

Integració d'aplicacions amb SAP/R3 en una indústria d'Arts Gràfiques



Alzamora | GROUP

Sergi Valls Vilà
Grau d'Enginyeria Informàtica
15/06/2015
Consultor: **Humi Guill Fuster**



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

Títol del treball:	<i>Integració d'aplicacions amb SAP/R3 en una indústria d'Arts Gràfiques</i>
Nom de l'autor:	<i>Sergi Valls Vilà</i>
Nom del consultor:	<i>Humi Guill Fuster</i>
Data de lliurament (mm/aaaa):	<i>06/2015</i>
Àrea del Treball Final:	<i>ERP</i>
Titulació:	<i>Grau d'Enginyeria Informàtica</i>
Resum del Treball (màxim 250 paraules):	
<p>El Treball tracta sobre com s'ha realitzat la integració de diversos sistemes d'informació amb un sistema SAP ERP, en una reconeguda empresa d'arts gràfiques catalana. En aquest document, es fa una explicació de l'estructura a nivell dels SI que conformen el <i>core business</i> de l'organització i posteriorment es fa l'anàlisi de com s'ha realitzat la integració d'aquest nucli informàtic amb el nou ERP de l'empresa, valorant aspectes tècnics així com conceptuals. Es fa un repàs també a cadascuna de les integracions centrant-nos en els aspectes més tècnics del procés, donat que per cada SI integrat, hi ha una metodologia i una tècnica diferent usada.</p>	

Abstract (in English, 250 words or less):

This TFG talks about the process about integration of some information systems with the SAP ERP, in a known printing company. In this document, we explain the core business structure held by the different information systems of the organisation, and later, we analyse the process of integration those information systems with the new ERP, valuing technical and conceptual aspects.

Paraules clau (entre 4 i 8):

SAP, ERP, ABAP, JAVA

Índex

Índex	5
Agraïments	8
1. Introducció.....	10
1.1. Context i justificació del Treball	10
1.2. Objectius generals i específics.....	10
1.3. Objectius generals	11
1.4. Objectius específics	11
1.5. Enfocament i mètode seguit	11
1.6. Planificació amb fites i temporització.....	11
1.7. Breu sumari de productes obtinguts.....	14
1.8. Breu descripció dels altres capítols de la memòria	14
2. Presentació d'Alzamora Group.....	15
2.1. Visió global dels SI del grup Alzamora.....	15
2.1.1. Antecedents.....	17
2.1.2. SAP ERP.....	18
2.2. Detall dels SI i aplicacions principals d'Alzamora	19
2.2.1. Sistema de control de producció en planta (CPP).....	19
2.2.2. Sistema de control de presència del personal (SCP):	20
2.2.3. Sistema de gestió de magatzems (SGM):	21
2.2.4. Aplicació per la generació de banderoles identificatives de palet (GDB):	22
2.2.5. Aplicació per la cerca d'arxius de preimpresió (CDP):	23
2.2.6. Sistema pel disseny de maquetes (SDM):	24
2.2.7. Sistema de preimpresió (SDP):.....	26
2.2.8. Aplicació per l'emalatge i etiquetat (SEE):	28
3. Presentació de les integracions a realitzar	30

3.1.	Eines per interconnectar sistemes	31
4.	Definició de les integracions.....	32
4.1.	Integració del sistema de control de producció en planta (CPP)	32
4.1.1.	Procés de definició de les connexions.....	33
4.1.2.	Definició de les RFC	34
4.1.2.1.	RFC ZQPLANT_CLIENTE_OF: Obtenció del client donada una Ordre de Fabricació.....	34
4.1.2.1.	RFC ZQPLANT_ORDEN: Obtenció de dades de la ordre de fabricació.....	36
4.1.3.	Exemples d'ús de les RFC.....	39
4.2.	Integració del sistema de control de presència del personal (SCP)	41
4.2.1.	Procés de definició de les connexions.....	42
4.2.2.	Definició de la RFC.....	42
4.2.2.1.	RFC ZQPRESENT_OPERARIS: manteniment del mestre d'empleats de SAP.....	42
4.3.	Integració de l'aplicació per la generació de banderoles identificatives de palet (GDB).....	43
4.3.1.	Procés de definició de les connexions.....	45
4.3.2.	Exemples de crida del GDB.....	45
4.4.	Integració de l'aplicació per la cerca d'arxius de preimpressió (CDP).....	49
4.4.1.	Procés de definició de les connexions.....	50
4.4.2.	Definició de la RFC.....	50
4.4.2.1.	ZCERCA_ARXIUS: cerca informació de materials i clients	50
4.4.3.	Exemples de crida del CDP	53
4.5.	Integració de l'aplicació per l'embalatge i etiquetat (SEE).....	58
4.5.1.	Procés de definició de les connexions.....	59
4.5.2.	Definició de la RFC.....	59
4.5.2.1.	ZMACSA_LLEGIR_NOM_CLI: obtenir el nom del client.....	60
4.5.2.2.	ZMACSA_LLEGIR_CLI_ART: obtenir la descripció del material i del material del client.....	61
4.5.3.	Exemples de crida del programa	63
4.6.	Integració del sistema de magatzems (SGM).....	67

4.6.1.	Procés de definició de les connexions.....	67
4.6.2.	Exemples de crida del SGM	69
4.6.2.1.	Petició de sortida de material del magatzem.....	69
4.6.2.2.	Confirmació de sortida de material del magatzem	71
4.6.2.3.	Alta de referències de material al SGM.....	73
5.	Millores futures	75
6.	Conclusions.....	76
7.	Annex.....	78
8.	Bibliografia.....	86

Agraïments

En primer lloc, m'agradaria agrair als meus pares la oportunitat que em van donar al seu dia d'estudiar la primera carrera, i el seu suport i ànims per cursar la segona. Evidentment, sense ells jo no estaria escrivint aquestes línies! Gràcies per ensenyar-me el valor de l'esforç i la perseverança.

En segon lloc m'agradaria donar les gràcies a l'empresa Alzamora Group, en especial a l'Anna Alzamora, per la oportunitat que se'm va donar de participar en el projecte d'implantació del sistema SAP, i per els coneixements adquirits i sobretot l'experiència aconseguida en el procés; sens dubte, Alzamora és i ha estat una de les millors escoles! L'aventura viscuda no té preu i menys si s'ha viscut en un entorn on els resultats i, en especial els companys, són fantàstics.

Per últim, a la Sara, la meva dona, per el seu suport, comprensió, ànims i amor. I sens dubte per haver portat al món a la nostre Bruna! Sou la força que fa moure el món!

Donada l'expressa petició d'Alzamora Group, no es publicaran dades privades de l'empresa, ni el nom de les solucions de *software* dels diferents Sistemes d'Informació (els anomenarem de forma abreviada per la funció que tenen). Única i exclusivament ens centrarem en la part tècnica de la solució, evitant així la publicació d'informació interna, com dades de clients o dades internes de producció

1. Introducció

1.1. Context i justificació del Treball

Aquest Treball final de Grau (en endavant TFG) està basat en la pròpia experiència de migració de un ERP sobre AS400 a un sistema SAP R3. En aquest sentit, aquest TFG tractarà específicament sobre la part de les integracions de certs sistemes d'informació (en endavant SI) de l'empresa d'arts gràfiques on treballa (Alzamora Group) amb el sistema SAP, donat que no tots els SI que conformen l'organització es van poder migrar a SAP, ja sigui per la incompatibilitat de solucions, per qüestions tècniques o per qüestions de simplicitat.

El resultat final del treball que obtindrem serà la integració (total o parcial) dels SI claus d'Alzamora Group, amb el SAP R3, veient els aspectes tècnics de la solució i com les estratègies adoptades han permès aquesta integració sense haver de recórrer a modificacions molt extenses del codi estàndard de SAP ni haver de fer servir mòduls de SAP específics per aquestes tasques.

En qualsevol cas, aquest document no pretén explicar què és el SAP i ni molt menys aprofundir en els seus mòduls; creiem que ja hi ha suficient literatura al respecte i per tant, si es vol aprofundir en aquests temes ja es té documentació disponible en d'altres circuits. Aquest TFG no pretén ensenyar a programar sobre SAP ni sobre cap altre llenguatge usat per realitzar aquestes integracions. El TFG pretén mostrar una problemàtica que vivim molts professionals que treballem amb SAP i d'altres SI heterogenis en el sentit de les integracions de SI, donat que moltes vegades cal que les illes d'informació de l'empresa es comuniquin entre elles, i moltes vegades no disposem de les eines necessàries per aconseguir aquests objectius i hem de recórrer a diferents alternatives tecnològiques per tal d'aconseguir-ho.

Per últim, som conscients que les integracions s'haguessin pogut fer millor o diferent, però en el TFG les explicarem i mostrarem tal com nosaltres vam decidir fer-ho donat que és un cas pràctic, específic i particular de la nostra organització, per tant, les solucions adoptades podrien no ser compatibles i/o viables en d'altres organitzacions.

1.2. Objectius generals i específics

A grans trets, els principals objectius del TFG són mostrar la problemàtica existent a la nostra empresa quan es va decidir instal·lar un sistema SAP R3, havent d'integrar aplicacions i SI de tercers que no tenien res a veure amb SAP, amb aquest nou sistema, i mostrar les solucions adoptades per tal d'aconseguir aquestes integracions.

1.3. Objectius generals

Els objectius generals del TFG són:

- Mostrar la naturalesa dels SI actuals a l'empresa i les seves característiques generals i particulars
- Definir les integracions a realitzar amb d'altres SI
- Descriure els problemes principals que ens trobem per aconseguir-ho
- Aportar i descriure les solucions tècniques a aquestes integracions

1.4. Objectius específics

En quant als objectius específics tenim:

- Explicar les característiques tècniques de les eines utilitzades en les integracions: RFC's, connexions a BD directament, etc.
- Mostrar amb exemples reals de programació, les solucions d'integració creades: llenguatge JAVA, ABAP, Delphi, etc.

1.5. Enfocament i mètode seguit

El treball tracta directament de les solucions tècniques adoptades per integrar diversos SI amb SAP, per tant, el que hem fet és mostrar aquestes solucions.

Creiem que la integració dels SI fent servir programes a mida i modificant en certs punts el codi estàndard de SAP, ha estat la correcte tenint en compte certes premisses inicials:

- No volíem fer servir el mòdul de SAP per integrar aplicacions, per qüestions de preu, cost en recursos i complexitat.
- Teníem el temps limitat.
- Era més fàcil adaptar la resta dels SI a SAP que viceversa.

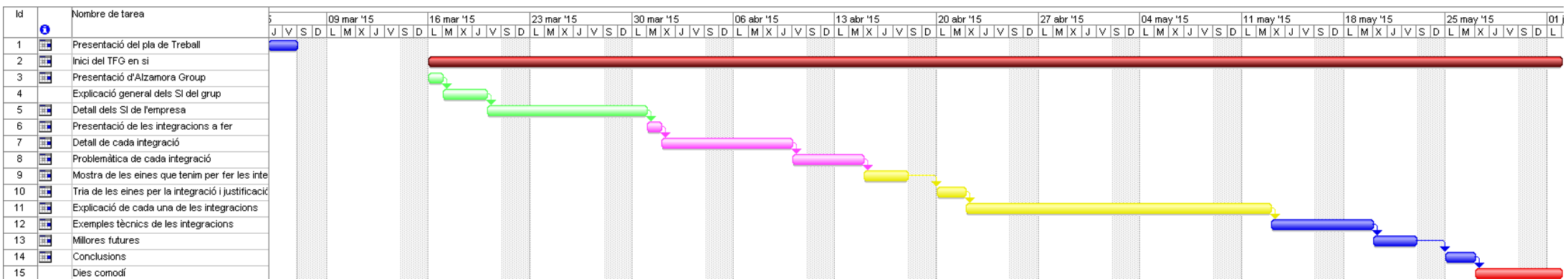
1.6. Planificació amb fites i temporització

La planificació del TFG l'hem organitzat en 4 grans fites:

- Visió general dels SI de l'empresa
- Definició de les integracions a fer
- Tria de les eines per tal de fer aquestes integracions
- Explicació tècnica (amb exemples) de les integracions realitzades

Cada bloc estarà format per fites més petites que consistiran en el detall de cada un dels blocs arrel. A continuació mostrarem la temporalització del TFG:

Presentació del pla de Treball	2 dies?	jue 05/03/15	vie 06/03/15
Inici del TFG en si	56 dies?	lun 16/03/15	lun 01/06/15
Presentació d'Alzamora Group	1 dia?	lun 16/03/15	lun 16/03/15
Explicació general dels SI del grup	3 dies	mar 17/03/15	jue 19/03/15
Detall dels SI de l'empresa	7 dies	vie 20/03/15	lun 30/03/15
Presentació de les integracions a fer	1 dia	mar 31/03/15	mar 31/03/15
Detall de cada integració	7 dies	mié 01/04/15	jue 09/04/15
Problemàtica de cada integració	3 dies	vie 10/04/15	mar 14/04/15
Mostra de les eines que tenim per fer les	3 dies	mié 15/04/15	vie 17/04/15
Tria de les eines per la integració i justific:	2 dies	lun 20/04/15	mar 21/04/15
Explicació de cada una de les integracion	15 dies	mié 22/04/15	mar 12/05/15
Exemples tècnics de les integracions	5 dies	mié 13/05/15	mar 19/05/15
Millores futures	3 dies	mié 20/05/15	vie 22/05/15
Conclusions	2 dies	lun 25/05/15	mar 26/05/15
Dies comodí	4 dies	mié 27/05/15	lun 01/06/15



1.7. Breu sumari de productes obtinguts

Com ja hem avançat anteriorment, el que obtindrem amb aquest TFG és les solucions tècniques adoptades per a realitzar la integració dels SI d'Alzamora Group amb el producte SAP. No s'ha obtingut un producte en concret ni cap *software* concret, sinó que hem creat varies interfases per la interconnexió de sistemes heterogenis.

1.8. Breu descripció dels altres capítols de la memòria

Aquest TFG estarà estructurat de forma que inicialment farem una introducció de l'estructura dels SI de l'empresa i del propi sistema SAP. Un cop sabem en quin entorn ens movem, mostrarem quines eines ens proporciona SAP a fi d'aconseguir aquests objectius, fent una repassada a les alternatives. Per últim, mostrarem exemples reals de part d'aquestes integracions, veient així la forma que ho hem resolt i justificant aquestes decisions.

2. Presentació d'Alzamora Group

Com ja s'ha avançat, aquest TFG s'ha dut a terme dins l'empresa Alzamora Group (fig 1 a 3, imatges de l'empresa). Aquesta és una empresa familiar d'arts gràfiques de la comarca de la Garrotxa amb més de 100 anys d'història, amb més de 150 empleats, molt tecnificada i amb els últims avenços tecnològics en matèria d'impressores offset i processos de preimpresió. L'empresa es dedica a la fabricació de packaging (envasos de cartró) per varietat de sectors: alimentació, cosmètica, begudes, farmàcia, etc. Es pot dir que és una de les empreses líder en el seu sector, tant per facturació anual com per volum de fabricació.



Fig.1: Vista de les instal·lacions



Fig.2: Vista de les instal·lacions



Fig.3: Vista de les instal·lacions

www.alzamora.com

2.1. Visió global dels SI del grup Alzamora

Al ser una empresa moderna i tecnificada, els SI de l'organització acompanyen en tot moment aquestes característiques, estant plenament alienats amb l'objectiu i la missió del

negoci. A continuació (fig. 4) es mostra un quadre a mode de resum dels diferents tipus de SI de que disposa l'empresa:



Fig. 4 – Visió global dels SI

Com es pot observar d'entrada, el ventall de SI de que es disposa és molt ampli i heterogeni, barrejant tecnologies i metodologies que no tenen massa en comú, pivotant totes elles sobre el nucli del SI que és SAP.

Fent una visió general i sense entrar en detall degut a la petició expressa per part d'Alzamora de no fer-ho, podem dir que es disposa de:

- Sistema SAP ERP
- Sistemes de col·laboració Exchange
- Sistemes Windows Server
- SQL Server, Oracle i Microsoft Dynamics
- Aplicacions de *prepress* (orientades a les arts gràfiques)

- Aplicacions comercials per el control de la producció, embalatge automàtic, gestió de magatzems, etc.

Els sistemes podem dir que són una barreja entre aplicacions comercials conegudes àmpliament (productes Microsoft) i aplicacions d'empreses més petites dedicades a funcions molt concretes (control de producció en planta, gestió de normes de qualitat, control de magatzem, etc.).

Aquestes aplicacions corren sobre maquinari principalment IBM, tot i que també es disposa de maquinari HP i DELL. A nivell de comunicacions, es confia la gestió sobre maquinari CISCO. Respecte el *hardware*, no entrarem més detalladament donat que no és l'objectiu del TFG tractar aquest àmbit.

2.1.1. Antecedents

Com ja s'ha vist, el nucli informàtic el conforma el SAP ERP. Aquest sistema però, no fa massa que està implantat a Alzamora. Inicialment, l'empresa disposava de un ERP fet a mida sobre AS400 (fig. 5 es pot veure la pantalla clàssica verda i negra d'aquests entorns) i mantingut per el propi Departament d'Informàtica de l'organització. Aquest ERP, tot i que cobria les necessitats de l'empresa, era poc escalable i massa rígid, fet que provocava que cada vegada que es demanava una millora o modificació del programa, aquestes peticions es convertien en petits projectes, fet que feia que a vegades, peticions de clients es convertissin en inviàbles.

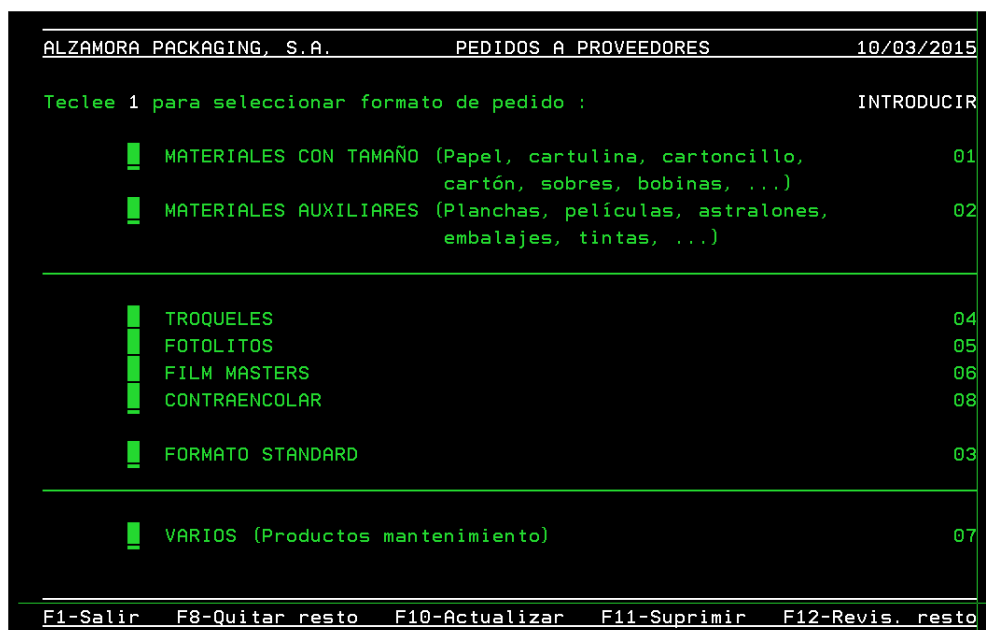


Fig. 5 – Interfície clàssica de l'entorn d'usuari sobre OS400

En quant al nostre SAP ERP, la major part dels mòduls s'han mantingut de forma estàndard, però evidentment, una gran part de la resta de mòduls s'han hagut d'adaptar a la nostra forma de treballar. Aquesta personalització ha estat forçosa donat que un ERP en si, tal com te'l serveixen, et serveix de res sinó hi ha una mínima adaptació, i és en aquesta personalització on rau la diferenciació de les empreses i de la seva forma de treballar, ja que si no hi hagués una mínima personalització, totes treballarien igual. En el nostre cas, tenim de tot: mòduls estàndards, transaccions fetes a mida (anomenades Z), etc. Entre tot això, també tenim les diferents integracions que s'han fet per tal d'adaptar el SAP a la resta de SI que no es van poder migrar (tot i que des del meu punt de vista, el SAP no s'ha d'adaptar a la resta sinó al revés, que per això és el nucli).

Per últim, un dels objectius pel qual vam decidir apostar per la implantació de un ERP com SAP va ser la necessitat d'eliminar les illes d'informació de que disposàvem en aquell moment (en el nostre cas, una illa d'informació és un sistema d'informació "satèl·lit" que conté dades repetides e inclús inconsistent, còpia de les dades del SI nucli de l'empresa). Principalment teníem illes d'informació en l'àmbit de dades mestres de producció, clients, mestres de materials, etc. Evidentment no s'han eliminat del tot, però amb la integració dels sistemes amb SAP hem posat ordre i hem eliminat la duplicitat de dades.

2.2. Detall dels SI i aplicacions principals d'Alzamora

Ara que hem fet la introducció general dels SI de l'empresa, passem a parlar amb més detall dels diferents SI i diferents aplicacions de que disposem i les seves funcions. L'ordre presentat no té res a veure amb la seva importància (o criticitat). *Com a recordatori, no ens referirem a ells amb el nom de la solució per qüestions de privacitat, sinó per la seva funció dins l'organització, tot i que per facilitar la lectura els abreviarem:*

2.2.1. Sistema de control de producció en planta (CPP).

Com el nom indica, aquest sistema porta el control de la producció a planta: control d'unitats produïdes, operari que ho ha fet, incidències, temps empleat, temps previstos i unitats previstes de cada feina, planificació de feines, etc. És una aplicació comercial feta a mida per Alzamora. En quant a característiques tècniques:

- Servidor físic dedicat
- Base de dades SQL SERVER 2005

- S.O. Windows Server 2003
- Terminals repartits a planta i a cada màquina a disposició dels operaris
- Tipus de dades mestres rellevants que necessita l'aplicació de SAP:
 - Clients
 - Materials
 - Ordres de fabricació
 - Operacions de les ordres de fabricació

Després del SAP, és un dels sistemes més importants de l'organització donades les seves funcions i importància.

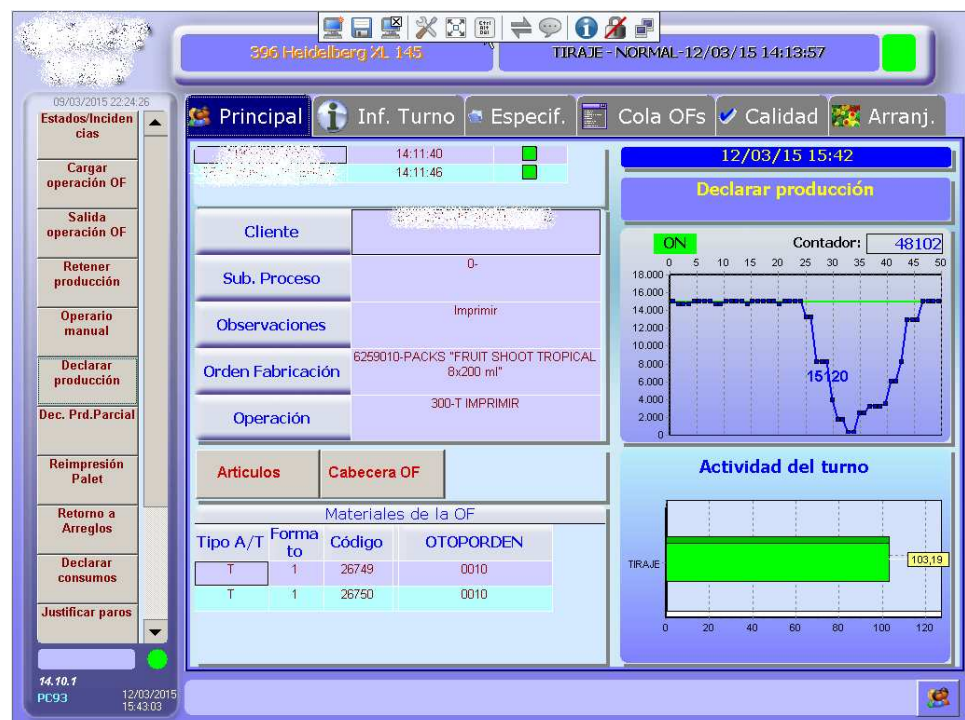


Fig. 7 – Interfície del CPP

2.2.2. Sistema de control de presència del personal (SCP):

Aquest sistema és l'encarregat de mantenir el mestre personal de l'empresa, a part que també és el que gestiona els horaris dels empleats, els permisos d'accés a cada secció, els dies de vacances i llicències, les primes de producció, absentismes, etc. És una aplicació comercial feta a mida per Alzamora. En quant a característiques tècniques:

- Servidor físic dedicat
- Base de dades SQL SERVER 2005
- S.O. Windows Server 2003
- Tipus de dades mestres rellevants que necessita l'aplicació de SAP:
 - Empleats (nom, cognoms, secció)
 - Status dels empleats (alta, baixa)

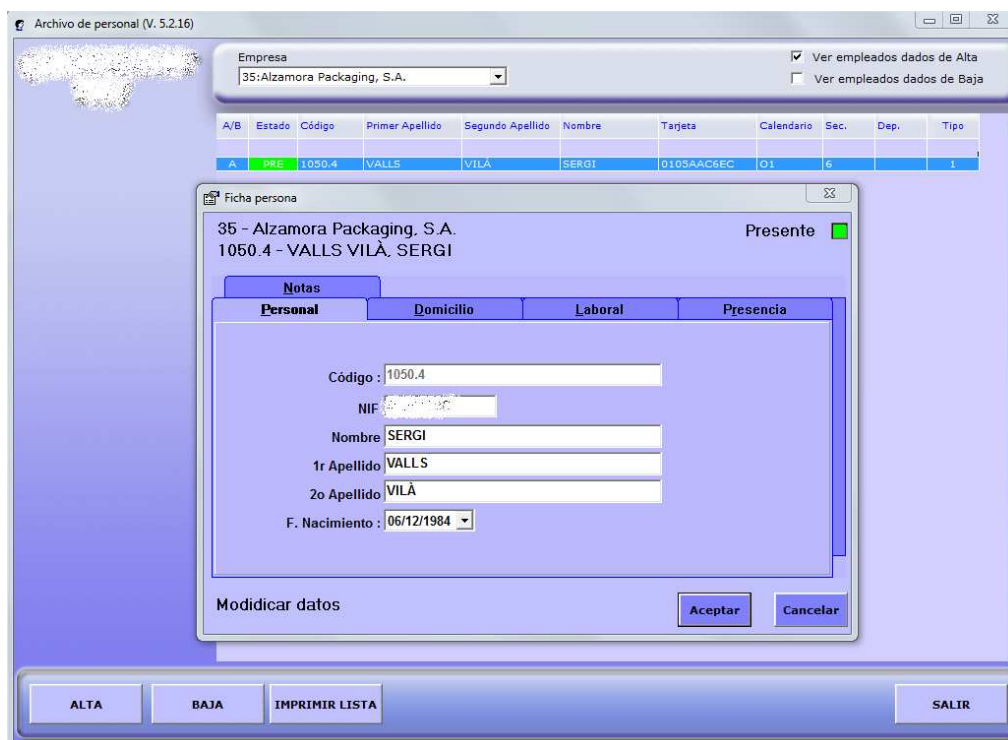


Fig. 8 – Interfície del SCP

2.2.3. Sistema de gestió de magatzems (SGM):

Aquest sistema és l'encarregat de gestionar el magatzem automàtic de que disposa l'empresa. Aquest és un magatzem completament robotitzat capaç d'emmagatzemar més de 8000 palets en els seus més de 10 pisos. Aquest magatzem és capaç de guardar tant matèria prima, com producte acabat com qualsevol altre material. És una aplicació comercial estàndard, amb les mínimes adaptacions per Alzamora. En quant a característiques tècniques:

- Servidor físic dedicat

- Base de dades Oracle 10i
- S.O. Windows Server 2008
- Tipus de dades mestres rellevants que necessita l'aplicació de SAP:
 - Materials
 - UMP (unitats de manipulació, és a dir, palets codificats)

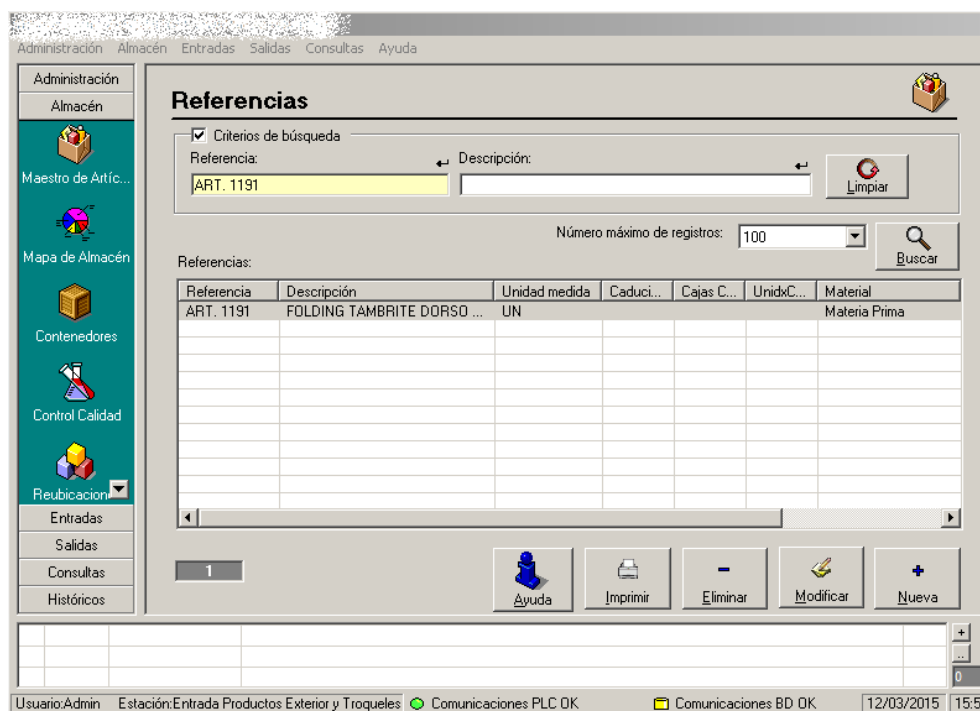





Fig. 9 – Interfície del SGM

2.2.4. Aplicació per la generació de banderoles identificatives de palet (GDB):

Aquest no és un sistema en si, sinó que és una aplicació feta a mida per el propi departament d'informàtica d'Alzamora, per tal d'imprimir banderoles (una banderola és un full imprès que s'enganxa de forma visible al palet informant de dades referents al palet en si, com pot ser el pes, material, client, descripció, matrícula identificativa (EAN128)) que es posaran amb els palets que s'entregaran als clients, a banda de les banderoles estàndard que Alzamora ja entrega (per exemple, hi ha clients que a banda de la banderola estàndard d'Alzamora, volen que al palet també hi hagi una banderola especial per ells, informant d'alguna dada extra que l'estàndard no té). En quant a característiques tècniques:

- Aplicació realitzada en Delphi 2007.
- Corre sobre Windows, en els diferents equips que imprimeixen les banderoles estàndard d'Alzamora.
- Tipus de dades mestres rellevants que necessita l'aplicació de SAP:
 - Clients
 - Materials
 - UMP (unitats de manipulació, és a dir, palets codificats)

ALZAMORA PACKAGING, S.A. CTRA. N-260, KM. 81 (OLOT EST) 17857 SANT JOAN LES FONTS	PRODUCTO PROV: 0006204 MRMS: 98960292 FECHA PROD.: 27/02/2012 PALET: 013000256
Caja 98960292 P3	
IRMS/CGAS#: 98960292	CANTIDAD: 600 Unidades
LOTE: ALZA00062153010	TIPO PALET: P3
 <small>(81) 98960292 (37) 600</small>	
 <small>(10) ALZA00062153010 (60) P3</small>	
 <small>(00) 19999990130002561</small>	



Cliente FERRERES UNIBELSA, S.A.

Fabricación/Artículo 3.441 21702.03

Descripción C-20 DIGESTIVE PLUS P3 P3 P3 P3

Artículo cliente 7048

Nº pedido cliente

Unidades por caja 25.000

Cajas por paleta 1

Cantidad total paleta 25.000

O.F.: 3441010 16.01.2014


(00)384000000130362239

Fig. 10 – Exemples de banderoles. La de l'esquerra és feta a mida amb el GDB i la de la dreta és l'estàndard del SAP

2.2.5. Aplicació per la cerca d'arxius de preimpresió (CDP):

Aquest tampoc és un sistema en si, sinó que és una aplicació feta a mida per el propi departament d'informàtica d'Alzamora. Aquesta en concret, serveix per cercar fitxers de preimpresió en un repositori de dades dins una cabina de discos SAN. El que l'aplicació permet,

a grans trets, és a partir de informació d'un client o d'un material del client obtinguda de SAP, munta la ruta exacta on el fitxer està guardat i l'obre, sense obligar a l'usuari navegar per tot l'arbre de fitxers cercant el fitxer en concret. En quant a característiques tècniques:

- Aplicació realitzada en JAVA, per tal de poder córrer tant sobre OSX com en PC. Principalment és usada per el Departament de Preimpressió i per aquest motiu és creada en JAVA, ja que al ser tots equips MAC amb OSX és més fàcil de desplegar.
- Tipus de dades mestres rellevants que necessita l'aplicació de SAP:
 - Clients
 - Materials

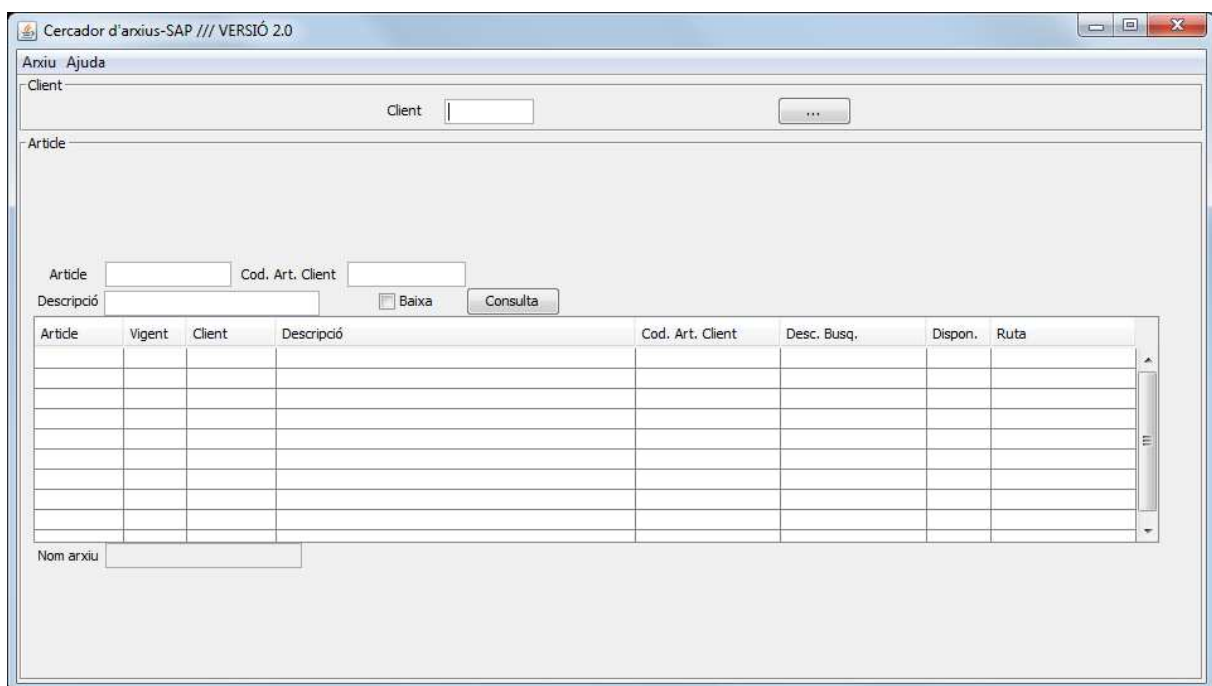


Fig. 11 – Interfície del CDP

2.2.6. Sistema pel disseny de maquetes (SDM):

Aquest sistema és l'encarregat de gestionar tota la part I+D i de la gestió de totes les maquetes i projectes tècnics que es realitzen. És un sistema CAD que serveix per realitzar els dissenys físics del packaging (les caixes), però a més, a l'estar centralitzat en una BD, relaciona aquests dissenys amb clients, comercials, materials, proveïdors... A la figura 13 podem veure un exemple de una caixa en estadis inicials del seu disseny, creat en el departament de R+D. En quant a característiques tècniques:

- Servidor físic dedicat
- Base de dades SQL SERVER 2008
- S.O. Windows Server 2008
- Aplicació instal·lada al Departament de I+D i a la resta de oficina tècnica per poder consultar els dissenys de les caixes
- Tipus de dades mestres rellevants que necessita l'aplicació:
 - Clients
 - Materials
 - Comercials/representants

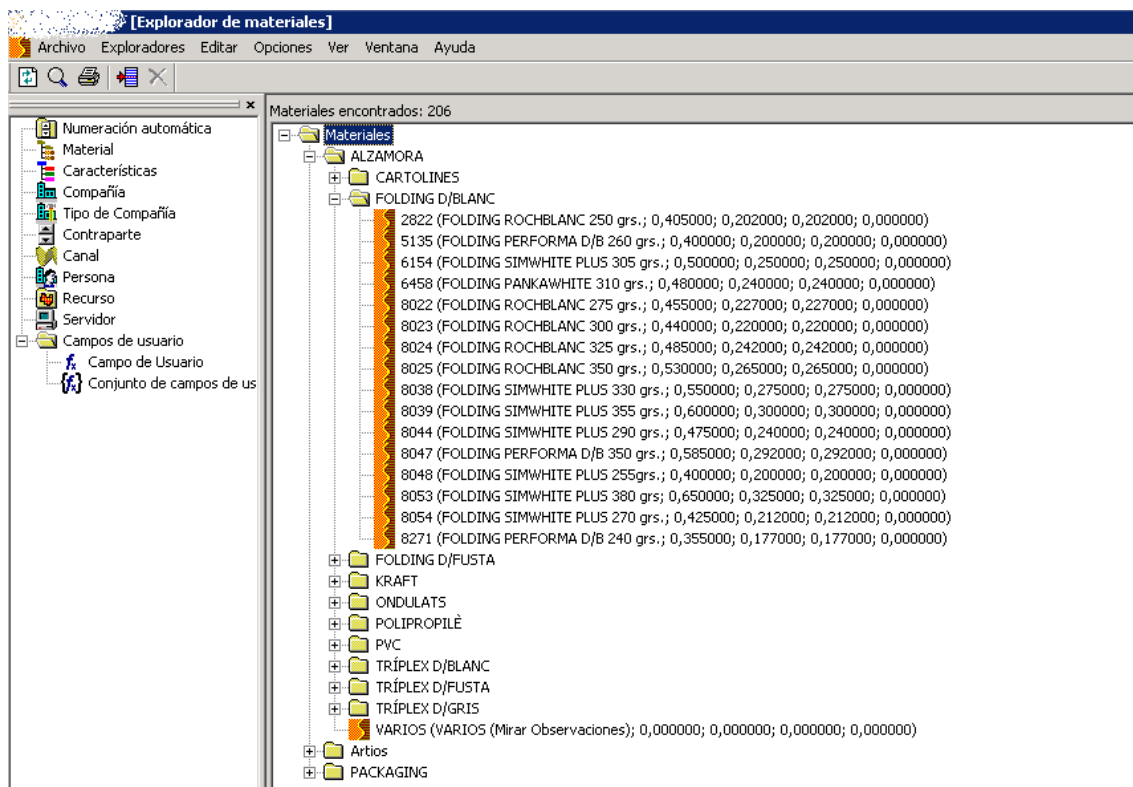


Fig. 12 – Interfície del SDM

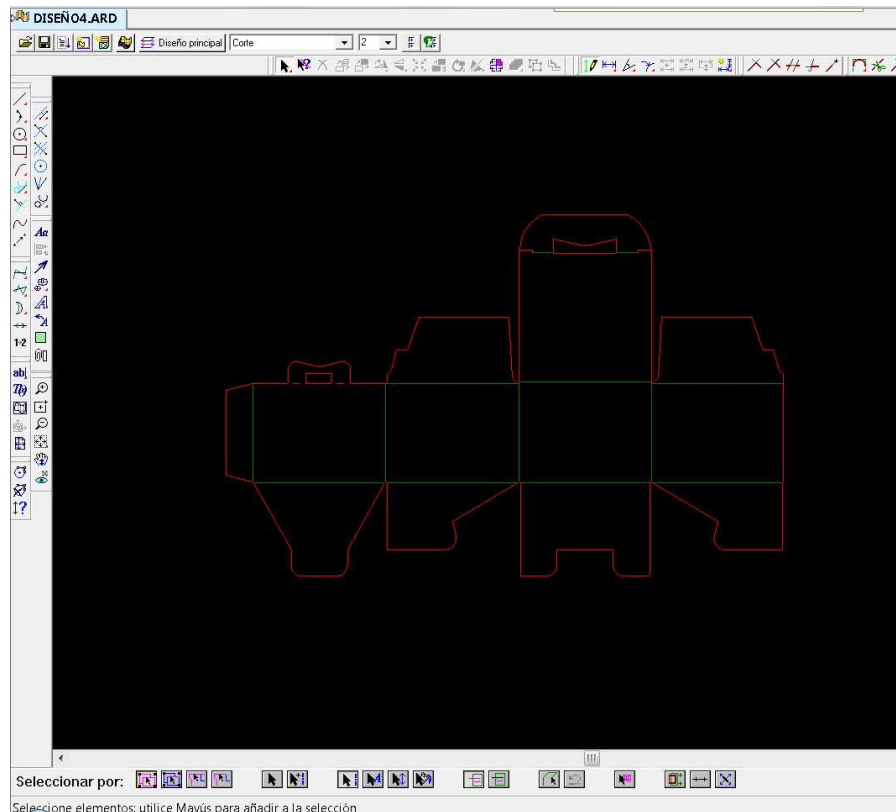


Fig. 13 – Exemple de disseny obtingut i gestionat amb el SDM

2.2.7. Sistema de preimpresió (SDP):

Aquest sistema és l'encarregat de gestionar tota la part de preimpresió de l'empresa. Sense entrar en detall, per tal de poder imprimir en impressores offset un disseny, prèviament cal haver processat les imatges, preparar-les i transformar-les en un format que els sistemes que controlen aquestes impressores, puguin entendre. Si el departament de I+D és l'encarregat de dissenyar la caixa física, el departament de preimpresió és l'encarregat de posar imatges a aquesta caixa i preparar-les en els formats d'impressió per tal de que en un sol format (fulla de cartró) n'hi puguin haver varies i aprofitar així al màxim la fulla. A les figures 15 i 16 es poden veure els dissenys ja acabats.

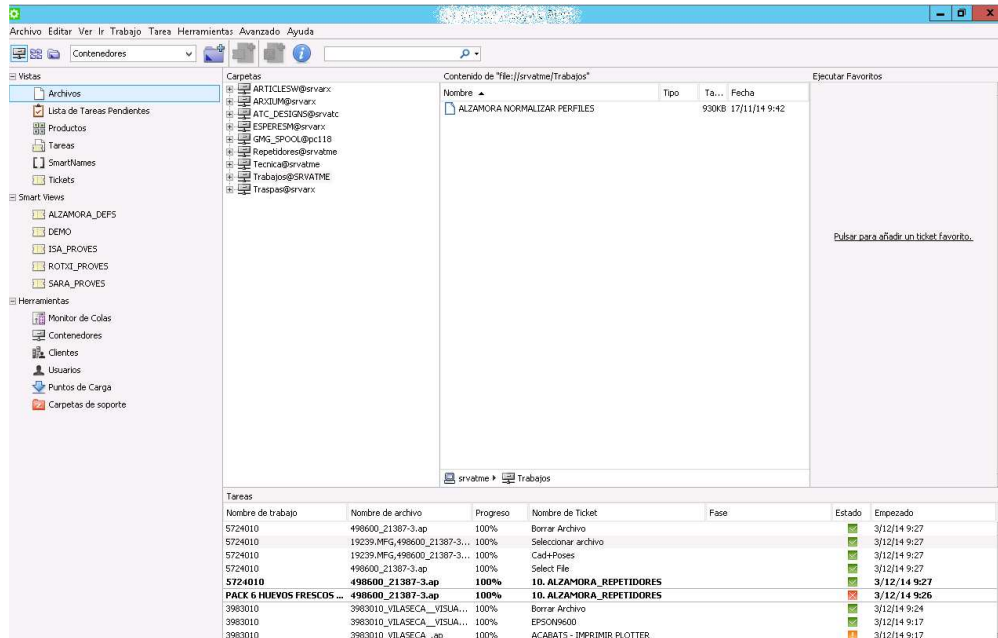


Fig. 14 – Interficie de l'SDP



Fig. 15 – Exemple del disseny d'una caixa ja acabat



Fig. 16 – Exemple del disseny en 3D d'una caixa ja acabada

- En quant a característiques tècniques:

- Servidor físic dedicat
- Base de dades SQL SERVER 2008
- S.O. Windows Server 2012
- Aplicació instal·lada al Departament de Preimpressió sobre equips MAC.
- Tipus de dades mestres rellevants que necessita l'aplicació:
 - Clients
 - Ordres de Fabricació
 - Comercials/representants

2.2.8. Aplicació per l'embalatge i etiquetat (SEE):

Aquest sistema és l'encarregat de gestionar tota la part d'etiquetatge de les caixes d'embalatge que es col·loquen sobre palets i que després s'entreguen als clients. Aquestes etiquetes realment són dades impreses directament sobre la caixa d'embalatge, que a mesura que van circulant per la cinta transportadora, van passant per davant d'uns injectors de tinta que van pintant les dades més rellevants d'aquella fabricació, com poden ser el client, la data, el número de caixa per la traçabilitat posterior, la descripció del material, etc. Aquesta aplicació és l'encarregada d'anar a buscar a SAP aquesta informació i proporcionar-la als injectors de tinta. És una aplicació comercial estàndard, amb les mínimes adaptacions per Alzamora. En quant a característiques tècniques:

- Instal·lada sobre un PC estàndard.
- Base de dades Access 2007.
- S.O. Windows XP
- Tipus de dades mestres rellevants que necessita l'aplicació de SAP:
 - Materials
 - Clients
 - Ordres de fabricació

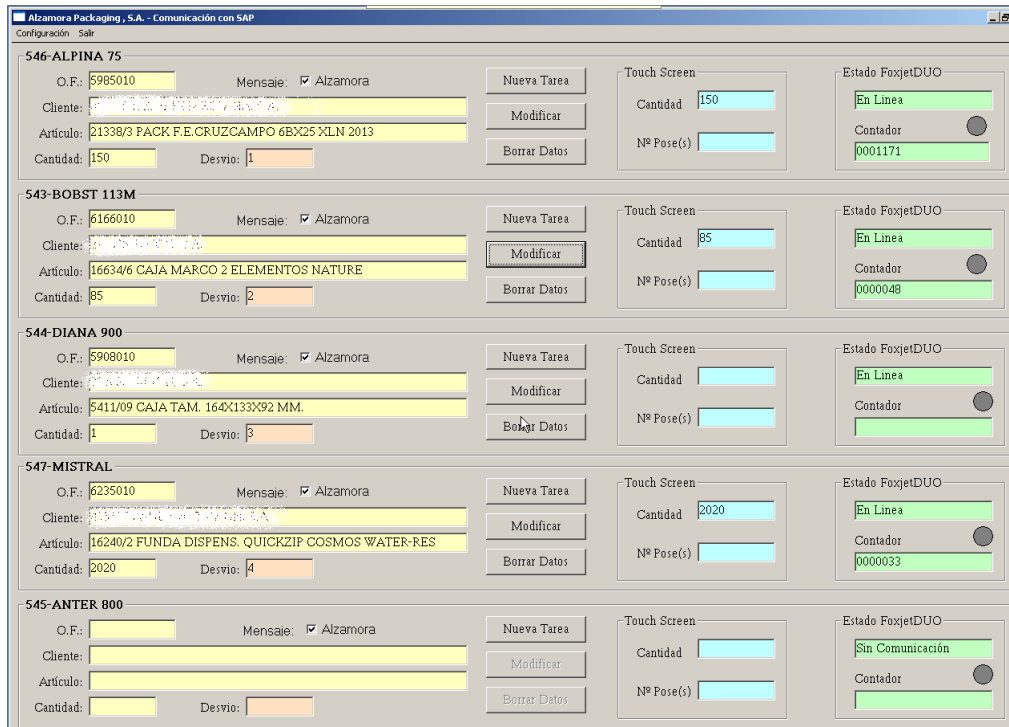


Fig. 17 – Interfície de l'SEE



Fig. 18 – Diagrama general de les aplicacions

Un cop presentats els principals sistemes d'informació d'Alzamora, passarem a introduir en detall quins d'aquests sistemes es van decidir integrar amb SAP i quines adaptacions en aquests sistemes es van haver de fer per tal d'aconseguir-ho.

3. Presentació de les integracions a realitzar

En aquest capítol el que farem serà centrar-nos en els sistemes d'informació que finalment s'integraran amb el SAP. Els que no, explicarem els motius pels quals no es fa, argumentant la decisió presa.

Dels sistemes presentats, es farà una integració de:

- Sistema de control de producció en planta (CPP)
- Sistema de control de presència del personal (SCP)
- Sistema de gestió de magatzems (SGM)
- Aplicació per la generació de banderoles identificatives de palet (GDB)
- Aplicació per la cerca d'arxius de preimpresió (CDP)
- Aplicació per l'embalatge i etiquetat (SEE)

Els sistemes que no s'integraran són:

- Sistema de preimpresió (SDP):
- Sistema pel disseny de maquetes (SDM):

Els motiu pel qual aquests últims no s'integraran, bàsicament són:

- En el cas de l'SDP (sistema de preimpresió), el sistema no existia quan es va posar en marxa el sistema SAP, per tant, aquest és un dels sistemes que s'integraran en properes fases.
- En el cas de l'SDM (sistema de disseny de maquetes), els motius van ser varis: el sistema, al ser una solució comercial distribuïda per una multinacional, és un sistema tancat que no permet la manipulació directe de taules i molt menys de la

pròpia aplicació, per tant, quedava descartada l'adaptació directe d'aquesta a SAP. Vam estar mirant alternatives amb la pròpia empresa distribuïdora per tal d'alimentar el sistema amb eines externes, però la única solució proposada per ells era adquirir un nou sistema d'informació (*middleware*) que fes de interconnexió entre SAP i el seu sistema. La solució era bona, però el cost de les llicències superava amb escreix el pressupost assignat per aquest projecte. Per aquest motiu, es va decidir posposar-ho fins que presentessin una aplicació més econòmica per tal objectiu.

3.1. Eines per interconnectar sistemes

Ara que ja sabem què hem d'integrar, hem de donar un cop d'ull a les eines que tenim per interconnectar diferents sistemes d'informació. En el nostre cas, hem fet servir 3 tecnologies diferenciades:

- Funcions/connexions RFC (*Remote Function Call*)
- Fitxers plans de text amb *hotfolders*
- Escriptura/lectura directe de taules

De les tres, les dues últimes són les més clares i simples. En el cas dels fitxers plans de text, el que es fa bàsicament és enviar les dades al sistema destí mitjançant l'escriptura d'aquestes en fitxers de text (txt, csv, etc.) i guardar aquests fitxers en una carpeta del sistema destí, que aquest va monitoritzant cada cert temps, per veure si hi ha fitxers per processar. Aquest temps tant poden ser cada *x* segons, com minuts o inclús hores. Un cop el sistema el té, l'agafa, el processa i l'esborra, en espera de l'arribada de més fitxers.

En el cas de l'escriptura i lectura directes a taula, és un mètode prou simple, però amb certes trampes. Té l'inconvenient que s'ha de conèixer prèviament l'estructura de les taules que s'escriuen o llegeixen, a part que s'ha d'anar molt en compte de no causar inconsistències a la base de dades per insercions o actualitzacions mal fetes. A més, cal vigilar amb les actualitzacions de la base de dades per si l'estructura de taules o camps canvia, per tal d'adaptar-nos a aquests canvis. En general, és un mètode fàcil si es coneix bé la base de dades que es tracta, però que es pot complicar en excés si aquests accessos no es fan amb cura i de forma rigurosa.

Per últim tenim el mètode de les connexions RFC. Una connexió RFC és una connexió mitjançant TCP/IP entre 2 sistemes diferents (que poden no tenir res a veure entre ells en quant

a arquitectura o tecnologia) la qual permet cridar funcions de forma remota en el sistema destí, això és:

- Des del sistema origen s'obre una connexió RFC contra el sistema destí
- Un cop oberta la connexió, el sistema origen crida la funció RFC contra el sistema destí. Encara que aparentment l'executi l'origen, realment és el destí qui l'executa
- El sistema destí obté les dades fruit de l'execució de la RFC (si és que aquesta ha de tornar dades, pot no retornar res) i les envia al sistema origen
- El sistema origen agafa les dades i tanca la connexió

Com es pot veure, és un sistema d'intercanviar informació molt potent degut a la simplicitat del procés, i només coneixent la forma de connectar al sistema destí, el nom i els paràmetres de la funció que es desitja executar, ja en tenim prou per obtenir dades d'un sistema sense conèixer res de la seva estructura interna, ni arquitectura ni dades internes.

4. Definició de les integracions

Ara que ja sabem què haurem d'integrar i les eines que podrem fer servir, anem a entrar en detall a cada una de les integracions dels sistemes escollits.

4.1. Integració del sistema de control de producció en planta (CPP)

Com hem comentat abans, aquest sistema és el que porta el control de la producció a planta: control d'unitats produïdes, operari que ho ha fet, incidències, temps empleat, temps previstos i unitats previstes de cada feina, planificació de feines, etc. És una aplicació comercial adaptada a Alzamora, per tant, al ser un producte comercial no en disposem el codi font, ni l'estructura interna, ni l'estructura de la base de dades, etc. per tant, totes les modificacions les han hagut d'implementar l'empresa propietària de la solució, i Alzamora no ha pogut tocar res del programa.

Si recordem, les dades que l'aplicació necessita de SAP, entre d'altres, són:

- Clients

- Materials
- Ordres de fabricació
- Operacions de les ordres de fabricació

Aquestes dades es mostren tant als terminals repartits per les màquines de planta com en els PC d'usuari que tenen l'aplicació instal·lada. Aquestes dades cal que el sistema les llegeixi del sistema SAP, donat que és qui conté la informació per omplir la base de dades del CPP. Cal també que les dades generades per el CPP (quantitat de tiratges realitzats, temps emprat dels operaris, incidències, etc.) pugin a SAP, donat que és aquest qui conté les dades principals de les ordres de fabricació i és sobre aquestes les que s'ha d'imputar tota aquesta informació generada per el CPP.

4.1.1. Procés de definició de les connexions

Al cap de moltes reunions amb els tècnics de l'aplicació i de molt d'anàlisi, es va arribar a la conclusió que optaríem per un sistema mixt entre l'ús de RFC i l'escriptura directe a taula. Les dades principals necessàries per el funcionament del CPP les extrauríem de SAP via RFC; les dades generades per el CPP que han de pujar a SAP, les enviarem directament escrivint a les taules de SAP, bàsicament per motius de simplicitat i de rendiment.

En les reunions amb l'equip tècnic del CPP, es van definir varies RFC encarregades d'obtenir la informació del SAP i enviar-la al CPP (de fet és el CPP qui executa les RFC, obté les dades i les processa en el seu sistema). Aquestes RFC es van definir juntament amb l'equip tècnic implantador del SAP, donat que són ells qui realment les van haver d'implementar. El procés seguit va ser:

- El CPP necessita omplir el seu sistema amb les dades de SAP
- L'equip tècnic del CPP defineix com han de ser les interfícies de les RFC per obtenir la informació
- L'equip tècnic de l'implantador de SAP implementa aquestes funcions
- L'equip tècnic del CPP juntament amb Alzamora valida que les RFC retornen correctament les dades i aquestes omplen correctament el sistema del CPP

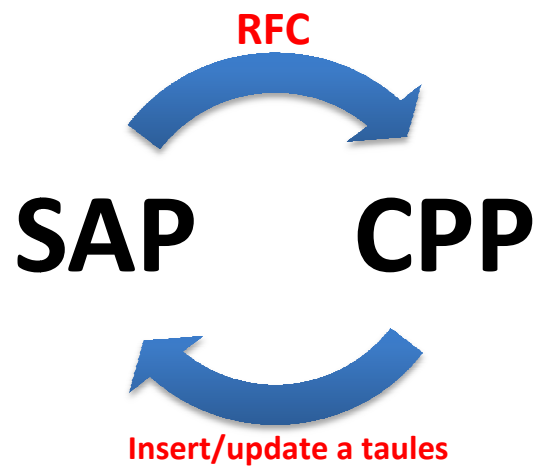


Fig. 19 – Flux d'informació entre SAP i el CPP

Passem doncs a descriure algunes de les RFC més importants (no posem les més específiques del negoci donat que no s'entendrien fora de context, donat que implementen estructures internes molt particulars d'Alzamora) i més fàcil d'explicar perquè implementen necessitats prou genèriques de qualsevol empresa.

4.1.2. Definició de les RFC

Les RFC més importants implementades són:

4.1.2.1. RFC ZQPLANT_CLIENTE_OF: Obtenció del client donada una Ordre de Fabricació

Aquesta funció el que fa és, donat un codi d'ordre de fabricació, retorna informació del client associada amb aquesta ordre de fabricació.

Els paràmetres que la funció retorna són:

Estructura	ZQPLANT_CLI_OF	activo				
Descripción breve	Clientes OF					
<div style="display: flex; justify-content: space-between;"> Atributos Componentes Ayuda p./Verif.ent. Campos de moneda/cantidad </div>						
<div style="display: flex; justify-content: space-between;"> <div style="display: flex; align-items: center;"> <div style="margin-left: 10px;"> Tipo instalado 1 / 7 </div> </div> </div>						
Componente	Tp...	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve
<u>CCLIER</u>	<input type="checkbox"/>	<u>KUNNR</u>	CHAR	10	0	Nº de cliente 1
<u>CGREMP</u>	<input type="checkbox"/>	<u>HKUNNR_KH</u>	CHAR	10	0	Número de cliente de la jerarquía de clientes superior
<u>CESTAD</u>	<input type="checkbox"/>	<u>ZCERCA_ESTAT</u>	CHAR	1	0	Estat
<u>CCLIEN</u>	<input type="checkbox"/>	<u>NAME1</u>	CHAR	30	0	Nombre
<u>CATESTI</u>	<input type="checkbox"/>	<u>KATR5</u>	CHAR	2	0	Atributo 5
<u>CATESTT</u>	<input type="checkbox"/>	<u>KATR6</u>	CHAR	3	0	Atributo 6
<u>CATESTP</u>	<input type="checkbox"/>	<u>KATR7</u>	CHAR	3	0	Atributo 7

Fig. 20 – Estructura per retornar dades a l'CPP

- Codi del client
- Codi del grup d'empreses al qual pertany (sempre buit donat que ara no s'usa)
- Estat del client (alta/baixa)
- Nom del client
- 3 camps indicant el tipus d'exigència de qualitat en els acabats

Fixem-nos ara amb la implementació de la funció:

```

FUNCTION zqplant_cliente_of.

DATA : l_aufnr(12) , l_aufnr1(12).

CLEAR t_cliente."t_cliente és l'estructura que guarda les dades a retornar

SELECT SINGLE * FROM aufk WHERE aufnr = l_aufnr.
SELECT SINGLE * FROM vbak WHERE vbeln = aufk-kdauf.
SELECT SINGLE * FROM knvh WHERE kunnr = vbak-kunnr AND
                vkorg = vbak-vkorg AND
                vtweg = vbak-vtweg AND
( datab <= sy-datum AND datbi >= sy-datum ).

SELECT SINGLE * FROM knal WHERE kunnr = vbak-kunnr.
SELECT SINGLE * FROM knvv WHERE kunnr = vbak-kunnr AND
                vkorg = vbak-vkorg AND
                vtweg = vbak-vtweg .

IF knal-ktokd = 'Z001'.
    t_cliente-cestad = 'A'. "estat del client
ENDIF.
IF knal-ktokd = 'Z012'.

```

```

t_cliente-cestad = 'P'. "estat del client
ENDIF.
IF knal-sperr IS NOT INITIAL OR
   knal-aufsd IS NOT INITIAL OR
   knal-lifsd IS NOT INITIAL OR
   knal-faksd IS NOT INITIAL OR
   knal-cassd IS NOT INITIAL OR
   knvv-aufsd IS NOT INITIAL OR
   knvv-lifsd IS NOT INITIAL OR
   knvv-faksd IS NOT INITIAL OR
   knvv-cassd IS NOT INITIAL.
t_cliente-cestad = 'K'. "estat del client
ENDIF.
IF knal-loevm = 'X'.
t_cliente-cestad = 'B'. "estat del client
ENDIF.

t_cliente-cclien = knal-name1. "nom del client

SELECT SINGLE katr5 katr6 katr7 INTO (t_cliente-catesti,t_cliente-catestt,t_cliente-
catestp) "categoria estètica del client
FROM knal
WHERE kunnr = vbak-kunnr.

APPEND t_cliente.

ENDFUNCTION.

```

Si mirem un exemple d'execució de la funció, veiem com ens retorna els camps omplerts amb les dades pertinents:

Editor estructuras: Visualizar T_CLIENTE desde entrada 1

Columna Entrada Metadatos

1 Entrada

CCLIER	CGREMP	C	CCLIEN	CA	CAT	CAT
25967		A	LACREM, S.A.			

Fig. 21 – Sortida de la RFC

4.1.2.1. RFC ZQPLANT_ORDEN: Obtenció de dades de la ordre de fabricació

Aquesta funció el que fa és, donat un codi d'ordre de fabricació, retorna informació referent a aquesta ordre de fabricació.

Els paràmetres que la funció retorna són:

Estructura: ZQPLANT_OF activo
 Descripción breve: QPLANT : datos maestros

Atributos Componentes Ayuda p./Verif.entr. Campos de moneda/cantidad

Tipo instalado 1 / 8

Componente	Tip...	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve
DNUMOF	<input type="checkbox"/>	CHAR12	CHAR	12	0	Campo carácter longitud 12
OCLIEN	<input type="checkbox"/>	KUNNR	CHAR	10	0	Nº de cliente 1
GLTRP	<input type="checkbox"/>	CHAR8	CHAR	8	0	Campo de caracteres long.8
OFECHA	<input type="checkbox"/>	CHAR8	CHAR	8	0	Campo de caracteres long.8
ODESC1	<input type="checkbox"/>	AUFTEXT	CHAR	40	0	Texto breve
QTROQU	<input type="checkbox"/>	XFELD	CHAR	1	0	Casilla de selección
OOKCLI	<input type="checkbox"/>	XFELD	CHAR	1	0	Casilla de selección
OLIBER	<input type="checkbox"/>	ZOLIBER	CHAR	1	0	OF Liberada

Fig. 22 – Estructura per retornar dades al CCP

- Codi de la ordre de fabricació
- Codi del client
- Data inici previst de la ordre de fabricació
- Data final previst de la ordre de fabricació
- Descripció de la ordre de fabricació
- 3 camps indicant status interns de la ordre

Fixem-nos ara amb la implementació de la funció:

```

FUNCTION zqplant_orden.

DATA : l_aufnr(12) , l_aufnr1(12) ,
       l_objid LIKE affh-objid ,
       l_eqnr LIKE crve_a-eqnr ,
       l_objnr LIKE equi-objnr.
TABLES: jest.

SELECT SINGLE * FROM aufk WHERE aufnr = l_aufnr.
SELECT SINGLE * FROM afko WHERE aufnr = l_aufnr.
SELECT SINGLE * FROM vbak WHERE vbeln = aufk-kdauf.

SELECT SINGLE objid INTO l_objid
FROM affh WHERE aufpl = afko-aufpl.

SELECT SINGLE eqnr INTO l_eqnr
FROM crve_a
WHERE objid = l_objid AND objty = 'FH'.

```

```

IF sy-subrc EQ 0.
*Comprovamos status correcto
  SELECT SINGLE objnr INTO l_objnr
  FROM equi
  WHERE equnr = l_equnr.

  CALL FUNCTION 'STATUS_CHECK'
  EXPORTING
    objnr          = l_objnr
    status         = 'I0184'  "Status ALMA
  EXCEPTIONS
    status_not_active = 2.

IF sy-subrc = 0.
  CALL FUNCTION 'STATUS_CHECK'
  EXPORTING
    objnr          = l_objnr
    status         = 'E0002'  "Status perfil OK
  EXCEPTIONS
    status_not_active = 2.

IF sy-subrc EQ 0.

ELSE.
  t_datos-qtroqu = 'X'.
ENDIF.

ELSE.
  t_datos-qtroqu = 'X'.
ENDIF.
ENDIF.

t_datos-onumof = l_aufnr1.

t_datos-gltrp = afko-gltrp.
t_datos-ofecha = aufk-erdat.

t_datos-odescl = aufk-ktxt.

CALL FUNCTION 'STATUS_CHECK'
EXPORTING
  objnr          = vbak-objnr
  status         = 'E0001'  "ZREV En revisió
EXCEPTIONS
  object_not_found = 1
  status_not_active = 2.
IF sy-subrc = 0.
  t_datos-ookcli = 'X'.
ELSE.
  CALL FUNCTION 'STATUS_CHECK'
  EXPORTING
    objnr          = vbak-objnr
    status         = 'E0003'  "ZPRS Pruebas SI
  EXCEPTIONS
    object_not_found = 1

```

```

        status_not_active = 2.
    IF sy-subrc = 0.
        t_datos-ookcli = 'X'.
    ENDIF.
ENDIF.

SELECT SINGLE * FROM jest WHERE objnr = aufk-objnr AND
    stat = 'I0001' AND "ABIE
    inact = ' '.

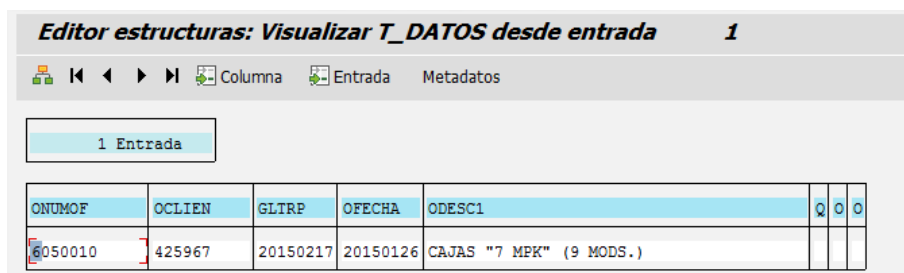
IF sy-subrc = 0.
    t_datos-oliber = 'X'.
ENDIF.

APPEND t_datos. t_datos és l'estructura que guarda les dades a retornar

ENDFUNCTION.

```

Si mirem un exemple d'execució de la funció, veiem com ens retorna els camps omplerts amb les dades pertinents:



The screenshot shows a window titled "Editor estructuras: Visualizar T_DATOS desde entrada" with a sub-header "1". Below the title bar are navigation icons and labels for "Columna", "Entrada", and "Metadatos". A button labeled "1 Entrada" is visible. The main area contains a table with the following data:

ONUMOF	OCLIEN	GLTRP	OFECHA	ODESC1			
050010	425967	20150217	20150126	CAJAS "7 MPK" (9 MODS.)			

Fig. 23 – Sortida de la RFC

4.1.3. Exemples d'ús de les RFC

Fixem-nos ara com aquestes RFC es criden mitjançant el codi del programa. Com ja hem dit, al no tenir el font disponible de l'aplicació donat que és propietària, ens han proporcionat un esquelet de com seria la connexió i posterior crida de qualsevol RFC. En aquest cas, està escrit en Visual Basic 6 (VB6). Aquí es pot veure com ho fan per obrir la connexió:

```

Connect (False)

Private Function ConnectSAP(Extern As Boolean) As Boolean
    Try
        _SapConnection = New SAPConnection
        Dim Parameters = New SAPParameters(_QplantConnection.Connection).GetParameters

        If Parameters.Count > 0 Then

            _SapConnection.Name = Parameters("Name").ToString ' "ALZAMORA" 'nom de la connexió
            _SapConnection.User = Parameters("User").ToString ' "RFC ZQPLANT" 'nom de l'usuari RFC
            _SapConnection.Password = Parameters("Password").ToString ' "12345678" 'password
            _SapConnection.Client = Parameters("Client").ToString ' "4444" 'mandant
            _SapConnection.SystemID = Parameters("SystemID").ToString ' "00" ' sistema
            _SapConnection.Language = Parameters("Language").ToString ' "ES" 'idioma
            _SapConnection.AppServerHost = Parameters("AppServerHost").ToString ' "192.168.1.100" IP del sistema
            _SapConnection.SystemNumber = Parameters("SystemNumber").ToString ' "00" 'codi del sistema

            If Extern = True Then
                _SapConnection.SAPRouter = Parameters("SAPRouter").ToString ' "ZQPLANT.SAP.COM"
            End If

            If _SapConnection.Connect(Extern) Then
                Return True
            Else
                _ErrorMessage = _SapConnection.ErrorMessage
                Return False
            End If
        Else
            _ErrorMessage = "Falta parametros de conexión"
            Return False
        End If
    Catch ex As Exception
        _ErrorMessage = ex.Message
        Return False
    End Try
End Function

Public Function Connect(ByVal Extern As Boolean) As Boolean
    Try
        _Destination = RfcDestinationManager.GetDestination(CreateParametersConnection(Extern))
        RfcSessionManager.BeginContext(_Destination) 'Abrir conexión
        _Destination.Ping()
        Return True
    Catch ex As Exception
        _ErrorMessage = ex.Message
        Return False
    End Try
End Function

```

I aquí com ho fan per obtenir les dades de la RFC ZQPLANT_CLIENTE_OF:


```

Dim WoClientTable As DataTable = GetData("ZQPLANT_CLIENTE_OF", "I_CLIENTE", "I_QNUMOF", WorkOrderNumber)

Public Function GetData(FunctionName As String, TableName As String, NameFieldFilter As String, ValueFieldFilter As String)
    As System.Data.DataTable Implements IWorkOrdersRows.GetData
    Dim Service As New SAPService
    Dim Dt As DataTable = Service.GetTable(_SAPConnection, FunctionName, TableName, NameFieldFilter, ValueFieldFilter)
    If Dt Is Nothing Then
        RaiseEvent InformationWorkOrders(FunctionName, "Error SAP " & Service.ErrorMessage, WorkOrder.Status.Invalid)
    End If
    Return Dt
End Function

Public Function GetTable(ByVal SapConnection As SAPConnection, ByVal FunctionName As String, ByVal TableName As String, ByVal FieldNameFilter As String,
    ByVal ValueFilter As String, Optional ByVal NameParameterReturn As String = "") As DataTable
    Try
        _RfcFunction = SapConnection.Destination.Repository.CreateFunction(FunctionName) 'Creo la funcion
        _RfcFunction.SetValue(FieldNameFilter, ValueFilter)
        _RfcFunction.Invoke(SapConnection.Destination) 'Ejecuto

        If NameParameterReturn.Trim.Length > 0 Then
            _ValueReturn = _RfcFunction.GetValue(NameParameterReturn)
        End If

        Return GetDataTableFromRFCTable(_RfcFunction.GetTable(TableName))
    Catch ex As Exception
        _ErrorMessage = ex.Message
        Return Nothing
    End Try

Private Function GetDataTableFromRFCTable(lrfcTable As IRfcTable) As DataTable
    Dim loTable As New DataTable()

    '... Create ADO.Net table.
    For liElement As Integer = 0 To lrfcTable.ElementCount - 1
        Dim metadata As RfcElementMetadata = lrfcTable.GetElementMetadata(liElement)
        loTable.Columns.Add(metadata.Name)
    Next

    '... Transfer rows from lrfcTable to ADO.Net table.
    For Each row As IRfcStructure In lrfcTable
        Dim ldr As DataRow = loTable.NewRow()
        For liElement As Integer = 0 To lrfcTable.ElementCount - 1
            Dim metadata As RfcElementMetadata = lrfcTable.GetElementMetadata(liElement)
            ldr(metadata.Name) = row.GetString(metadata.Name)
        Next
        loTable.Rows.Add(ldr)
    Next

    Return loTable
End Function
End Function

```

4.2. Integració del sistema de control de presència del personal (SCP)

Aquest sistema és l'encarregat de gestionar el personal de l'empresa i tots els temes relacionats amb aquest, és a dir, és el que gestiona els horaris dels empleats, els permisos d'accés a cada secció i a cada porta d'entrada i sortida de l'empresa, els dies de vacances i llicències, les primes de producció, absentismes, etc. És una aplicació comercial adaptada a Alzamora, per tant, al ser un producte comercial no en disposem el codi font, ni l'estructura interna, ni l'estructura de la base de dades, etc. per tant, totes les modificacions les han hagut d'implementar l'empresa propietària de la solució, i Alzamora no ha pogut tocar res del programa.

Si recordem, les dades que l'aplicació necessita de SAP, entre d'altres, són:

- Empleats (nom, cognoms, secció)
- Status dels empleats (alta, baixa)

Aquestes dades les utilitza el SCP per tal de configurar tot el relacionat amb l'empleat, a part que el SAP també ho necessita per tal de poder lligar les dades de producció amb la persona que ho ha fet.

4.2.1. Procés de definició de les connexions

En les reunions amb l'equip tècnic de l'aplicació i el partner de SAP, es va arribar a la conclusió que el seria més fàcil en aquest cas, i com que les dades mestres a exportar de l'SCP a SAP són molt simples (nom, cognoms, secció de l'empleat, status), seria crear una RFC a SAP que permetés enviar-li les dades i poder així donar d'alta l'empleat a SAP (el mòdul de HR que porta el propi ERP és força complex de gestionar, i per tant és molt millor cridar les funcions de manteniment d'empleats directament).

A grans trets, tal com s'ha configurat, és que quan l'usuari crea o modifica un empleat a l'SCP, just després es crida la RFC amb les dades pertinents indicant a SAP les modificacions que ha de fer. En aquest cas, les dades mestres no les té el SAP sinó que justament les necessita estirar del SCP, que és realment el mestre de les dades, per tant, el que necessitem en aquest cas és que el SCP envii les dades a SAP.



Fig. 24 – Flux d'informació entre SAP i el SCP

4.2.2. Definició de la RFC

La RFC més important implementada és:

4.2.2.1. RFC ZQPRESENT_OPERARIS: manteniment del mestre d'empleats de SAP

Aquesta funció el que permet és fer el manteniment dels empleats de SAP. La funció té com a paràmetres d'entrada:

Módulo funciones		ZQPRESENT_OPERARIS		Activo		
Atributos	Import	Export	Modif.	Tablas	Excep.	Cód.fte.
Nombre parám.	Tipif...	Tipo ref.	Valor propuesta	Op...	Tr...	Texto breve
PI_OPERARI	TYPE	CHAR10		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Operari
PI_NOM	TYPE	CHAR35		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre
PI_COGNOM1	TYPE	CHAR40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cognom 1
PI_COGNOM2	TYPE	CHAR40		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cognom 2
PI_SECCIO	TYPE	NUMC4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Secció
PI_TIPUS	TYPE	CHAR1		<input type="checkbox"/>	<input checked="" type="checkbox"/>	A(Alta) B(Baixa) M(Modificacio) E(Esborrar)

Fig. 25 – Paràmetres d'entrada de la RFC

- Codi operari
- Nom
- Cognom1
- Cognom2
- Secció assignada
- Tipus de moviment: Alta, Baixa, Modificació.

Llavors, si des del mestre de personal de l'SCP es dona d'alta un operari, es cridarà la RFC amb les dades mestres de l'operari i una A al tipus de moviment. Si pel contrari es modifica, s'enviarà una M i si es dona de baixa, una B.

En aquest cas, no disposem del codi font del programa que la crida donat que el proveïdor no ens l'ha pogut facilitar. Tampoc mostrem el codi font de la RFC perquè és molt semblant a les RFC anteriors que hem mostrat.

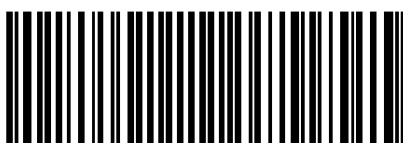
4.3. Integració de l'aplicació per la generació de banderoles identificatives de palet (GDB)

Com hem dit abans, a cada palet que es genera, cal imprimir i penjar una banderola identificativa única per cadascun dels palets, per tal de poder-los identificar unívocament i habilitar així una traçabilitat total. A més, aquesta banderola conté informació extra referent a aquell palet, com pot ser el client, la ordre de fabricació que l'ha creat, la data que s'ha creat, la

quantitat, el material, etc. Algunes vegades passa, però, que el client no en fa prou amb aquesta banderola estàndard que el SAP genera, sinó que en necessita alguna altra més perquè les seves especificacions així ho demanen. Aquestes banderoles s'identifiquen amb un codi de barres EAN128.

Alzamora | PACKAGING

Cliente FARMACIA MARIJA, S.A.
Fabricación/Artículo 3.441 21702.03
Descripción C-20 DIGESTIVE PLUS FARMACIA
Artículo cliente 7048
Nº pedido cliente
Unidades por caja 25.000
Cajas por paleta 1
Cantidad total paleta 25.000
O.F.: 3441010 16.01.2014



(00)384000000130362239

Fig. 26 – Banderola estàndard de SAP en DINA4

Aquesta aplicació és l'encarregada de generar aquestes banderoles "extres" que alguns clients demanen. Aquestes banderoles s'imprimeixen juntament amb les estàndards de SAP i es posen també amb els palets, seguint les especificacions dels clients. Les dades mestres rellevants que necessita l'aplicació de SAP per tal de poder obtenir aquesta informació extra són:

- Clients
- Materials
- UMP (unitats de manipulació). Per el SAP, un palet generat és una UMP.

Es va decidir no crear aquestes banderoles a SAP per una qüestió purament pràctica en el sentit de la programació i versatilitat, ja que l'eina que SAP disposa per generar aquest tipus de formularis (*smartforms*) no és suficientment àgil ni potent per arribar al nivell que els clients

demanen en les seves banderoles fetes a mida. Per això es va decidir crear una aplicació interna d'Alzamora feta en el llenguatge Delphi amb OO per cobrir aquestes necessitats. Es va triar aquest llenguatge perquè ja es disposava de llicència per crear formularis i codis de barres EAN128, cosa que no teníem amb l'IDE de JAVA de que disposàvem.

4.3.1. Procés de definició de les connexions

En les reunions amb l'equip d'Alzamora i el partner de SAP, es va arribar a la conclusió que, en aquest cas, com que les dades que necessitava l'aplicació eren poques i que la complexitat no era molt elevada, es faria servir un fitxer de text per tal de proporcionar a l'aplicació GDB la informació per poder generar la banderola especial. En aquest txt, cada línia del fitxer serà una dada diferent. Evidentment, en funció de cada banderola diferent, aquestes línies tindran significats diferents. El GDB ja sabrà en cada cas quina informació hi ha a cada línia per cada banderola diferent. En concret, el comportament de l'aplicació serà el següent:

1. A SAP es genera un palet de producte acabat (material)
2. Al generar-se el palet, s'imprimeix la banderola estàndard del SAP
3. Quan ha imprès la banderola estàndard, el SAP consulta si el client en concret necessita banderola especial
4. En cas afirmatiu, genera un fitxer txt (el guarda a la c:\exe) amb les dades que el client necessita i crida el GDB amb els paràmetres pertinents
5. Un cop el DGB ha imprès la banderola especial, elimina el txt generat

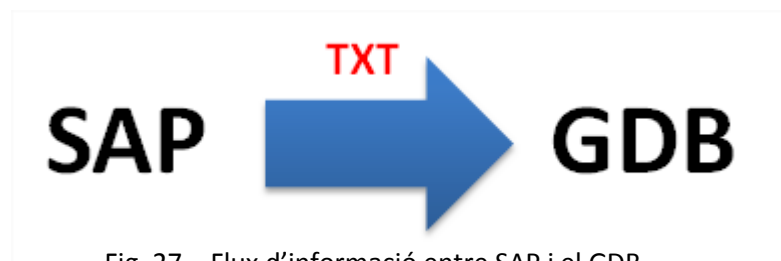


Fig. 27 – Flux d'informació entre SAP i el GDB

4.3.2. Exemples de crida del GDB

El GDB espera 3 paràmetres:

- El codi de la banderola del client a generar (cada banderola especial de client té un codi identificatiu)

- El path del fitxer on anar a buscar les dades
- El número de còpies que s'han d'imprimir de la banderola

L'aplicació està instal·lada a la ubicació c:\exe\generadorBanderolesEspecials.exe.

Una crida d'exemple seria c:\exe\generadorBanderolesEspecials.exe BPOMPADOUR c:\exe\BPO_20150316155710.txt 2. En aquest cas estem dient que ha d'imprimir la banderola especial del model BPOMPADOUR, que ha d'anar a llegir les dades al fitxer c:\exe\BPO_20150316155710.txt i que ha d'imprimir 2 còpies.

Com a exemple, el contingut del fitxer txt és aquest:

```

1 013033860
2 3060030
3 20131210
4 17184.02
5 CAJA 20 MELISA(CIDREIRA) DIA
6
7 10700.000

```

En el cas de la banderola BPOMPADOUR, tenim que la primera posició és la matrícula de la UMP, la segona és la ordre de fabricació, la tercera és la data de creació de la UMP en format SAP, la quarta és el material, la cinquena és la descripció, la sisena és el codi de comanda de client (en aquest cas no en té) i la última és la quantitat de la UMP.

Fixem-nos ara amb parts importants del codi font de l'aplicació:

Main:

```

type TGeneradorBanderolesEspecials=class(TList)
  constructor Create(codiBandEsp:string;pathFitxDad:string;numcopi:integer);
  Destructor Destroy;
private
  codiBanderolaEspecial:string;
  pathFitxDades:string;
  numcopies:integer;
  dadesBanderolaEspecial:Tstringlist;
  fitxDades:textfile;
  fitxerLog:textfile;
  pathFitxerLog:string;
  function obtenirDadesFitxer():boolean;
  function dataSAPDataNormal(data:string):string; //passa de aaaammdd a dd/mm/aaaa

public
  procedure generaBanderolaEspecial();
end:

```

La definició de la classe del generador:

```
// obtenim els 3 paràmetres
codiBanderola:=ParamStr(1);
nomfitxer:=ParamStr(2);
numcopies:=strtoint(ParamStr(3));

mem.lines.add('Generant banderola especial: '+codiBanderola + ' - ' + nomfitxer + ' - ' + ParamStr(3));
//instanciem l'objecte banderolaEspecial
banderola:=TGeneradorBanderolesEspecials.Create(codiBanderola,nomFitxer,numcopies);
//la imprimim
banderola.generaBanderolaEspecial;
banderola.Free;
//esborrem el fitxer txt amb les dades
DeleteFile(nomfitxer);
```

El mètode que llegeix del fitxer i obté les dades:

```
function TGeneradorBanderolesEspecials.obtenirDadesFitxer():boolean;
var
    valorTxt:string;
    estat:boolean;
begin
    if UpperCase(codiBanderolaEspecial)='BPOMPADOUR' then
    begin
        while not eof(fitxerDades) do //anem llegint el fitxer txt de les dades i ens les guardem
        begin
            try
                Readln(fitxerDades,valorTxt);
                dadesBanderolaEspecial.Add(valorTxt);
            except
                on e:exception do
                begin
                    estat:=false;
                    Writeln(fitxerLog,datetimetostr(now)+' - error on obtenirDadesFitxer BPOMPADOUR: '+e.Message);
                    Application.Terminate;
                end;
            end;
        end;
    end;

    CloseFile(fitxerDades); //el tanquem aquí perquè ja l'han fet servir
    result:=estat;
end;
```

El mètode que a partir de les dades i del tipus de banderola, la instancia i la imprimeix:

```

procedure TGeneradorBanderolesEspecials.generaBanderolaEspecial();
var
  banEspecialPompadour:TBanderolaPompadour;
  llistatPompadour:tqrbanderolaEspecialPompadour;
begin
  if obtenirDadesFitxer() then //emplenem l'stringlist
  begin
    if UpperCase(codiBanderolaEspecial)='BPOMPADOUR' then
    begin
      //llegim el txt. Cada línia és una dada
      matricula:=dadesBanderolaEspecial[0]; //el codi de la UMP
      nof:=dadesBanderolaEspecial[1]; //ordre de fabricació
      data:=dadesBanderolaEspecial[2]; //data en format SAP
      data:=dataSAPADataNormal(data); //passem a la data a format DD/MM/AAAA
      nostreCodiArticle:=dadesBanderolaEspecial[3]; //codi material
      desc:=dadesBanderolaEspecial[4]; // descripció de l'article
      pedidoCliente:=dadesBanderolaEspecial[5]; //codi comanda del client
      totalPalet:=dadesBanderolaEspecial[6]; //quantitat del palet

      //muntem el codi EAN per pintar al codi de barres
      eanPompadour:='8400000'; //84 + 00000 del codi ean nostre.
      eanPompadour:=eanPompadour+fin+'0'; //els 8400000+ ultims 5 dígits nostre codi article +0

      eanPompadour:=convertirEan13aEan14('0',eanPompadour); //afegeix un 0 a davant i calcula el check = de 13 a 14 dígit
      lot:=trim(nof); //el lot és l'OF
      banEspecialPompadour:=TBanderolaPompadour.create(eanPompadour,matricula,lot,data,nostreCodiArticle,desc,
        matricula,totalPalet,pedidoCliente);

      //preparem la banderola
      banEspecialPompadour.prepararBanderola;
      //creem el llistat
      llistatPompadour:=tqrbanderolaEspecialPompadour.Create(banEspecialPompadour);
      llistatPompadour.Prepare;
      //imprimim
      for i := 0 to numcopies - 1 do
      begin
        llistatPompadour.Print;
      end;
      llistatPompadour.free;
      banEspecialPompadour.free;
    end;
  end;
end;
end;

```

Un cop s'ha imprès, ja tenim la banderola creada e impresa:

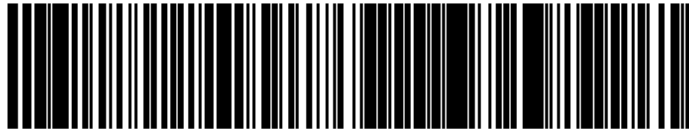


Alzamora PACKAGING		Fecha: 10/12/2013
CTRA. N-260, KM. 81 (OLOT EST)		
17857 SANT JOAN LES FONTS		
CLIENTE: PONSANO, S.L.		
Nº Pedido		
Articulo CAJA 20 MELISA(CIDREIRA) DIA		
Cod. Artículo 17184.02	EAN: 08400000171847	
Lote: 3060030	Nº Palet 013033860	
Cantidad 10700		
 (02) 08400000171847 (37) 10700 (10) 3060030		
 (00) 184000000130338605		
 400		

Fig. 28 – Banderola especial generada amb el GDB

4.4. Integració de l'aplicació per la cerca d'arxius de preimpressió (CDP)

El departament de preimpressió d'Alzamora, és l'encarregat de rebre els dissenys dels clients i és l'encarregat de manipular aquests dissenys per tal de poder entregar aquesta informació als sistemes que controlen les impressores offset. Com es pot entreveure, es disposa de un gran volum d'informació rebuda, entre pdfs d'alta resolució, imatges, documentació, imatges del client, etc. Tota aquesta informació és guardada en un repositori comú pel

departament organitzada en jerarquies de carpetes. Calia doncs, una aplicació que permetés cercar tots aquests fitxers sense necessitat d'haver de navegar per l'arbre de carpetes cada vegada, i a més, que es poguessin cercar aquests fitxers a partir d'informació del material o el client. Un cop obtinguda la informació, l'aplicació munta la ruta del fitxer i el permet obrir.

És una aplicació realitzada en JAVA, per tal de poder córrer tant sobre OSX com en PC. Principalment és usada per el Departament de Preimpresió i per aquest motiu és creada en aquest llenguatge portable, ja que al ser tots equips MAC amb OSX és més fàcil de desplegar. Si recordem, les dades que l'aplicació necessita de SAP, entre d'altres, són:

- Materials
- Clients

4.4.1. Procés de definició de les connexions

Parlant amb l'equip tècnic del partner implantador, i consensuat amb Alzamora, es va arribar a l'acord que en el cas del CDP, l'extracció de la informació del SAP es faria via RFC, donat que l'accés a les dades havia de ser a temps real i que l'estructura de les taules de SAP on hi havia la informació guardada era massa complexa com per fer les cerques de forma externa. Es va descartar l'ús de fitxers TXT perquè no proporcionaven l'agilitat ni flexibilitat que proporciona una RFC. Finalment, caldria reescriure el codi de la part de cerques de l'aplicació, canviant les antigues cerques directes sobre l'antic AS400 i fent servir la nova RFC.



Fig. 29 – Flux d'informació entre SAP i el CDP

4.4.2. Definició de la RFC

La RFC més important implementada és:

4.4.2.1. ZCERCA_ARXIUS: cerca informació de materials i clients

Aquesta RFC és l'encarregada de obtenir informació del SAP de materials i clients, a partir de certa informació d'entrada. Els paràmetres que espera d'entrada són:

Módulo funciones ZCERCA_ARXIUS Activo

Atributos Import Export Modif. Tablas Excep. Cód.fte.

Nombre parám.	Tipif...	Tipo ref.	Valor propuesta	Op...	Tr...	Texto breve
I_KUNNR	TYPE	ZCERCA_KUNNR		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Cliente
I_NAME1	TYPE	NAME1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Nombre
I_MATNR	TYPE	ZCERCA_MATNR		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Material
I_MAKTG	TYPE	MAKTG		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Texto breve de mat. en mayúsculas ...
I_VMSTA	TYPE	VMSTA		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Status material específico cadena dist...
I_KDMAT	TYPE	KDMAT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Material de cliente
I_CLI_BAIXA	TYPE	CHAR01		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	'B' només clients de baixa

Fig. 30 – Paràmetres d'entrada de la RFC

- Codi client
- Nom del client
- Codi material
- Descripció del material
- Status del material (alta/baixa/provisional)
- Codi material del client
- Status del client (alta/baixa)

Tal com funciona la RFC, aquesta cerca materials i clients que compleixin amb els paràmetres d'entrada. El resultat de la cerca el retorna en 2 estructures diferents, una per materials i una altre per clients:

Bibl.funciones: ZCERCA_ARXIUS Visualizar

Módulo funciones ZCERCA_ARXIUS Activo

Atributos Import Export Modif. **Tablas** Excep. Cód.fte.

Nombre parám.	Tipificac.	Tipo ref.	Opcional	Texto breve
T_CLIENTS	LIKE	ZCERCA_CLIENTS	<input checked="" type="checkbox"/>	Cercador arxius. Clients
T_ARTICLES	LIKE	ZCERCA_ARTICLES	<input checked="" type="checkbox"/>	Cerca arxius. Articles
			<input type="checkbox"/>	
			<input type="checkbox"/>	

Fig. 31 – Estructura de sortida de la RFC

Estructura per retornar les dades dels clients:

Estructura	ZCERCA_CLIENTS	activo				
Descripción breve	Cercador arxius. Clients					
Atributos Componentes Ayuda p./Verif.entr. Campos de moneda/cantidad						
Tipo instalado 1 / 5						
Componente	Tp...	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve
KUNNR	<input type="checkbox"/>	KUNNR	CHAR	10	0	Nº de cliente 1
NAME1	<input type="checkbox"/>	NAME1	CHAR	30	0	Nombre
ORT01	<input type="checkbox"/>	ORT01	CHAR	25	0	Población
KTOKD	<input type="checkbox"/>	KTOKD	CHAR	4	0	Grupo de ctas.deudor
ESTAT	<input type="checkbox"/>	ZCERCA_ESTAT	CHAR	1	0	Estat

Estructura per retornar les dades dels materials:

Estructura	ZCERCA_ARTICLES	activo				
Descripción breve	Cerca arxius. Articles					
Atributos Componentes Ayuda p./Verif.entr. Campos de moneda/cantidad						
Tipo instalado 1 / 5						
Componente	Tp...	Tp.componente	Tipo de d...	Long.	Deci...	Descripción breve
MAINR	<input type="checkbox"/>	ZCERCA_MAINR	CHAR	18	0	Material
MAKTG	<input type="checkbox"/>	MAKTG	CHAR	40	0	Texto breve de mat. en mayúsculas para matchcodes
KUNNR	<input type="checkbox"/>	ZCERCA_KUNNR	CHAR	10	0	Cliente
VMSTA	<input type="checkbox"/>	VMSTA	CHAR	2	0	Status material específico cadena distribución
KDMAT	<input type="checkbox"/>	KDMAT	CHAR	35	0	Material de cliente

Si mirem ara un exemple de l'ús de la RFC, veurem com en funció dels paràmetres d'entrada, obtenim la sortida pertinent (en aquest cas, entrem el codi de client i ens retorna els materials d'aquest client i informació del client):

```

Test para grupo funciones      ZCERCA_ARXIUS
Módulo funciones              ZCERCA_ARXIUS
Mayúsculas/Minúsculas 

Tiempo ejec.: 2.267.834 Microsegundos

Sist.dest.RFC:

```

Parámetros p.import	Valor
I_KUNNR	125260
I_NAME1	
I_MAINR	
I_MAKTG	
I_VMSTA	
I_KDMAT	
I_CLI_BAIXA	

Tablas	Valor
T_CLIENTS	0 Entradas
Resultado:	1 Entrada
T_ARTICLES	0 Entradas
Resultado:	35 Entradas

Fig. 32 – Exemple de crida de la RFC

Editor estructuras: Visualizar T_CLIENTS desde entrada 1

Columna Entrada Metadatos

1 Entrada

KUNNR	NAME1	ORT01	KTOK	E
25260	LA FAGEDA S.COOP.C.LDA.	SANTA PAU	Z001	K

Fig. 33 Sortida de les dades de client de la RFC

Editor estructuras: Visualizar T_ARTICLES desde entrada 1

Columna Entrada Metadatos

35 Entradas

MA1NR	MAKTG	KUNNR	VM	KDMAT
18498.01	INTERIOR VERDES (MACETAS-LLAVORS- NAVIDA	125260	A	
19254.01	FONDO LLAVORS	125260	A	
19255.01	TAPA LLAVORS (VERDE)	125260	A	
19256.01	FONDO GENERIC	125260	A	
19257.01	TAPA GENERIC (AZUL)	125260	A	
19258.01	INTERIOR GENERIC (SIN AGUJERO)	125260	A	
19857.01	CARPETA BANC DE SANG 2010	125260	A	
19858.01	FONDO BANC DE SANG 2010	125260	A	
19859.01	TAPA BANC DE SANG 2010	125260	A	
19860.01	INTERIOR BANC DE SANG 2010	125260	A	
19951.01	TAPA BANC DE SANG ESPECIAL	125260	P	
19963.01	FONDO BANC DE SANG 2010 S/IMPRIMIR	125260	A	
19964.01	INTERIOR BANC SANG 2010 S/IMPRIMIR	125260	A	
21914.04	PACK GREC MADUIXA	125260	A	LAAUFA018
22045.01	DIPTIC PROMO MOD.IOGURT GREC	125260	A	
22046.03	FOLLETOS PROMO MOD.GELATS DE LLET	125260	A	
22897.03	C/MARIDADGE MELMELAD.CARN-PEIX CAT/CASTE	125260	A	MEAUFA800
22898.03	C/MARIDADGE MELMELADA ESMORZARS CAT/CAST	125260	A	MEAUFA600
23070.03	CALENDARIO DIPTICO 2015	125260	A	
26191.01	CAJA ARROZ DE MONTAÑA	125260	A	80002427
26192.01	CAJA ARROZ NEGRO	125260	A	80002405
26325.01	PACK NATILLAS XOCOLATA CONDIS	125260	A	LAAUFA020
26815.01	CAPSA 4 GELATS 4 GUSTOS IOGURT	125260	A	GEAU103
26816.01	CAPSA 4 GELATS 2 GUSTOS XOCOLATA-VAINILL	125260	A	GEAU102
27906.01	BASE MERMELADA DERECHA (CASTELLANO)	125260	A	
27907.01	BASE MERMELADA IZQUIERDA (CASTELLANO)	125260	A	
27908.01	INTERIOR MERMELADA IZQUIERDO (CASTELL.)	125260	A	
27909.01	INTERIOR MERMELADA DERECHO (CASTELLANO)	125260	A	
27910.01	TAPA MERMELADA DERECHA (CASTELLANO)	125260	A	
27911.01	TAPA MERMELADA IZQUIERDA (CASTELLANO)	125260	A	
27912.01	REFUERZO BASE MERMELADA (CASTELLANO)	125260	A	
27913.01	CAJA MARIDAJE MERMELADA DESAYUNOS	125260	P	
27914.01	C/ MARIDAJE MERMELADA QUESO CAST./INGLES	125260	A	
27915.01	CAJA DE 4 PACKS ARROSSOS	125260	A	
27936.01	C/ MARIDAJE MERMELADA QUESO CATALA/CAST.	125260	A	

Fig. 34 – Sortida de les dades de materials de la RFC

Com que el codi de la RFC és força extens, l'hem posat a l'annex del TFG, per la seva consulta posterior si es vol.

4.4.3.Exemples de crida del CDP

Fixem-nos ara com, des del CDP, es crida aquesta funció, i s'omplen les dades pertinents mostrant-se als usuaris. La part de codi que mostrarem és l'encarregada de mostrar els materials donat un client (el mateix exemple de crida de la RFC anterior):

Definició de la classe per crear la connexió a la RFC:

```
import javax.swing.JOptionPane;
import com.sap.conn.jco.JCoDestination;
import com.sap.conn.jco.JCoDestinationManager;
import com.sap.conn.jco.JCoException;
import com.sap.conn.jco.JCoFunction;

public class Connexio
{
    JCoFunction connexio;
    JCoDestination destination;
    MyDestinationDataProvider myProvider;

    public Connexio(String [] acces) throws Exception
    {
        connexio = null;
        //acces guarda els paràmetres de connexió, llegits de un txt de configuració. Usuari, sistema, IP, mandant...
        SapSystem system = new SapSystem(acces[0], acces[1],acces[2],acces[3], acces[4], acces[5],acces[6]);
        myProvider = new MyDestinationDataProvider(system);
        com.sap.conn.jco.ext.Environment.registerDestinationDataProvider(myProvider);
        destination = JCoDestinationManager.getDestination("sergi");
    }

    public JCoFunction connectar() throws Exception
    {
        connexio = null;
        try
        {
            //preparam la connexió a la RFC
            connexio = (destination.getRepository()).getFunction("ZCERCA_ARXIUS");
        }
        catch (JCoException e)
        {
            System.err.println("Error "+ "JCoException: " + e.getMessage());
        }
        return connexio;
    }

    public JCoDestination getDestination()
    {
        return destination;
    }

    public void desconnectar() throws Exception
    {
        com.sap.conn.jco.ext.Environment.unregisterDestinationDataProvider(myProvider);
    }
}
```

Part de la classe encarregada d'executar la RFC:

```
import com.sap.conn.jco.AbapException;
import com.sap.conn.jco.JCoFunction;
import com.sap.conn.jco.JCoTable;
import java.util.*;
import javax.swing.JOptionPane;

public class cerques
{
    String []dades; //guarda les dades de connexió a SAP
    Connexio con;
    JCoFunction connexio;
    ResourceBundle fitxerConfig;
    public cerques()
    {
        fitxerConfig = ResourceBundle.getBundle("conf/config", Locale.getDefault()); //nom del fitxer que guarda la configuració, config.properties
        //per obtenir les dades de connexió a SAP
        dades = new String[7];
        dades[0]="ALZAMORA";
        dades[1] = fitxerConfig.getString("ipSap");
        dades[2]="";
        dades[3] = fitxerConfig.getString("Mandant");
        dades[4] = fitxerConfig.getString("Sistema");
        dades[5] = fitxerConfig.getString("UsuariSap");
        dades[6] = fitxerConfig.getString("PasswdUsuari");
    }
}
```

Mètode que a partir del client, obté la llista d'articles:

```

public Vector cercarArticleXClient(String vigencia, int client) throws Exception
{
    Vector articles = new Vector(); //guardarà el resultat de la cerca
    String codiArticleJunt = "";
    String codiArticleJunt2;
    String codiClientRetornat="";
    String [] aux;

    if (vigencia == "")
        vigencia = "B";
    else
        vigencia = "";

    try
    {
        try
        {
            con = new Connexio(dades); //connectem a SAP
            connexio = con.connectar(); //fem la connexió a la RFC
        }
        catch ( Exception e)
        {
            System.out.println("exception connexio");
            System.out.println(e.toString());
        }
        connexio.getImportParameterList().setValue("I_KUNNR", client);
        //executem la RFC amb els paràmetres pertinents, en aquest cas el codi client
        connexio.getImportParameterList().setValue("I_VMSTA", vigencia);
        //executem la RFC amb els paràmetres pertinents, en aquest cas l'estatus del material (alta o baixa)

        try
        {
            connexio.execute(con.getDestination()); //executem la RFC
        }
        catch (AbapException e)
        {
            System.out.println(e.toString());
        }

        JCoTable tArticles = connexio.getTableParameterList().getTable("T_ARTICLES");
        //obtenim el contingut de l'estructura que guarda les dades demanades
        for (int loop = 0; loop < tArticles.getNumRows(); loop++) //recorrem l'estructura
        {
            codiArticleJunt = (tArticles.getValue("MATNR")).toString(); //material
            codiArticleJunt2 = codiArticleJunt.replace(".", "");
            aux = codiArticleJunt2.split(";");

            if (aux[1].length() == 1)
                aux[1]="0"+aux[1];

            int strMatnr1 = Integer.parseInt(aux[0]); //separem el material
            int strMatnr2 = Integer.parseInt(aux[1]); //separem la versió del material
            codiClientRetornat=tArticles.getValue("KUNNR").toString(); //codi client
            String strvmsta = ((tArticles.getValue("VMSTA")).toString()); //status del client
            String strmaktg = (tArticles.getValue("MAKTG")).toString(); //descripció del material
            int strkumnr = Integer.parseInt((tArticles.getValue("KUNNR")).toString());
            String strkdmnt = (tArticles.getValue("KDMAT")).toString(); //material del client

            Article art = new Article(strMatnr1,strMatnr2,strvmsta,strmaktg,strkumnr,strkdmnt, "");
            //creem l'objecte que guardarà el material trobat
            articles.addElement(art);
            //l'afegim a la llista

            tArticles.nextRow();
        }
    }
    catch (Exception e)
    {
        con.disconnectar();
        throw new Exception("ERROR en el cercar article per client.\nArticle amb problemes: " + codiArticleJunt
            + "\nCodi client: " + codiClientRetornat + "\nError SAP:" + e.getMessage());
    }
    con.disconnectar(); //desconnectem

return articles; //retorna l'objecte amb tots els materials trobats

```


Finalment, el mètode per omplir la taula de resultats de la interfície del programa:

```

public void emplenarTaulaArticlesXClient() //omplim la taula de resultats
{
    try {
        String avigen = "";

        int codcliInt = Integer.parseInt(codcliText.getText().trim());
        //el passem a enter i treiem espais en blanc
        String articleText = descartText.getText().trim().toUpperCase();
        //agafem el text de descripcio de l'article i el passem a majuscules
        Vector articles = cerca.cercarArticleXClient(avigen, codcliInt);
        //cerquem els materials del client amb el mètode corresponent

        for (int i = 0; i < articles.size(); i++) {
            //recorrem l'objecte amb els materials retornats
            String codi = String.valueOf( ( (Article) articles.get(i)).getCodiarticle());
            String versio = String.valueOf( ( (Article) articles.get(i)).getVersio());
            String client = String.valueOf( ( (Article) articles.get(i)).getCodiclient());
            int codiINT = Integer.parseInt(codi);
            int versioINT = Integer.parseInt(versio);
            int clientINT = Integer.parseInt(client);
            String vigencia = ( (Article) articles.get(i)).getVigencia();
            String descripcio = ( (Article) articles.get(i)).getDescripcio();
            String codartcli = ( (Article) articles.get(i)).getCodartcli();
            String busqueda = ( (Article) articles.get(i)).getAbusqu();

            String disponibilitat = ExisteixArticleVersio(clientINT, codiINT, versioINT);
            //comprova que existeix el fitxer al repositori
            String ruta = "";

            if (disponibilitat == "S") { //Obtenim la ruta on hi ha l'arxiu, si aquest existeix
                ruta = obtenirPath(clientINT, codiINT, versioINT);
            }

            Object[] fila = {codi.concat("." + versio), vigencia, client, descripcio, codartcli, busqueda, disponibilitat, ruta };
            //preparem l'objecte que guardarà el registre amb la informació obtinguda
            dtma.addRow(fila); // Afegim el registre a la taula
        }
    }
    catch (NoSuchElementException ex) {
        JOptionPane.showMessageDialog(this, "\nArticle/ver incorrecte\n\n" , "Error d'entrada de dades",
            JOptionPane.WARNING_MESSAGE);
        ex.printStackTrace();
    }

    catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(this, "\nFormat de número incorrecte\n\n" , "Error d'entrada de dades",
            JOptionPane.WARNING_MESSAGE);
        ex.printStackTrace();
    }

    catch (Exception ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage(), "Error",
            JOptionPane.ERROR_MESSAGE);
        ex.printStackTrace();
    }
}

```

Si donem una ullada a la interfície, això és el que es veuria:

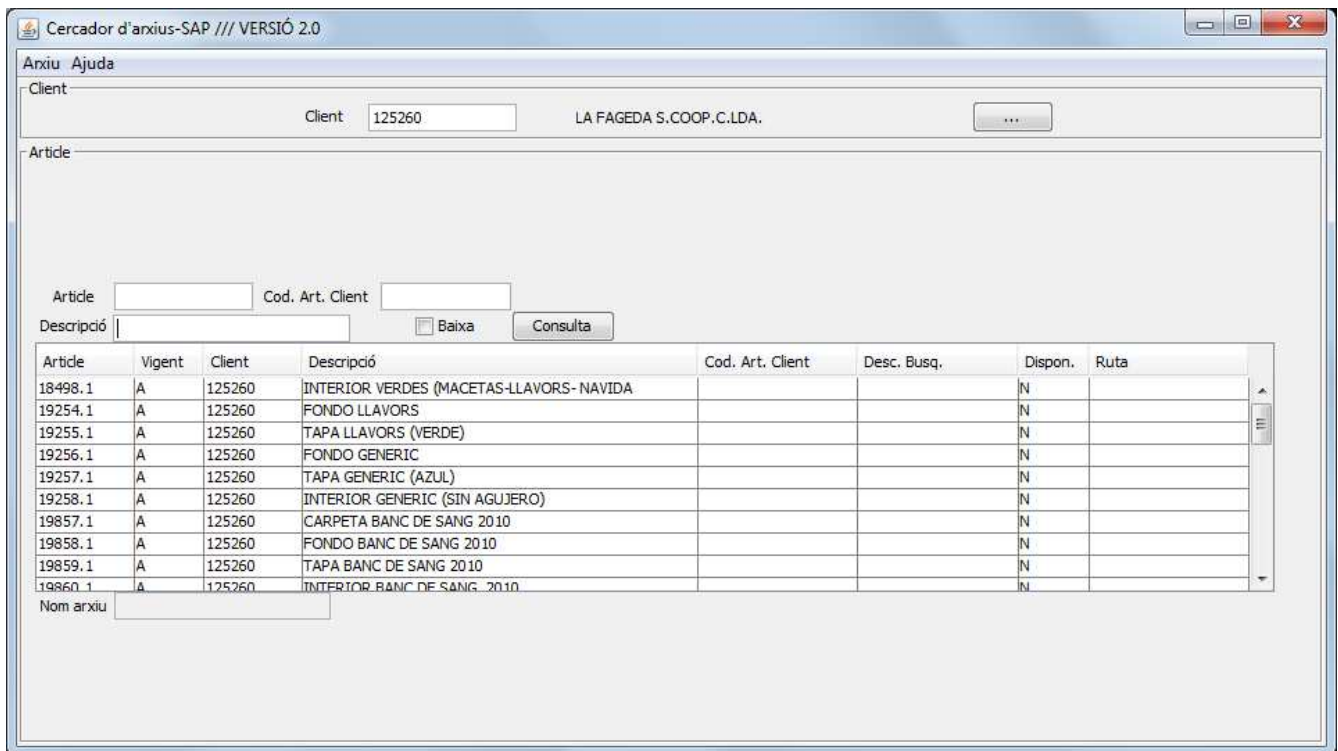


Fig. 35 – Interfície del CDP amb les dades rebudes

4.5. Integració de l'aplicació per l'embalatge i etiquetat (SEE)

Quan una fabricació s'ha d'entregar a un client, la majoria de les vegades cal emmagatzemar tot el material en palets. Com que el material no es pot posar directament al palet, cal guardar aquest producte acabat en caixes d'embalatge. Aquestes caixes d'embalatges són les que es van omplint amb el producte acabat i les que es van apilant en els palets que posteriorment s'entregaran al client.

Aquestes caixes d'embalatge no es poden entregar sense cap identificació, donat que es necessita una traçabilitat per tal de poder esbrinar, si hi ha alguna reclamació, de quina fabricació es tracta, quan es va fer, a quina màquina es va fer, etc. Aquesta traçabilitat s'aconsegueix codificant aquestes caixes d'embalatge mitjançant un número de caixa i un codi de barres. Aquesta informació, evidentment s'ha d'obtenir del SAP, i és l'aplicació SEE l'encarregada de extreure la informació de la fabricació i enviar-la als aparells que imprimeixen aquestes dades a la caixa d'embalatge (a mesura que les caixes d'embalatge van circulant per la cinta transportadora, van passant per davant d'uns injectors de tinta que van pintant les dades més rellevants d'aquella fabricació, com poden ser el client, la data, el número de caixa per la

traçabilitat posterior, la descripció del material, etc.). És una aplicació comercial estàndard, amb les mínimes adaptacions per Alzamora.. Si recordem, les dades que l'aplicació necessita de SAP, entre d'altres, són:

- Materials
- Clients
- Ordres de fabricació

4.5.1. Procés de definició de les connexions

En les reunions amb l'equip tècnic de l'aplicació i el partner de SAP, es va decidir que la millor forma que l'aplicació SEE tenia per obtenir les dades de la fabricació era mitjançant funcions RFC. Es va descartar llegir la informació directament del SAP i l'ús de fitxers txt perquè calia modificar l'aplicació en excés, a banda de que el rendiment no hagués estat l'òptim. A part, aquesta aplicació al ser relativament estàndard, ja suportava aquest mètode d'obtenir informació del SAP donat que està implantada a d'altres indústries semblants, per la qual cosa l'adaptació més gran va ser per part de SAP i no pas del propi SEE.

En resum, la forma de treballar del programa és la següent:

1. Al començar la feina d'embalar el producte acabat, l'operari carrega al SEE les dades de la fabricació (ordre de fabricació i material)
2. El SEE valida a SAP que les dades són correctes
3. Si ho són, extreu de SAP les dades que s'han de pintar a les etiquetes: material, descripció, client, ordre de fabricació...



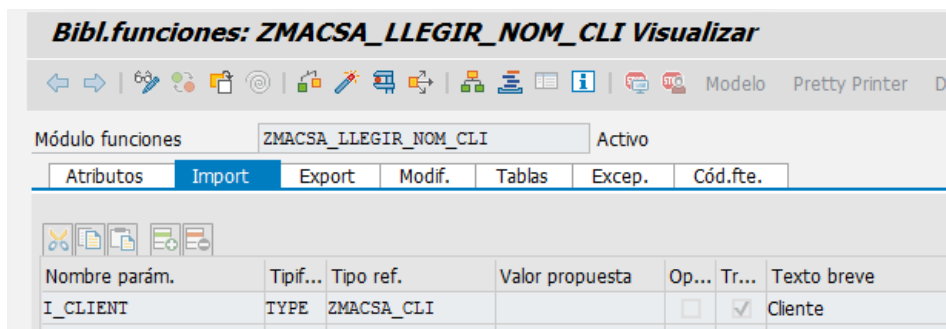
Fig. 36 – Flux d'informació entre SAP i el SEE

4.5.2. Definició de la RFC

En el cas del SEE, les RFC implementades són molt simples donat que la gran majoria són cerques directes o comprovacions, per tant, mostrarem les que tenen més pes dins l'aplicació.

4.5.2.1. ZMACSA_LLEGIR_NOM_CLI: obtenir el nom del client

Aquesta funció el que permet és, a partir del codi de client, obtenir el seu nom. El paràmetre d'entrada de la funció és:

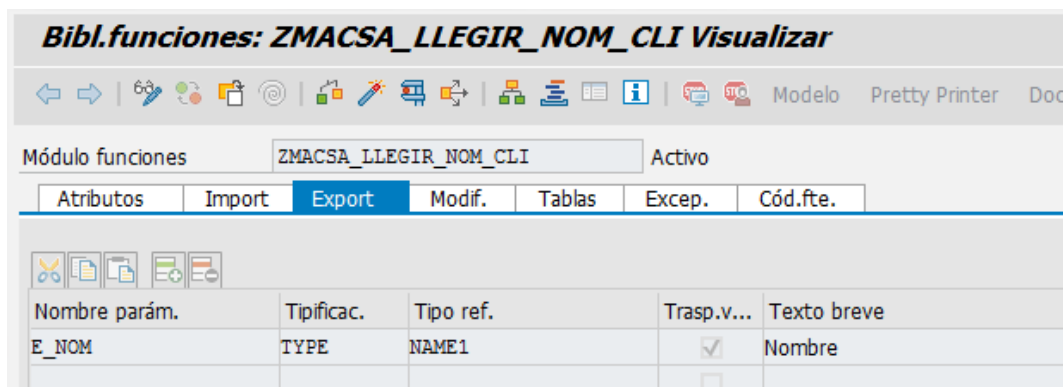


Nombre parám.	Tipif...	Tipo ref.	Valor propuesta	Op...	Tr...	Texto breve
I_CLIENT	TYPE	ZMACSA_CLI		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Cliente

Fig. 37 – Paràmetre d'entrada de la funció

- Codi de client

A partir del codi, el que fa és anar a llegir el nom directament a la taula, comprovant que efectivament el client existeix. Per retornar-lo ho fa amb el paràmetre de sortida:



Nombre parám.	Tipificac.	Tipo ref.	Trasp.v...	Texto breve
E_NOM	TYPE	NAME1	<input checked="" type="checkbox"/>	Nombre

Fig. 38 – Paràmetre de sortida de la funció

Si executem la RFC, el que obtindrem serà:

Test para grupo funciones ZMACSA
 Módulo funciones ZMACSA_LLEGIR_NOM_CLI
 Mayúsculas/Minúsculas

Tiempo ejec.: 3.061 Microsegundos

Sist.dest.RFC:

Parámetros p.import	Valor
I_CLIENT	471110

Parámetros export	Valor
E_NOM	POMPADOUR IBERICA, S.A.

Fig. 39 – Exemple de crida de la RFC

Si mirem el codi font:

```
FUNCTION ZMACSA_LLEGIR_NOM_CLI .
data : l_kunnr type kunnr .
select single name1 into e_nom " select directe a la taula de clients
from kna1 where kunnr = l_kunnr .
ENDFUNCTION .
```

4.5.2.2. ZMACSA_LLEGIR_CLI_ART: obtenir la descripció del material i del material del client

Aquesta funció el que permet és, a partir del codi de material, obtenir la seva descripció i el codi de material del client. El paràmetre d'entrada de la funció és:

Bibl.funciones: ZMACSA_LLEGIR_CLI_ART Visualizar

Módulo funciones ZMACSA_LLEGIR_CLI_ART Activo

Atributos Import Export Modif. Tablas Excep. Cód.fte.

Nombre parám.	Tipif...	Tipo ref.	Valor propuesta	Op...	Tr...	Texto breve
I_CAJART	TYPE	ZMACSA_MATNR		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Material

Fig. 40 – Paràmetre d'entrada de la funció

- Codi de material

A partir del codi de material, el que fa és una cerca directe a taula per obtenir la descripció del material i el codi d'article del client. Ho retorna amb 3 paràmetres:

Bibl.funciones: ZMACSA_LLEGIR_CLI_ART Visualizar

Módulo funciones: Activo

Atributos Import **Export** Modif. Tablas Excep. Cód.fte.

Nombre parám.	Tipificac.	Tipo ref.	Trasp.v...	Texto breve
E_CLIENT	TYPE	ZMACSA_CLI	<input checked="" type="checkbox"/>	Cliente
E_ART_CLI	TYPE	MATNR_KU	<input checked="" type="checkbox"/>	Número de material del cliente
E_NOM_ART	TYPE	MAKTX	<input checked="" type="checkbox"/>	Texto breve de material

Fig. 41 – Paràmetres de sortida de la funció

Si executem la RFC, el que obtindrem serà:

```
Test para grupo funciones      ZMACSA
Módulo funciones              ZMACSA_LLEGIR_CLI_ART
Mayúsculas/Minúsculas 

Tiempo ejec.:  33.844 Microsegundos

Sist.dest.RFC:
[                               ]
```

Parámetros p.import	Valor
I_CAJART	22505.01

Parámetros export	Valor
E_CLIENT	495028
E_ART_CLI	PM484
E_NOM_ART	FDA. REC.TRAD. COCO & PIÑA CATALA

Fig. 42 – Exemple de crida de la RFC

Si mirem el codi font:

```
FUNCTION ZMACSA_LLEGIR_CLI_ART.

data : l_matnr type matnr.
data : l_objnum type objnum.
data : t_caract like clobjdat occurs 0 with header line.
data : t_class like sclass occurs 0 with header line.
```

```
clear : e_client , e_art_cli.
refresh t_caract.
```

```
select single maktx into e_nom_art "obtenim la descripció del material
from makt where matnr = l_matnr and
           spras = sy-langu.
```

```
l_objnum = l_matnr.
```

"la següent funció obté el codi de client del material

```
call function 'CLAF_CLASSIFICATION_OF_OBJECTS'
```

```
exporting
```

```
class           = 'CLAS_MAT_PA'      "producto acabado
classtext       = 'X'
classtype      = '001'
features       = 'X'
language       = sy-langu
object         = l_objnum
objecttable    = 'MARA'
key_date       = sy-datum
initial_charact = ' '
change_service_clf = 'X'
```

```
tables
```

```
t_class        = t_class
t_objectdata   = t_caract
```

```
exceptions
```

```
no_classification = 1
no_classtypes    = 2
invalid_class_type = 3.
```

```
read table t_caract with key atnam = 'ZCODCLIEN'.
```

```
if sy-subrc eq 0.
```

```
  e_client = t_caract-auspl.
```

```
  select single kdmat into e_art_cli "si ha trobat el client, obtenim el
  from knmt where vkorg = '350' and      "codi article del client
                vtweg = '10' and
                matnr = l_matnr.
```

```
endif.
```

4.5.3. Exemples de crida del programa

Com ja hem dit abans, l'aplicació és de un proveïdor de software comercial orientat a la indústria del packaging, per la qual cosa no disposem del font del programa directament. Els hi hem demanat que ens proporcionessin parts del codi font on es veïés a forma d'exemple, les crides que fan amb les RFC comentades anteriorment.

En quant a la part de preparació de la connexió:

```

Dim mFunctions As Object
Dim eSAPCONF As SAPCONF

Private Sub Class_Initialize() 'inicialitzem
    Dim Index As Integer, Field As Integer
    Dim Value As String, Cadena As String

    Open App.Path + "\Sapconfig.txt" For Input As #1
    If LOF(1) = 0 Then 'els paràmetres de connexió estan en un fitxer de text
        Close #1
        RaiseEvent SAPDataFailed("SAP: Fallo al recuperar los datos de configuración del txt")
    End If
    Field = 0
    Do While Not EOF(1)
        Line Input #1, Cadena

        Index = InStr(1, Cadena, "=")
        If Index > 0 Then
            Value = Mid(Cadena, Index + 1, Len(Cadena) - Index) 'paràmetres de connexió com usuari, sistema,
            Select Case Field 'mandant, password...
                Case 0
                    eSAPCONF.sSystem = Value
                Case 1
                    eSAPCONF.sUser = Value
                Case 2
                    eSAPCONF.sPwd = Value
                Case 3
                    eSAPCONF.sClient = Value
                Case 4
                    eSAPCONF.sLang = Value
                Case 5
                    eSAPCONF.sSysNumber = Value
                Case 6
                    eSAPCONF.sServer = Value
            End Select
            Field = Field + 1
        End If
    Loop
    Close #1
    Connect
End Sub

Private Function Connect() As Boolean
    Dim sErr As String
    On Error GoTo ErrorCon
    Connect = False

    Set mFunctions = CreateObject("SAP.Functions") 'creem l'objecte amb els paràmetres obtinguts abans
    mFunctions.Connection.System = eSAPCONF.sSystem
    mFunctions.Connection.user = eSAPCONF.sUser
    mFunctions.Connection.Password = eSAPCONF.sPwd
    mFunctions.Connection.Client = eSAPCONF.sClient
    mFunctions.Connection.Language = eSAPCONF.sLang
    mFunctions.Connection.SystemNumber = eSAPCONF.sSysNumber
    mFunctions.Connection.ApplicationServer = eSAPCONF.sServer

    Connect = True
    Exit Function

ErrorCon:
    MsgBox ("Error de Conexión a SAP")

End Function

```


Exemple de crida de la RFC ZMACSA_LLEGIR_NOM_CLI per obtenir el nom del client:

```
' obtenir el nom del client
Friend Function ClientName(Client As String) As String
  Dim oParam1 As Object
  Dim mMyFunc As Object
  Dim sError As String

  On Error GoTo ErrorOnGet

  If mFunctions.Connection.Logon(0, True) <> True Then
    RaiseEvent SAPDataFailed(sError)
    Exit Function
  End If

  Set mMyFunc = mFunctions.Add("ZMACSA_LLEGIR_NOM_CLI") 'nom de la RFC

  Set oParam1 = mMyFunc.exports("I_CLIENT") 'paràmetre d'entrada codi client

  oParam1 = Client

  If mMyFunc.Call Then 'la cridem i obtenim el nom
    ClientName = mMyFunc.imports("E_NOM")
  Else
    ClientName = ""
    RaiseEvent SAPDataFailed("SAP: Respuesta vacía a la consulta.")
  End If

  mFunctions.Connection.LOGOFF 'desconnectem

  Exit Function
```

Exemple de crida de la RFC ZMACSA_LLEGIR_CLI_ART per obtenir el codi client i la descripció del material i article client:

```
'Devuelve el nombre del material del cliente, el cliente y descripción material
Friend Function GetArticleClient(ByRef Datos As SAP_CIENT_MAT) As Boolean
  Dim oParam1 As Object
  Dim mMyFunc As Object
  Dim sError As String

  On Error GoTo ErrorOnGet

  If mFunctions.Connection.Logon(0, True) <> True Then
    RaiseEvent SAPDataFailed(sError)
    Exit Function
  End If

  Set mMyFunc = mFunctions.Add("ZMACSA_LLEGIR_CLI_ART") 'cridem la RFC

  Set oParam1 = mMyFunc.exports("I_CAJART") 'paràmetre d'entrada és el material

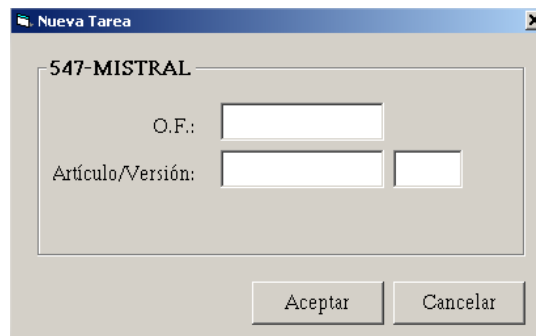
  oParam1 = Datos.Article

  If mMyFunc.Call Then 'cridem la RFC i obtenim les dades
    Datos.Cliente = mMyFunc.imports("E_CLIENT")
    Datos.MaterialCliente = mMyFunc.imports("E_ART_CLI")
    Datos.Descripcion = mMyFunc.imports("E_NOM_ART")
    GetArticleClient = True
  Else
    GetArticleClient = False
    RaiseEvent SAPDataFailed("SAP: Respuesta vacía a la consulta.")
  End If

  mFunctions.Connection.LOGOFF

  Exit Function
```

Aquesta última es crida quan l'operari entra ordre de fabricació i article al començar la feina per validar les dades:



Les dades es mostren a la pantalla de l'aplicació un cop obtingudes:

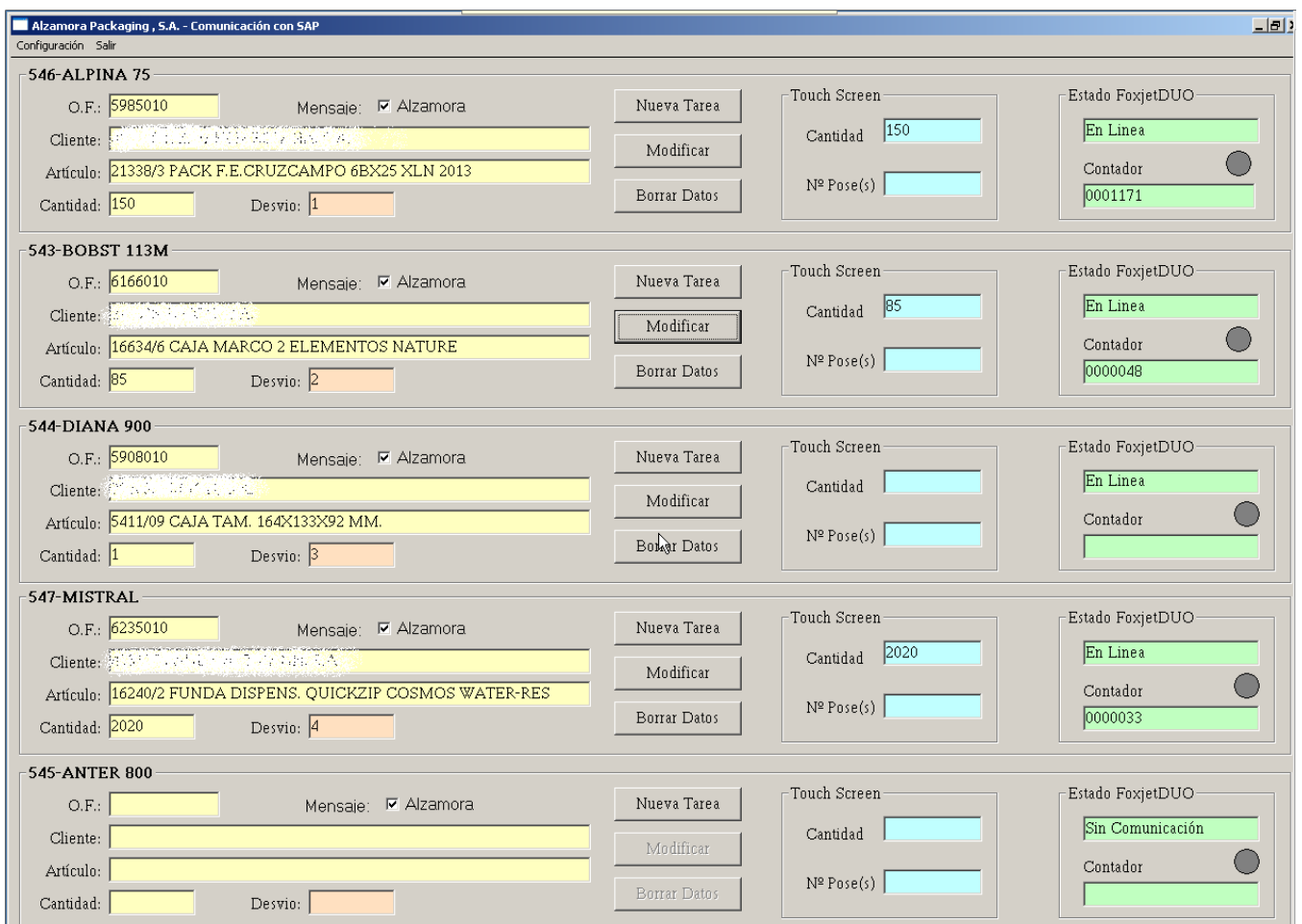


Fig. 43 – Interfície del SEE amb les dades obtingudes de les RFC

4.6. Integració del sistema de magatzems (SGM)

Per últim tenim el sistema encarregat de gestionar el magatzem automàtic. L'hem deixat per l'últim perquè la integració més complexa de totes les que vam haver de fer. Aquest és un magatzem completament robotitzat capaç d'emmagatzemar més de 8000 palets en els seus més de 10 pisos. Aquest magatzem és capaç de guardar tant matèria prima, com producte acabat com qualsevol altre material. El magatzem, quan es va adquirir, venia amb un servidor i un software que era l'encarregat de gestionar els materials del seu interior, i en general, aquest software és el que conté tota la intel·ligència de l'aplicació. Per exemple, és l'encarregat de balancejar la col·locació dels palets per tal que el pes de l'estructura estigui equilibrat, és l'encarregat d'agrupar palets de forma més pròxima en funció de la rotació que hi hagi d'aquell material, és l'encarregat de gestionar la sortida de material que fa més temps que és a dins, etc. Tota aquesta gestió pròpia de un magatzem, aquest sistema és el que ho fa.

Malauradament, aquest software és completament tancat i no disposa de masses característiques que el permetin interfasar fàcilment amb altres sistemes d'informació, per la qual cosa no es va poder fer cap millora respecta la integració inicial. A grans trets, s'ha mantingut la mateixa integració que hi havia amb l'antic sistema AS400, amb el sistema SAP, donat que ens hem d'imaginar que el SGM és una gran caixa negra, i només en coneixem les comandes bàsiques per fer-li fer accions (aquestes comandes se li envien mitjançant fitxers de text pla contra *hotfolders*). En aquest sentit, les dades mestres que els 2 sistemes necessiten són:

- Materials
- Ordres de fabricació

4.6.1. Procés de definició de les connexions

En les reunions amb el partner de SAP, i després de mostrar-los-hi la integració antiga que s'havia fet amb l'AS400, ens van proposar de refer l'antiga integració via fitxers de text, i convertir-la a una integració més moderna: no va ser possible. Els motius principals pels quals modernitzar la integració no era viable són:

- Els enginyers que van dissenyar l'adaptació del nostre magatzem a la forma de treballar d'Alzamora ja no estaven a l'empresa, i la gent que hi havia actualment no veien massa clar modificar l'aplicació (eren més partidaris de refer-la de nou). Com que no volíem que això es convertís en un segon projecte paral·lel, vam descartar la opció.

- No tenien la documentació tècnica del nostre projecte, per tant, per entendre què feia la nostre aplicació havien de seguir pràcticament totes les línies del codi.
- La última opció va ser que, en comptes que fos el SGM qui portés tota l'algorísmica de gestió del magatzem, fos el propi SAP qui ho fes. Aquesta opció era una bestiesa donat que havíem de replicar totes les funcionalitats i algorismes de gestió de l'SGM a SAP (uns algorismes que no podíem conèixer donat que no disposàvem del codi font i ni l'empresa instal·ladora sabia què feia)

Davant d'aquest panorama, només quedava la opció de transferir la funcionalitat de l'antic AS400 a SAP, és a dir, que el SAP fos l'encarregat d'enviar els fitxers de text a l'SGM per tal de fer-li fer accions, reaprofitant l'antiga integració (recordar que el SGM és com una caixa negra, que en coneixem les entrades i les sortides, i res més).

Si ens centrem al funcionament de la integració, a grans trets podem dir que al SGM se li pot demanar que realitzi accions mitjançant fitxers de text amb una estructura concreta (una estructura per cada acció: treure material, entrar material). Per cada acció que se li demani, genera un txt de confirmació per tal d'informar de l'acció realitzada. Aquests fitxers de text es guarden en una *hotfolder* (carpeta compartida que es va comprovant cada cert temps si hi ha o no contingut) en el seu servidor, i cada minut, es comprova si hi ha accions per realitzar (si hi ha fitxer de text a processar). Esquemàticament:

1. Un usuari de SAP demana al magatzem que tregui 10000 unitats d'un material de una fabricació.
2. El SAP genera un txt amb la petició i el deixa a una *hotfolder* del servidor del SGM.
3. El SGM agafa el txt, l'interpreta, i entén que se li està demanant que tregui una quantitat d'un material/fabricació
4. El SGM realitza el seu procés i algorismes, i finalment veu que ha de treure 2 palets de material
5. El SGM treu els 2 palets del magatzem automàtic.
6. Quan l'operari els retira de la boca de sortida, el SGM genera un fitxer de text informant que ha tret les 10000 unitats d'aquella fabricació, i el guarda en una altra *hotfolder* per tal que el SAP el pugui anar a processar

7. El SAP agafa aquest txt, el processa, i entén que d'aquell material/fabricació han sortit del magatzem 10000 unitats, per tant, ha de fer un moviment d'estoc del magatzem a fora del magatzem.

Aquest és un esquema molt general del funcionament de l'aplicació, però es pot complicar molt més: casos en que no surt un pallet sencer sinó que ens en quedem una part (*picking*), casos en que el SAP no té suficient estoc informat en el seu magatzem i no pot fer moviments d'estoc de magatzem, casos en que hi ha un error en el traspàs, etc. La gestió d'errors és quasi tant gran com la integració, per tant, ens centrarem només en els casos simples, donat que la documentació de la integració amb l'SGM seria tant extensa com l'SGM, i aquest no és l'àmbit del TFG.



Fig. 44 – Flux d'informació entre el SAP i l'SGM

4.6.2.Exemples de crida del SGM

Tal com hem comentat abans, la petició d'accions (i les seves confirmacions) es fan mitjançant fitxers de text pla amb una estructura concreta. En quant a l'estructura, podem dir que és un fitxer de text on cada posició del document significa una dada o una altra. A continuació presentem alguns exemples de fitxers txt per realitzar certes accions (en funció de si és matèria prima o producte acabat tenen una estructura o una altra, però a tall didàctic mostrarem un exemple de petició de sortida de producte acabat, donat que és la sol donar més joc al procés).

4.6.2.1. Petició de sortida de material del magatzem

Aquest fitxer de text serveix per fer una petició al magatzem de sortida de material. Aquesta petició es realitza a SAP amb unes pantalles a mida, on s'introdueixen dades del material i fabricació a fer sortir del magatzem. Un cop validades les dades, es crea el fitxer de text amb la informació pertinent. L'estructura és la següent (mostrem l'estructura tallada en varis troços perquè no hi cap tot en una sola línia; sobre l'estructura hi ha un exemple real de crida, per veure què va a cada posició):

En l'exemple anterior, el SGM ens ha tret 700 unitats del material 22521.01 de la fabricació (lot) 190, en el palet amb matrícula 013000951. L'ha tret el 23/07/2013 a les 16:17:16 amb un pes de 209 KG.

4.6.2.3. Alta de referències de material al SGM

Aquest fitxer de text serveix per donar d'alta al SGM al mestre de materials, referències noves. Quan al SAP es crea una partida de matèria prima o quan es comencen a generar palets de producte acabat, s'envia aquest fitxer de text al SGM per tal que també el doni d'alta al sistema. L'estructura és la següent (mostrem l'estructura tallada en varis troços perquè no hi cap tot en una sola línia; sota l'estructura hi ha un exemple real de crida, per veure què va a cada posició):

En l'exemple anterior, hem donat d'alta al SGM la referència 21206.01 – “CABALLETE CABEZA DE CERDO 100G 8 LONCHAS TIPO D950”. Al SGM ja el veuríem donat d'alta:

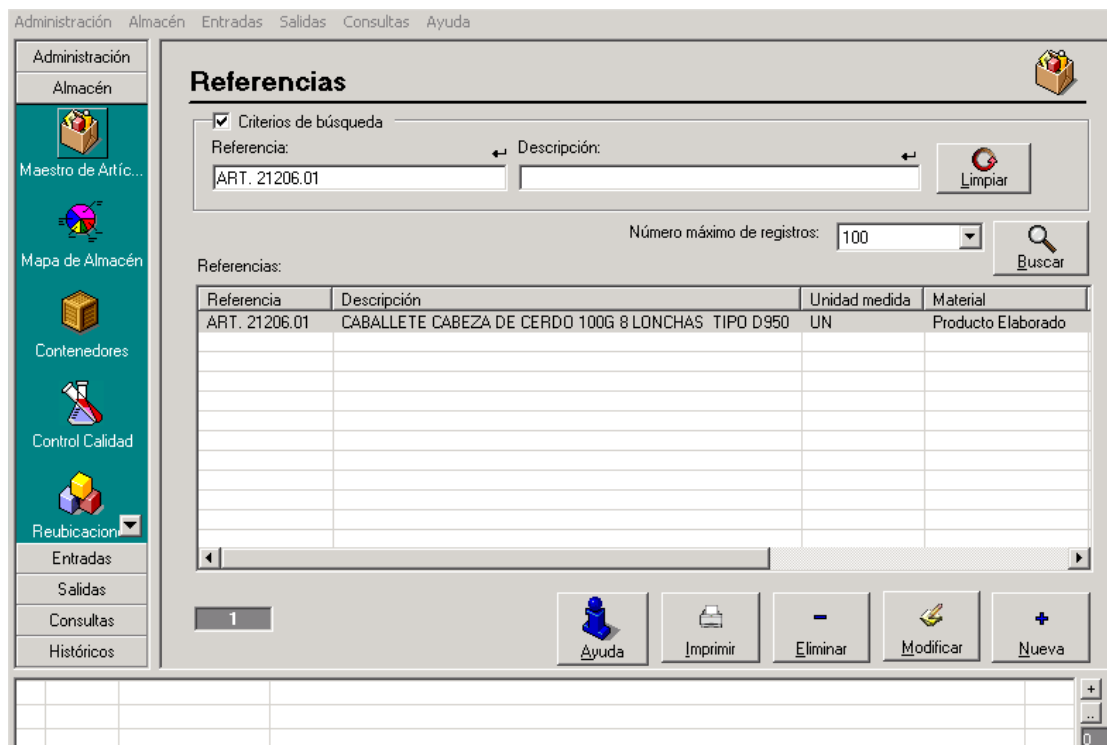


Fig. 46 – Mestre de referències del SGM

5. Millores futures

En el cas de millores futures, creiem que en podríem aplicar dues de clares als diferents SI:

- Integració del SGM (sistema de gestió de magatzems)
- Integració amb el GDB (generador de banderoles)

En el primer cas, i si recordem, la integració es realitza a través de fitxers de text. Com ja s'ha comentat al capítol, la integració es va fer amb fitxers de text per la limitació pròpia del sistema informàtic que gestiona el magatzem (massa antic). No es va valorar al seu moment actualitzar-lo ja que s'hagués convertit en un segon projecte paral·lel, cosa que no volíem. En quant a les millores, la més clara és la de substituir els fitxers de text per funcions RFC per tal d'enviar les peticions de sortida del magatzem, i per tal de gestionar les confirmacions

d'entrades i sortides, evitant així l'ús de fitxers que compliquen en excés la gestió de la interconnexió. Caldria, però, modernitzar el sistema d'informació que gestiona el magatzem, fet que implicaria una revisió total de l'aplicació donat que la que tenim actualment està discontinuada i el proveïdor no en dóna suport, fet que suposa l'adaptació des de zero de l'actual aplicació distribuïda.

En el segon cas, estaríem parlant de no fer servir una aplicació externa per la generació de les banderoles especials, sinó de fer servir el propi SAP per crear-les. La millora en si no implicaria una millora de la productivitat, sinó que seria una millora en el sentit conceptual (el propi SAP ja disposa d'eines per la generació de formularis amb codis de barres e imatges), ja que tindriem integrat en un sol entorn de programació (ABAP-*smartforms*), la creació de les banderoles, tant estàndards com especials, evitant així haver de fer ús de un llenguatge de programació "extra" com és el Delphi, i a part, ens estalviaríem de tenir una integració per poder cridar aquesta aplicació cada cop que es fa una banderola estàndard.

Per últim, tot i que no estaria centrada en cap integració de les ja fetes, seria la integració de les aplicacions que es va decidir no integrar:

- Sistema de preimpresió (SDP):
- Sistema pel disseny de maquetes (SDM):

En el primer cas, com que no existia quan es va implantar el SAP, no es va poder integrar; caldria doncs, interconnectar-lo amb SAP.

En el segon cas, caldria adquirir la capa de *middleware* que fa falta per tal de fer la interconnexió, que no es va adquirir per motius de temps i econòmics.

6. Conclusions

En quant a les conclusions, és difícil extreure sentències finals concretes dels temes tractats, però a grans trets, el que si que en podem extreure, és que la integració entre els diferents sistemes d'informació és un element vital en qualsevol projecte d'implantació de qualsevol ERP, i relacionat amb això, també ens podem atrevir a dir que sempre que es pugui, es traspassi la funcionalitat de les aplicacions a integrar a l'ERP principal. Recalquem "si es pot" perquè som conscients que moltes vegades és impossible, però en cas que si que es pugui, és un fet molt positiu, perquè el procés es simplifica, les incidències es redueixen, la complexitat es

redueix i en general, tota la implantació i la instal·lació final de l'ERP queda molt més neta, ordenada i conceptualment més correcte.

Evidentment no estem dient que les integracions no són bones, el que passa que amb l'experiència adquirida en aquest projecte, podem dir que a vegades aporten més complexitat o problemes, que no pas beneficis a la instal·lació de l'ERP.

En quant als objectius del TFG, podem dir que els hem assolit tots, donat que hem tractat tots els temes que volíem desenvolupar i hem aprofundit i explicat tots els plantejats a l'inici del TFG:

- Mostrar la naturalesa dels SI actuals a l'empresa i les seves característiques generals i particulars
- Definir les integracions a realitzar amb d'altres SI
- Descriure els problemes principals que ens trobem per aconseguir-ho
- Aportar i descriure les solucions tècniques a aquestes integracions
- Explicar les característiques tècniques de les eines utilitzades en les integracions: RFC's, connexions a BD directament, etc.
- Mostrar amb exemples reals de programació, les solucions d'integració creades: llenguatge JAVA, ABAP, Delphi, etc.

Respecte la planificació podem estar molt contents, donat que hem assolit les fites marcades quasi 3 setmanes abans del previst, donat que s'ha hagut d'anar a un ritme alt donada la compaginació de la realització del TFG amb la feina personal, cosa que ens ha fet "esperonar" per realitzar les tasques no només quan tocaven, sinó molt abans, permetent disposar de un temps de marge força elevat per repassar el TFG i millorar aquelles parts que no han quedat prou clares.

Finalment, i lligat amb el que ja s'ha exposat al capítol 5, ens hagués agradat afegir en aquest TFG, el procés d'integració de certes aplicacions que en el seu moment no es van poder integrar. A part, ens hagués agradat també poder mostrar el funcionament de les RFC dins les integracions en viu, per exemple amb vídeos, tot i que probablement s'escapés una mica de l'àmbit del TFG, al ser aquest eminentment realitzat via escrita.

7. Annex

En l'apartat de l'annex, hi hem inclòs el codi de la funció RFC ZCERCA_ARXIUS, massa extens per la inclusió en l'apartat 4.4.2.1. Mostrem doncs el font:

```

FUNCTION zcerca_arxius.
* "-----
* " * "Interfase local
* "   IMPORTING
* "     VALUE(I_KUNNR) TYPE  ZCERCA_KUNNR OPTIONAL
* "     VALUE(I_NAME1) TYPE  NAME1 OPTIONAL
* "     VALUE(I_MATNR) TYPE  ZCERCA_MATNR OPTIONAL
* "     VALUE(I_MAKTG) TYPE  MAKTG OPTIONAL
* "     VALUE(I_VMSTA) TYPE  VMSTA OPTIONAL
* "     VALUE(I_KDMAT) TYPE  KDMAT OPTIONAL
* "     VALUE(I_CLI_BAIXA) TYPE  CHAR01 OPTIONAL
* "   TABLES
* "     T_CLIENTS STRUCTURE  ZCERCA_CLIENTS OPTIONAL
* "     T_ARTICLES STRUCTURE ZCERCA_ARTICLES OPTIONAL
* "-----

*****
*25.04.2012 : Solo se envian clientes Z001
*****

DATA : l_clients  LIKE zcerca_clients .
DATA : l_articles LIKE zcerca_articles OCCURS 0 WITH HEADER LINE.
DATA : BEGIN OF lt_art OCCURS 0,
      matnr LIKE mara-matnr,
END OF lt_art.
DATA : l_where(100).
DATA : l_like(40).
DATA : l_tipus(4).
DATA : l_estat.
data : t_class like sclass occurs 0 with header line. "ins by epr
DATA : t_caract LIKE clobjdat OCCURS 0 WITH HEADER LINE. "ins by epr
data : l_obj like ausp-objek. "ins by epr
data : l_mtart type mtart.

ranges: r_matnr for mara-matnr.
ranges: r_maktg for makt-maktg.
ranges: r_vmsta for mvke-vmsta.
ranges: r_kdmat for knmt-kdmat.
ranges: r_kunnr for knal-kunnr.

refresh r_matnr.
if not i_matnr is initial.
  r_matnr-sign = 'I'.

```

```

    r_matnr-option = 'EQ'.
    r_matnr-low   = i_matnr.
    append r_matnr.
endif.

refresh r_maktg.
if not i_maktg is initial.
    r_maktg-sign   = 'I'.
    r_maktg-option = 'CP'.
    concatenate '*' i_maktg '*' into r_maktg-low.
    append r_maktg.
endif.

refresh r_vmsta.
if not i_vmsta is initial.
    r_vmsta-sign   = 'I'.
    r_vmsta-option = 'EQ'.
    r_vmsta-low   = i_vmsta.
    append r_vmsta.
endif.

refresh r_kdmat.
if not i_kdmat is initial.
    r_kdmat-sign   = 'I'.
    r_kdmat-option = 'EQ'.
    r_kdmat-low   = i_kdmat.
    append r_kdmat.
endif.

refresh r_kunnr.
if not i_kunnr is initial.
    r_kunnr-sign   = 'I'.
    r_kunnr-option = 'EQ'.
    CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
    EXPORTING
        input = i_kunnr
    IMPORTING
        output = r_kunnr-low.
    append r_kunnr.
endif.

l_estat = i_cli_baixa.
IF l_estat NE 'B' .
    CLEAR l_estat.
ENDIF.

CLEAR l_tipus.

IF NOT i_kunnr IS INITIAL OR NOT i_name1 IS INITIAL OR l_estat IS NOT INITIAL
.
    IF NOT i_kunnr IS INITIAL.
        CLEAR l_like.

```

```

CONCATENATE '''%' i_kunnr '%' INTO l_like.
CONCATENATE 'KUNNR LIKE' l_like INTO l_where SEPARATED BY space.
ENDIF.
IF NOT i_name1 IS INITIAL.
  IF NOT l_where IS INITIAL.
    CONCATENATE l_where 'AND' INTO l_where SEPARATED BY space.
  ENDIF.
  CLEAR l_like.
  CONCATENATE '''%' i_name1 '%' INTO l_like.
  CONCATENATE l_where '( NAME1 LIKE ' l_like 'OR' INTO l_where SEPARATED B
Y space.
  CONCATENATE l_where 'NAME2 LIKE ' l_like ')' INTO l_where SEPARATED B
Y space. ENDIF.
IF l_where IS INITIAL.
  l_where = 'KTOKD = ''Z001'''.
ELSE.
  CONCATENATE l_where 'AND KTOKD = ''Z001'' INTO l_where SEPARATED BY spac
e.
ENDIF.

* SELECT DE LA KNA1.
SELECT kunnr name1 ort01 ktokd sperr aufsd lifsd faksd
  cassd
INTO (l_clients-kunnr, l_clients-name1, l_clients-ort01,
  l_clients-ktokd, knal-sperr, knal-aufsd, knal-lifsd,
  knal-faksd, knal-cassd )
FROM knal
WHERE (l_where).
  CLEAR knb1. CLEAR knvv.
  SELECT SINGLE * FROM knb1 WHERE kunnr = l_clients-kunnr.
  SELECT SINGLE * FROM knvv WHERE kunnr = l_clients-kunnr AND
    vkorg = '350' AND
    vtweg = '10' AND
    spart = '10'.

IF knal-sperr IS NOT INITIAL OR
  knal-aufsd IS NOT INITIAL OR
  knal-lifsd IS NOT INITIAL OR
  knal-faksd IS NOT INITIAL OR
  knal-cassd IS NOT INITIAL OR
  knvv-aufsd IS NOT INITIAL OR
  knvv-lifsd IS NOT INITIAL OR
  knvv-faksd IS NOT INITIAL OR
  knvv-cassd IS NOT INITIAL.
  l_clients-estat = 'K'.
ENDIF.
IF knal-loevm = 'X'.
  l_clients-estat = 'B'.
ENDIF.
CALL FUNCTION 'CONVERSION_EXIT_ALPHA_OUTPUT'
  EXPORTING
    input = l_clients-kunnr

```



```

      IMPORTING
        output = l_clients-kunnr.
      APPEND l_clients TO t_clients.
    ENDSELECT.
  ENDIF.

IF l_estat = 'B'.
  DELETE t_clients WHERE estat NE 'B'.
ENDIF.

IF NOT i_kunnr IS INITIAL OR NOT
  i_matnr IS INITIAL OR NOT
  i_maktg IS INITIAL OR NOT
  i_vmsta IS INITIAL OR NOT
  i_kdmat IS INITIAL.
  CLEAR l_where.
  IF NOT i_matnr IS INITIAL.
    l_tipus = 'A'.
    CLEAR l_like.
    CONCATENATE 'i_matnr %' INTO l_like.
    CONCATENATE 'MARA~MATNR LIKE' l_like INTO l_where SEPARATED BY space.
  ENDIF.

  IF NOT i_maktg IS INITIAL.
    l_tipus = 'A'.
    IF NOT l_where IS INITIAL.
      CONCATENATE l_where 'AND' INTO l_where SEPARATED BY space.
    ENDIF.
    CLEAR l_like.
    CONCATENATE 'i_maktg %' INTO l_like.
    CONCATENATE l_where 'MAKTG LIKE' l_like INTO l_where SEPARATED BY space.
  ENDIF.

  IF NOT i_vmsta IS INITIAL.
    IF NOT l_where IS INITIAL.
      CONCATENATE l_where 'AND' INTO l_where SEPARATED BY space.
    ENDIF.
    CLEAR l_like.
    CONCATENATE 'i_vmsta %' INTO l_like.
    CONCATENATE l_where 'VMSTA LIKE' l_like INTO l_where SEPARATED BY space.
    CONCATENATE l_tipus 'B' INTO l_tipus.
  ELSE.
    *En el select borrarémos status 'B'
  ENDIF.

  IF NOT i_kunnr IS INITIAL.
    IF NOT l_where IS INITIAL.
      CONCATENATE l_where 'AND' INTO l_where SEPARATED BY space.
    ENDIF.
    CLEAR l_like.
    CONCATENATE 'i_kunnr %' INTO l_like.
    CONCATENATE l_where 'KNMT~KUNNR LIKE' l_like INTO l_where SEPARATED BY sp

```

```

ace.
    CONCATENATE l_tipus 'C' INTO l_tipus.
ENDIF.
IF NOT i_kdmat IS INITIAL.
    IF NOT l_where IS INITIAL.
        CONCATENATE l_where 'AND' INTO l_where SEPARATED BY space.
    ENDIF.
    CLEAR l_like.
    CONCATENATE ''%' i_kdmat '%'' INTO l_like.
    CONCATENATE l_where 'KDMAT LIKE' l_like INTO l_where SEPARATED BY space.
    CONCATENATE l_tipus 'C' INTO l_tipus.
ENDIF.

REFRESH l_articles.
CLEAR l_articles.

CASE l_tipus.
    WHEN 'A'.

        SELECT mara~matnr maktg
        INTO (l_articles-matnr, l_articles-maktg)
        FROM mara INNER JOIN makt
        ON ( mara~matnr = makt~matnr AND makt~spras = 'S' )
        WHERE (l_where)
            and mara~LVORM <> 'X'.***** SVV 27.11.2013 *****

        SELECT SINGLE vmsta INTO l_articles-vmsta
        FROM mvke WHERE matnr = l_articles-matnr AND
            vkorg = '350' AND
            vtweg = '10'.

        SELECT * FROM knmt WHERE matnr = l_articles-matnr AND
            vkorg = '350' AND
            vtweg = '10'.

            l_articles-kunnr = knmt-kunnr.
            l_articles-kdmat = knmt-kdmat.
        APPEND l_articles .
    ENDSELECT.
    IF sy-subrc NE 0.
        APPEND l_articles.
    ENDIF.
    CLEAR l_articles.
ENDSELECT.

WHEN OTHERS.

    IF l_tipus CS 'B' AND l_tipus CS 'C'.

        SELECT mara~matnr maktg mvke~vmsta
        INTO (l_articles-matnr, l_articles-maktg, l_articles-vmsta)

```

```

FROM mara INNER JOIN makt
ON ( mara~matnr = makt~matnr AND makt~spras = 'S' )
INNER JOIN mvke ON ( mvke~matnr = mara~matnr AND
                    mvke~vkorg = '350'      AND
                    mvke~vtweg = '10'      )
WHERE mara~matnr in r_matnr AND " = i_matnr AND
      mvke~vmsta in r_vmsta      "= i_vmsta.
      and mara~LVORM <> 'X'. "***** SVV 27.11.2013 *****

SELECT SINGLE kunnr kdmat INTO (l_articles-kunnr, l_articles-kdmat)
FROM knmt WHERE matnr = l_articles-matnr AND
              kunnr = i_kunnr AND
              vkorg = '350' AND
              vtweg = '10' AND
              kdmat in r_kdmat.
if not r_kdmat[] is initial.
  check sy-subrc eq 0.
endif.

APPEND l_articles.
CLEAR l_articles.
ENDSELECT.
*fi ins by epr.
ELSEIF l_tipus CS 'B'.
CALL FUNCTION 'CONVERSION_EXIT_ALPHA_INPUT'
EXPORTING
  input  = i_kunnr
IMPORTING
  output = i_kunnr.
SELECT mara~matnr maktg vmsta
INTO (l_articles-matnr, l_articles-maktg, l_articles-vmsta)
FROM mara INNER JOIN makt
ON ( mara~matnr = makt~matnr AND makt~spras = 'S' )
INNER JOIN mvke ON ( mvke~matnr = mara~matnr AND
                    mvke~vkorg = '350'      AND
                    mvke~vtweg = '10'      )
WHERE (l_where)
      and mara~LVORM <> 'X'. "***** SVV 27.11.2013 *****.
      SELECT * FROM knmt WHERE matnr = l_articles-matnr AND
                    vkorg = '350'      AND
                    vtweg = '10'.

      l_articles-kunnr = knmt-kunnr.
      l_articles-kdmat = knmt-kdmat.
      APPEND l_articles .
ENDSELECT.
IF sy-subrc NE 0.
  APPEND l_articles.
ENDIF.
CLEAR l_articles.
ENDSELECT.
ELSEIF l_tipus CS 'C'.

```

```

SELECT mara~matnr maktg
INTO (l_articles~matnr, l_articles~maktg)
FROM mara INNER JOIN makt
ON ( mara~matnr = makt~matnr AND makt~spras = 'S' )
WHERE mara~matnr in r_matnr and "= i_matnr.
      maktg      in r_maktg
and mara~LVORM <> 'X'.***** SVV 27.11.2013 *****.

SELECT SINGLE kunnr kdmat INTO (l_articles~kunnr, l_articles~kdmat)
FROM knmt
WHERE matnr = l_articles~matnr AND
      kunnr in r_kunnr and

      vkorg = '350' AND
      vtweg = '10' AND
      kdmat in r_kdmat.

if not r_kdmat[] is initial or not r_kunnr[] is initial.
  check sy-subrc eq 0.
endif.

SELECT SINGLE vmsta INTO l_articles~vmsta
FROM mvke WHERE matnr = l_articles~matnr AND
      vkorg = '350' AND
      vtweg = '10' AND
      vmsta in r_vmsta.

if not r_vmsta[] is initial.
  check sy-subrc eq 0.
endif.

APPEND l_articles.
CLEAR l_articles.
ENDSELECT.

*fi ins by epr
ENDIF.

ENDCASE.

IF i_vmsta IS INITIAL.
  DELETE l_articles WHERE vmsta = 'B'.
ENDIF.

LOOP AT l_articles.
*07.02.2013 filtrem per ZACA i ZMER
  select single mstart into l_mstart
  from mara where matnr = l_articles~matnr.
  check l_mstart = 'ZACA' or l_mstart = 'ZMER'.
*

CALL FUNCTION 'CONVERSION_EXIT_ALPHA_OUTPUT'
EXPORTING
  input = l_articles~kunnr

```

```

IMPORTING
  output = l_articles-kunnr.

CALL FUNCTION 'CONVERSION_EXIT_MATN1_OUTPUT'
  EXPORTING
    input = l_articles-matnr
  IMPORTING
    output = l_articles-matnr.
*
  ins by epr
    IF l_articles-kunnr IS INITIAL.
      clear l_obj.
      l_obj = l_articles-matnr.
      CALL FUNCTION 'CLAF_CLASSIFICATION_OF_OBJECTS'
        EXPORTING
          class           = 'CLAS_MAT_PA'           "producto acabado
          classtext       = 'X'
          classtype       = '001'
          features        = 'X'
          language        = sy-langu
          object          = l_obj
          objecttable     = 'MARA'
          key_date        = sy-datum
          initial_charact = ' '
          change_service_clf = 'X'
        TABLES
          t_class         = t_class
          t_objectdata    = t_caract
        EXCEPTIONS
          no_classification = 1
          no_classtypes    = 2
          invalid_class_type = 3
          OTHERS           = 4.

      READ TABLE t_caract WITH KEY atnam = 'ZCODCLIEN'.
      IF sy-subrc EQ 0.
        l_articles-kunnr = t_caract-auspl.
      ENDIF.
    ENDIF.
*
  fi ins by epr

  MOVE-CORRESPONDING l_articles TO t_articles.
  APPEND t_articles.
ENDLOOP.
ENDIF.

ENDFUNCTION.

```

8. Bibliografia

En aquest TFG no hem hagut de consultar bibliografia donat que tota la documentació la vam extreure principalment dels BBP realitzats amb l'integrador, i de la pròpia experiència en la participació del projecte, per tant, encara que sembli estrany, no hem d'incloure referències bibliogràfiques a documentació externa. A part, tal i com està escrit al principi del document, no podem oferir referències a informació/documentació interna, per la naturalesa particular i privada de la solució, i per exprés motiu de l'empresa.

