 3: 40% del seu temps per al projecte.

Aquesta dedicació parcial no és homogènia, ja que els programadors 1 i 2 tindran més dedicació al començament que al final i en canvi el programador 3 farà la majoria de la seua tasca a la fase d'implementació i posada en funcionament.

Així mateix, i encara que no treballarà de forma directa en el projecte, es farà **una contractació** el mes de juny d'un nou programador a l'empresa que ajudarà a mantindre la resta de tasques del departament tenint en compte que part de l'equip dedicarà part del seu temps en la programació i posada en funcionament de l'ERP.

6.2 Abast de la implementació

També cal tindre en compte que la implementació es farà utilitzant processos àgils, tal com s'explica al capítol 3. Limitacions sobrevingudes al projecte, intentant donar forma a nuclis funcionals en cada cicle de programació. La següent planificació correspon no a la finalització completa del projecte, sinó a una **primera fase** amb les necessitats bàsiques de l'ERP que s'ha dissenyat.

Això vol dir que, encara que la programació només arribe en aquest calendari fins a setembre de 2021, està previst que continue amb una segona fase de programació fins a gener de 2022 i una tercera fase fins a juny de 2022 que no entren dins de l'abast d'aquest projecte.

6.3 Riscos en la implementació

Com en qualsevol projecte d'implementació de programari, existeixen una sèrie de riscos que cal tindre en compte i minimitzar en la mesura de les possibilitats. En aquest cas els **riscos identificats** són els que apareixen a la figura 17.

ID	Tipus	Descripció	Probabilitat	Impacte
#1	Gestió	No acabar el projecte en el temps de la planificació	Mitjana	Alt
#2	Funcional	El disseny no compleix amb els requeriments dels usuaris	Baixa	Molt alt
#3	Gestió de canvi	Existeix resistència per part dels departaments implicats	Baixa	Mitjà
#4	Funcional	L'estructura de dades no accepta el volum subministrat	Baixa	Alt
#5	Tècnic	No és possible la integració amb altres aplicacions	Baixa	Baix
#6	Gestió	No hi ha recursos per a les fases següents del projecte	Mitjana	Mitjà
#7	Gestió	Incapacitat de fer nova contractació	Alta	Alt

Figura 17: Riscos detectats a la implementació del projecte

Per a fer el càlcul de la severitat de cada risc i per tant preparar un pla d'actuació s'ha utilitzat la taula mostrada a la figura 18.

		Probabilitat		
		Baixa	Mitjana	Alta
Impacte	Molt Baix	Baixa	Baixa	Baixa
	Baix	Baixa	Baixa	Mitjana
	Mitjà	Baixa	Mitjana	Alta
	Alt	Mitjana	Alta	Alta
	Molt alt	Alta	Alta	Alta

Figura 18: Plantilla de severitat de riscos

I, amb això, es proposen els següents plans d'actuació:

- **Risc #1**, amb severitat **alta**:
 - **Preventiu**: Fer un seguiment estricte de cada ítem d'implementació per a trobar retards a temps de corregir-los.
 - **Correctiu**: Ampliar els recursos necessaris o si no és possible retallar funcionalitats no bàsiques.
- **Risc #2**, amb severitat **alta**:
 - **Preventiu**: Mostrar el disseny a tots els usuaris abans de la implementació, i durant aquesta de l'evolució de procediments i interfícies.
 - **Correctiu**: En cas de detectar errades al disseny, fer les modificacions oportunes.
- **Risc #3**, amb severitat **baixa**:
 - **Preventiu**: Explicat a l'apartat 7.5 Gestió del canvi d'aquesta memòria.

- **Correctiu:** En el cas de trobar resistència al canvi, es parlarà directament amb els treballadors afectats per a buscar alternatives viables per a tots els departaments.
- **Risc #4**, amb severitat **mitjana**:
 - **Preventiu:** S'han fet proves de càrrega de les diferents opcions de SGBD, tal com estan explicades a l'apartat 4.1.4.1 Proves de rendiment.
 - **Correctiu:** En el cas de que l'estructura de dades fora insuficient, es farien les modificacions oportunes al disseny, amb la pèrdua de recursos que això comporta.
- **Risc #5**, amb severitat **baixa**:
 - **Preventiu:** S'ha verificat que, a priori, totes les ferramentes que s'han d'integrar tenen actiu un API per a fer-ho.
 - **Correctiu:** En el cas que es trobe un problema a les integracions, s'estudiarà l'ús de bases de dades intermèdies per a mantindre les dades actualitzades (Kent Sandoe, Gail Corbitt, Raymond Boykin, [8])
- **Risc #6**, amb severitat **mitjana**:
 - **Preventiu:** El disseny de la primera fase definit al capítol 4. Disseny de l'aplicació és funcional sense necessitat de la continuació del projecte.
 - **Correctiu:** En el cas que el projecte no pugui continuar, es pondrà en marxa només la primera fase a l'espera de que en un futur es recuperen els recursos necessaris per a continuar.
- **Risc #7**, amb severitat **alta**:
 - **Preventiu:** La situació del mercat de treball de programació a l'actualitat fa pensar que no siga fàcil trobar una persona com està previst. A banda de els candidats de la nova oferta de treball, es contactarà a persones no seleccionades d'ofertes de programació anteriors.
 - **Correctiu:** En el cas de no aconseguir la nova contractació, s'externalitzaran temporalment alguns dels serveis d'informàtica de l'empresa per a poder realitzar la implementació amb els recursos previstos.

6.3 Calendari d'implementació

La planificació proposada per aquesta primera fase d'implementació, dividida per mesos i components, i amb la dedicació total de projecte i mes, la podem trobar a la figura 19.

	Hores/persona	Previ	Maig	Juny	Juliol	Agost	Setembre
Preparació	60		60				
BD	120	80	20	20			
Servidor	160		46	70	44		
Client	180			36	82	20	42
Implementació	140	40				20	80
Posada en marxa	45						45
Dedicació	705		126	126	126	40	167

Figura 19: Planificació de la dedicació per mesos

Per altra banda, el personal de programació implicat en cadascun dels mesos/fases del projecte està a la figura 20.

	Maig	Juny	Juliol	Agost	Setembre
Personal d'implementació implicat	JG+AC	JG+AC	JG+AC	AC+EV	JG+AC+EV

Figura 20: Personal de programació implicat per mes

El temps previst per a finalitzar la implementació d'aquesta primera fase serà de **6 mesos**, amb els recursos indicats anteriorment. En especial, si dividim per components:

- La preparació i entorns necessaris estaran tots realitzats en la finalització del mes de maig de 2021.
- La implementació del component Base de Dades estarà finalitzat a l'acabar el mes de juny de 2021.
- El component servidor estarà, en aquesta primera fase, implementat a final de juliol de 2021.
- El component client estarà, també en aquesta primera fase, acabat per a final de setembre de 2021.

Cal tindre en compte que s'ha afegit per a setembre de 2021 unes hores corresponents a posada en marxa, que no correspon a la posada en marxa real del projecte amb formació d'usuaris finals. Aquesta posada en marxa indica el temps necessari per a crear i provar tot el sistema en una instància interna de proves, la qual s'utilitzarà a la fase final per a verificar els canvis necessaris identificats al paral·lel abans d'aplicar-los als usuaris.

Aquest pla d'implementació anirà seguit del pla de posada en marxa amb usuaris, i fins que aquest segon no finalitze no es podrà posar en marxa el disseny i implementació de la segona fase de l'ERP.

7. Posada en marxa i gestió del canvi

La posada en marxa d'una nova aplicació és un dels moments més complicats, ja que l'èxit de tot l'esforç realitzat queda en mans dels usuaris finals, que són els que decidiran si han millorat o no la seua situació amb el canvi o no.

Cal tindre en compte que en aquest cas, un ERP, existeix una gran quantitat d'usuaris finals amb objectius completament diferents entre ells, de manera que es pot donar el cas que la migració cap a la nova aplicació siga un èxit per alguns d'ells i un fracàs per altres; però només es pot considera que funcionarà tot correctament si la migració es fa de forma correcta per a tots.

7.1 Personal implicat

La posada en marxa es refereix a tots els processos entre l'acabament de la implementació i el funcionament real de l'aplicació. Aquests processos es poden dividir en tres parts:

- **Formació:** cal formar a tots els usuaris que utilitzaran la nova aplicació d'una forma o altra. En aquest procés el personal implicat serà el departament d'informàtica de l'empresa com a dadors de la formació, i els departaments d'administració, comercial, logística i direcció com a alumnes d'aquesta formació.
- **Procés de paral·lel:** correspon al temps després de la formació on els usuaris comencen a utilitzar el nou programa, sense deixar d'utilitzar també les ferramentes anteriors. Aquest procés és el més costós des del punt de vista dels usuaris finals i per tant el que més trastoca el funcionament normal de l'empresa. Els personal implicat a aquesta fase és el mateix que el definit a la fase de formació.
- **Pas a funcionament:** Cal tindre en compte que, per la forma triada de desenvolupament del projecte, l'entrada del funcionament es farà per fases d'implementació, i per tant a les primeres fases no significarà deixar a banda totes les aplicacions anterior, però si d'algunes i el canvi de procediments interns. El personal implicat en aquesta fase crítica és el mateix que a les anteriors, però posant una major pressió sobre els usuaris finals i sobretot en els caps de cada departament que hauran de vetllar pel correcte funcionament de tots aquests nous processos.

7.2 Pla de formació

La formació dels usuaris finals es durà a terme en tres fases diferenciades:

- **Formació a superusuaris:** es farà per part de l'equip de programació i implementació cap als líders de cada departament de l'empresa implicats: administració, logística, comercial i direcció. Aquesta formació es farà durant 1 setmana, amb 2h al dia durant 3 dies alterns, per a limitar

l'impacte sobre el normal funcionament de l'empresa. L'objectiu és que els caps de departament puguin ajudar a transmetre el coneixement a la resta dels seus equips, i que puguin resoldre les seues preguntes directament, així com que els caps de departament tinguen una visió global del funcionament de l'ERP.

- **Formació a departaments:** amb l'ajuda dels superusuaris ja formats, es donarà la formació específica de cada departament, també durant 2 setmanes, amb 2h al dia i 3 dies alterns, però en aquest cas per departaments. L'objectiu d'aquesta formació és ensenyar a cada departament la part específica que li correspon de la ferramenta.
- Per últim, es faran 3 dies de **formació grupal** de tots els implicats, en les 3 primeres setmanes del paral·lel, per a dubtes i propostes. L'objectiu d'aquesta formació grupal és respondre a qualsevol pregunta o problema sorgit del canvi, així com generar sinergies entre tots els departaments implicats.

A banda de la formació personal, es donarà tota la documentació i manuals realitzats als usuaris per a que els puguin consultar en qualsevol moment. S'estudia també la possibilitat de gravar en vídeo la formació als superusuaris per a poder utilitzar en un futur si existeix canvis de personal. El pla de formació es repetirà si existeixen noves incorporacions de treballadors als departaments implicats en un futur.

	Setembre		Octubre		Novembre		Desembre	
	1 ^a quinzena	2 ^a quinzena	1 ^a quinzena	2 ^a quinzena	1 ^a quinzena	2 ^a quinzena	1 ^a quinzena	2 ^a quinzena
Formació								
Paral·lel								
Pas a producció								

Figura 21: Planificació de posada en marxa

A la figura 21 es pot trobar la planificació per a octubre, novembre i desembre de les tres fases del procés de formació, en el millor cas.

7.3 Paral·lel i canvi gradual

Com hem vist a la definició del personal implicat, el paral·lel és el procés on no es deixaran d'utilitzar encara els procediments i aplicacions anteriors, però es començaran a utilitzar en paral·lel els nous. L'objectiu d'això és localitzar qualsevol problema que no s'haja tingut en compte en la fase de disseny i implementació i solucionar-los abans que l'ús siga de producció.

Durant la fase de paral·lel, el departament d'informàtica estarà a disposició de cadascun dels altres departaments implicats per a trobar qualsevol problema que puga sorgir i resoldre-lo en el moment.

Es calcula que caldrà com a mínim 2 mesos de paral·lel des del final de la formació per afinar tots els processos i assegurar que tot el personal coneix els passos a seguir. No s'acabarà el paral·lel en cap cas abans que s'assegure que tots els processos clau de l'empresa estan perfectament representats als nous procediments i això quede reflectit en el funcionament del nou ERP implementat.

A la planificació de la posada en marxa que s'indica el mínim i màxim assignat a aquesta fase, que finalitzarà en el millor cas el mes de novembre, però que es podria ampliar fins a desembre si fora necessari.

7.4 Pas a funcionament en producció

Una vegada finalitzada la fase de paral·lel, es deixaran d'utilitzar els procediments antics i d'introduir informació nova a les bases de dades de les aplicacions anteriorment utilitzades que corresponguen a aquests nous procediments. Les aplicacions continuaran siguent utilitzades i mantingudes mentre hi haja necessitat de consulta en el dia a dia de dades del cycle econòmic anterior.

Està previst que el paral·lel comence en setembre perquè el cycle comptable a ICSA correspon als mesos entre octubre de l'any anterior i setembre del següent, de manera que el paral·lel ens ajudarà a que totes les dades dels procediments implementats del cycle en curs estiguen ja introduïdes a la plataforma per al començament del funcionament en producció.

Tal com apareix a la planificació de la figura 21, està previst que la posada en funcionament d'aquesta primera fase d'implementació siga en desembre o gener, depenent de l'evolució del paral·lel i els problemes trobats i solucionats. Aquesta planificació permet que les dades de l'últim trimestre 2020-2021 puguen ser consultables amb l'aplicació anterior durant el paral·lel i la primera posada en producció, i que a partir de gener ja el trimestre anterior estiga complet dins del nou ERP, limitant d'aquesta manera la necessitat d'accés a dades amb les aplicacions anteriors.

7.5 Gestió del canvi

La gestió del canvi és una **fase transversal** de tot el projecte, on assegurem que la solució adoptada serà acceptada per tota l'organització. Cal tindre en compte que en qualsevol empresa existeix una resistència al canvi, ja siga per interessos confrontats o per comoditat dels treballadors amb la situació prèvia.



Figura 22: Transversalitat de la gestió del canvi

La part més important de la gestió del canvi és la corresponent a la formació dels usuaris finals, que ja ha sigut explicada a l'apartat del pla de formació, però en ocasions aquesta resistència pot vindre de persones o entitats que no utilitzaran de forma directa el programari.

Per a trencar aquesta possible resistència, s'han dut a terme durant tot el projecte les següents accions:

- Com hem vist al capítol 2. Antecedents, abans de començar el projecte s'han fet **entrevistes personalitzades** a tots els departaments involucrats, de forma pública i notòria, per a obtenir les necessitats concretes del projecte. Aquestes necessitats s'han través a tots els departaments i a Direcció de l'empresa.
- També com apareix a la secció 2.3 Alternatives proposades, s'ha proposat **diverses alternatives** per evitar l'obligació de realitzar el projecte, amb els costos associats de cada alternativa en el cas que els hi haguera. Aquests costos s'han estret a través de comercials externs que ofereixen cadascuna d'aquestes alternatives.
- La decisió final que apareix a la secció 2.5 Presa de decisió ha sigut emesa per direcció de l'empresa, però també **notificada i consensuada** amb la resta de departaments.
- Una vegada començada la fase de disseny explicat al capítol 4. Disseny de l'aplicació, i després d'avaluar el volum general i per tant el cost del projecte, es consensua amb direcció i resta de departaments fer les **modificacions oportunes** explicades en el capítol 3. Limitacions sobrevingudes al projecte.
- El pla de formació de l'apartat 7.2 Pla de formació s'ha organitzat tenint en compte la **càrrega de treball** normal de tots els implicats, i amb l'objectiu de que aquesta siga el menys radical possible per a que no afecte al normal funcionament de tots els departaments.

Amb tot l'exposat, s'espera que no hi haja cap resistència important per part de cap de les parts en acceptar els canvis proposats a aquest projecte.

8. Conclusions

8.1 Objectius assolits

El primer objectiu assolit d'aquest projecte és la seua realització. En el moment de la presentació d'aquesta memòria ja s'han començat els treballs d'implementació tal com apareix a la planificació del capítol 6. Pla d'implementació. Direcció de l'empresa ha confiat de forma directa en la finalització com a mínim de la primera fase del projecte, i el nou ERP per a l'empresa serà una realitat.

Sobre els objectius d'aquest treball de fi de grau, fem un recopilatori dels inicials i verifiquem el seu compliment:

1. **Documentar** tots els processos que realitza l'ERP actual, així com la seua utilització actual, per assegurar que queden dins del disseny de la nova ferramenta en el cas de ser utilitzats

Encara que no s'ha pogut documentar tots els procediments, perquè molts d'ells són d'ús molt específic i el límit temporal del projecte ha sigut molt important, si que s'ha fet una recopilació i s'han documentat els bàsics i de procés clau per a poder fer el disseny. Al capítol 2. Antecedents d'aquesta memòria tenim un resum dels mètodes utilitzats fer la seua detecció.

2. **Documentar** tots els processos de compra-venda, facturació i tresoreria d'ICSA per a localitzar processos que no estigueren contemplats en l'ERP actual.

Es van documentar alguns processos no contemplats a l'antic ERP, però es va decidir deixar-los fora d'aquesta primera fase d'implementació per no formar part del procés clau i les limitacions temporals del projecte.

3. **Disseny** de la part interna del nou aplicatiu ERP: processos, emmagatzemament i consulta de dades. Integracions verticals i horitzontals. Compliment amb els processos antics i nous localitzats.

Tots aquests processos i estructures de dades han quedat dissenyats i documentats per a la seua implementació. En la secció 4.2 Disseny general d'aquesta memòria apareix un resum dels més importants.

4. **Disseny** bàsic de les interfícies d'usuari a utilitzar per usuaris i administradors del nou aplicatiu. Enllaç de les interfícies amb els processos.

S'ha realitzat el disseny de les interfícies bàsiques, i els procediments d'enllaç amb les accions de l'ERP. En la secció 4.5 Disseny del component client d'aquesta memòria hi ha un resum dels resultats obtinguts.

5. **Proves d'interfícies i processos amb els usuaris clau de l'aplicatiu.**

S'han fet les proves previstes amb 6 usuaris claus, de les quals s'han estret conclusions que han sigut útils per a la modificació del disseny per fer-lo més útil i amigable per als usuaris, i per a preparar els plans de formació i posada en marxa de l'aplicació. Tenim un resum de la metodologia utilitzada i conclusions trobades al capítol 5. Proves amb usuaris clau d'aquesta memòria.

6. **Modificacions en els dissenys després de les proves amb usuaris clau.**

Com s'indica al punt 5, no només ha sigut útil les proves per a fer modificacions de disseny, sinó també per fer més complet el pla de formació i posada en marxa, amb requisits més propers als usuaris finals. A la secció 5.6 Conclusions de les proves tenim les modificacions introduïdes després de les proves.

7. **Definició del pla de treball per a *implementar* els dissenys realitzats.**

S'ha definit un pla detallat d'implementació amb la seua planificació per a la primera fase d'aquesta; en el capítol 6. Pla d'implementació d'aquesta memòria tenim un resum d'aquest pla.

8. **Documentació inicial de la implementació i pla de *posada en funcionament*.**

S'ha preparat aquesta documentació inicial de la implementació, i s'ha preparat un pla de posada en funcionament, en tres fases: formació, paral·lel i posada en producció. Un resum d'aquestes fases es troba al capítol 7. Posada en marxa i gestió del canvi d'aquesta memòria.

Amb tot això, podem afirmar que s'han assolit la gran majoria dels objectius inicials d'aquest projecte, encara que queda marge obert per a continuar-lo en successives fases de disseny, implementació i posada en marxa futures.

Així mateix, es van definir uns objectius específics de la fase de disseny, que són els següents:

1. *Utilitzar de forma general els **coneixements** actuals de l'equip de treball.*

Encara que s'han introduït noves tecnologies actualment desconegudes per a l'equip d'implementació, els llenguatges utilitzats i les metodologies són conegudes per tots, de manera que s'ha reaprofitat aquests coneixements per reduir el temps d'aclimatació de l'equip.

2. *Utilitzar **tecnologies** modernes, actualitzades a la situació actual, i que puguen ser fàcilment mantingudes durant un mínim de 10 anys.*

Com hem comentat en el punt anterior, s'han introduït noves tecnologies i s'han actualitzat altres per a estar preparats per al futur.

3. *Els components han de ser **reutilitzables** al màxim.*

Dels 3 components proposats al disseny, 2 d'ells són molt reutilitzables: la base y estructura de dades al 100% i el component client al 40%.

4. ***Externalitzar** al màxim la gestió d'elements comuns, a través de servicis laaS i PaaS per a minimitzar la infraestructura interna.*

També s'ha aconseguit aquest objectiu, utilitzant plataformes externes en tot els casos que faciliten el manteniment i gestió de servidors i serveis.

8.2 Coneixements útils per a nous projectes

Durant la realització d'aquest projecte s'han trobat algunes limitacions important que cal tindre en compte per a la realització de futurs projectes pareguts, o de les següents fases dels projectes:

- Tindre en compte les **limitacions temporals**: l'abast del projecte va ser pensat massa ampli per a el temps del que es disposava realment. No només en la fase de disseny, sinó també en la fase d'implementació. Direcció de l'empresa va acceptar dur endavant el projecte i plasmar-lo en una aplicació real, però a canvi de limitar el seu abast i per tant tenir-lo preparat en un temps limitat. Per això va ser necessari reduir els objectius. Per a futurs projectes, és més útil començar en format reduït i anar ampliant-lo si hi ha disponibilitat.
- Tindre en compte les **limitacions del personal** de l'empresa: El pla de recaptació de dades i formació anava a ser més ambiciós que el planificat, però sempre cal tindre en compte les limitacions temporals del personal de l'empresa, ja que han de continuar fent el treball regular a banda de l'extra introduït per l'avanç del projecte.
- Confiar en **direcció de l'empresa**: És qui prendrà les decisions finals, i qui farà que el projecte es plasme a la realitat o no. És bàsica la esponsorització de l'empresa per a complir qualsevol objectiu.
- Comptar amb els **usuaris finals** en totes les fases: No només en la fase de disseny, sinó també anar mostrant els avanços als usuaris implicats en tot moment per evitar modificacions que afecten a tercers usuaris.
- Canvis de **metodologia** a àgil: En un començament aquest projecte va ser concebut utilitzant metodologies conservadores d'implementació de programari, sobretot basat en procediments en cascada. Quan es va veure que era impossible fer tot el projecte en el temps oferit per direcció de l'empresa, es va tindre que desmembrar en diferents fases, de manera que aquestes metodologies ja no eren útils. Per això es va canviar la forma del desenvolupaments a mètodes àgils, que permeten tin-

dre productes intermedis útils i continuar el desenvolupament en un futur.

8.3 Continuació del projecte

Com s'ha comentat durant tota aquesta memòria, el projecte original va ser dividit en diverses fases amb resultats utilitzables per l'empresa en cadascuna d'elles. En aquesta primera fase només s'han implementat els procediments clau de la gestió de l'empresa, i els lligams entre ofertes, vendes, compres, gestió d'estoc, facturació i comptabilitat.

Això deixa molt de marge per a futurs projectes d'ampliació de l'ERP, evolucionant amb tots els mòduls que no s'han implementat en aquesta ocasió.

En especial, una línia de treball que s'ha deixat inconclusa i em pareix especialment interessant són els indicadors i el **Business Intelligence** (BI). Nevis tindrà totes les dades de l'empresa, i dona l'opció d'utilitzar-les des de terceres aplicacions per a que els usuaris puguin prendre conclusions, però no gestiona de forma directa els indicadors ni fa cap tipus de projeccions futures amb les dades emmagatzemades.

Per l'estructura de dades de nevis, pareix molt raonable proposar construir directament sobre ell un mòdul d'explotació de dades i BI, que permeti fer aproximacions no només de facturacions futures, sinó també d'estocs òptims, sectors calents per a fer campanyes de venda i gestió automàtica d'impagaments.

En especial, pense que seria un molt bon treball fi de carrera independent per a una carrera d'economia, amb prou de contingut com a per a tindre part d'investigació per a l'explotació de dades així com ús directe en aquest ERP i la seua implementació.

Per tot això, i tenint en compte que el primer semestre del curs 2021-2022 hauré de fer el TFG del grau de ADE, presentaré les reflexions de continuació d'aquest projecte com a proposta, i si és acceptada es **continuarà amb la investigació d'explotació de dades i el projecte d'implementació d'un mòdul BI per a l'ERP descrit a aquesta memòria.**

9. Glossari

- **Acreditació:** Procés voluntari amb el que una organització pot mesurar la qualitat dels seus serveis o productes, i el rendiment d'aquests davant d'estàndards coneguts.
- **API:** Interfície de Programació d'Aplicacions, component d'un programa informàtic que permet que interaccione amb altres programes diferents.
- **API REST:** Arquitectura d'APIs d'intercomunicació pensada per utilitzar en aplicacions web.
- **Atribut:** Part mínima de la informació d'un element, que conté un nom i un contingut.
- **Back-end:** Components necessaris d'una aplicació que no interaccionen directament amb els usuaris finals.
- **Contenedor de programari:** Paquet de programari amb totes les llibreries necessàries per al seu funcionament, de forma independent a la màquina on finalment s'execute.
- **Cost d'oportunitat:** Quantitat monetària imaginària que es deixa d'ingressar per utilitzar els recursos en un projecte alternatiu.
- **Desenvolupament intern:** Desenvolupament que fa una empresa amb els recursos amb que compta, sense cap ajuda externa.
- **Desplegament:** Posada en funcionament d'un programari en producció. En contenidors, es diu desplegament a posar el contenidor en una màquina concreta per al seu funcionament.
- **Disseny:** En programari, definició teòrica del funcionament esperat de l'aplicació, així com la seua interacció amb els usuaris i forma d'emmagatzemament de dades.
- **Django:** Framework de desenvolupament de programari web de codi obert i escrit en python.
- **Docker:** Sistema d'automatització de desplegament d'aplicacions dins de contenidors de programari.
- **DocumentDB:** Gestor de base de dades ofert per Amazon que és compatible amb MongoDB i es gestiona com SaaS.
- **ERP:** Planificador de Recursos Empresarials, sistema que integra totes les dades i processos d'una organització de forma unificada.
- **Explorador web:** Programari capaç de entendre els llenguatges necessaris per a navegar en pàgines web.
- **Flask:** Framework de desenvolupament de programari web de codi obert i escrit en python.
- **Framework:** Entorn de treball que dóna les primitives bàsiques per a generar programari més eficient.
- **Front-end:** Components necessaris d'un programari que interactuen directament amb els usuaris finals.
- **HTTP:** Protocol d'intercanvi de pàgines d'hipertext, el protocol que s'utilitza per a navegar entre pàgines web.
- **IaaS:** Infraestructura com un Servei, el model més bàsic de la computació en núvol. Ofereix màquines virtuals completes on l'usuari pot treballar com si foren màquines físiques.
- **ICSA:** Instrumentos Científicos S.A.

- **Integració de programari:** Procés a través del qual s'aconsegueix l'intercanvi d'informació entre programari que originalment són independents.
- **Interfície d'usuari:** Part del programari de qualsevol aplicació amb el que interactua directament l'usuari final.
- **JavaScript:** Llenguatge de programació lleuger i interpretat, utilitzat massivament en la programació web. L'estàndard de JavaScript s'anomena ECMAScript.
- **JSON:** Estàndard obert basat en text que té com a objectiu l'intercanvi de dades entre programaris, mantenint aquestes dades en format llegible per humans.
- **Kubernetes:** Sistema d'orquestració de contenidors que ajuda al desplegament, escalat i gestió de programari.
- **LIMS:** Sistema de Gestió de la Informació de Laboratori, programari integral que gestiona els processos específics d'un laboratori.
- **Màquina virtual:** Peça de programari que abstrau el maquinari d'un computador físic per a que aplicacions i sistemes operatius no utilitzen de forma directa el maquinari real.
- **Modularitat de programari:** Capacitat d'alguns programari on l'aplicació només gestiona el funcionament bàsic i la resta es donada per mòduls intercanviables i no necessaris.
- **NevisLIMS:** Programari LIMS desenvolupat i comercialitzat per ICSA.
- **Núvol informàtic:** Forma de computació fonamentada en internet on els recursos compartits es proporcionen a la carta com a serveis.
- **Objecte:** En programari, un objecte és una estructura de dades encapsulada i independent on només part de la seua informació interna és accessible des de fora.
- **Open source:** Enforcament de disseny, desenvolupament i distribució de programari on els usuaris no només obtenen el resultat final sinó tot el codi font necessari per a poder arribar a aquest resultat.
- **ORM:** Tècnica de programació que permet convertir dades de llenguatges de programació orientats a objectes a bases de dades relacionals.
- **PaaS:** Plataforma com a Servei, és una categoria de serveis al núvol on el que s'ofereix és una plataforma completa, però sense accés real al maquinari d'aquesta.
- **PAC:** Prova d'Avaluació Contínua.
- **Pop-up:** Ferramenta del programari on es mostra o consulta informació a l'usuari com a una finestra sobre la de treball, de manera que la informació original mostrada no canvia.
- **Procediment:** Mètode definit i documentat per a tramitar o executar una acció concreta, o per aconseguir un producte concret.
- **Python:** Llenguatge de programació de codi obert, d'alt nivell i propòsit general.
- **RoR:** Ruby on Rails, entorn de treball de codi obert per a realitzar aplicacions web escrit en el llenguatge de programació Ruby.
- **Ruby:** Llenguatge de programació interpretat. de propòsit general, orientat a objectes.
- **SaaS:** Programari com a servei, és un model de distribució del programari, on aquest està allotjat i s'executa directament des del núvol, a través d'Internet.

- **Script:** Peça de programari independent amb propòsit específic que s'utilitza només amb una finalitat, en especial per a transformar estructures de dades o migrar-les de un lloc a altre.
- **SGBD:** Sistema de Gestió de Bases de Dades, conjunt de programari que facilita la gestió de dades dins d'un sistema informàtic.
- **Trigger:** Disparador, procediment executat quan es compleix una condició específica establerta.
- **Usuari clau:** En programari, usuari model que utilitza de forma extensiva les funcionalitats del programari.

10. Bibliografia

- [1] Robert Clarisó Viladrosa, José Ramón Rodríguez; *Treball final de màster*, Material docent de la UOC, FUOC, 2013.
- [2] Grant Norris, James R. Hurley, Kenneth M. Hartley et al; *E-Business and ERP: Transforming the Enterprise*, John Wiley & Sons, 2000.
- [3] Robert C. Martin, *Desarrollo ágil esencial: Vuelta a las raíces*, ANAYA, 2020.
- [4] Miguel Grinberg; *Flask Web Development 2e: Developing Web Applications with Python*, 2ª edició, O'Reilly, 2018.
- [5] Ernest Abadal Falgueras, Lluís Codina Bonilla; *Bases de datos documentales: características, funciones y método*, Síntesis, 2005.
- [6] David Flanagan; *JavaScript. La Guía Definitiva*, ANAYA/O'Reilly, 2007.
- [7] Marcos Antonio Catalán Vega; *Metodologías de evaluación de Interfaces Gráficas de Usuario*, accés web: http://eprints.rclis.org/6732/1/Metodologias_de_evaluaci%C3%B3n_de_interfaces_graficas_de_usuario.pdf.
- [8] Kent Sandoe, Gail Corbitt, Raymond Boykin; *Enterprise Integration*, John Wiley & Sons, 2001.