

Dgp21, entorn de desenvolupament i execució d'aplicacions.

Julio Castro Digón (jcastro@dims.com)
Conxita Marín Font (cmarin@dims.com)

Juny 2007

Resum.

Dgp21 es un entorn d'execució d'aplicacions client-servidor en 3 capes: Servidor d'Aplicacions, Motor de Base de Dades i Aplicació client que incorpora el seu propi entorn de desenvolupament d'aplicacions.

És un sistema orientat al desenvolupament ràpid d'aplicacions ERP. Considerem que és una molt bona alternativa a les aplicacions tipus web, aportant els següents avantatges: estalvi d'ample de banda de comunicacions molt significatiu, manteniment més fàcil i major interactivitat de l'aplicació client amb l'usuari. Tots els components del sistema han estat escrits en llenguatge C i C++. El sistema i les aplicacions desenvolupades tenen llicència lliure.

1. Introducció.

La nostra experiència com a desenvolupadors d'aplicacions des de l'any 1989 ens ha fet constatar les següents vulnerabilitats:

- Obsolescència ràpida de la interfície gràfica que veu l'usuari.
- Dependència de les eines de desenvolupament:
 - Inversions continuades en actualitzacions de llicències i reprogramació de codi. Un exemple el tenim amb el Visual Basic, cada nova versió ha comportat reprogramacions més o menys importants de les aplicacions. Només cal fer una cerca a Google i veurem que hi ha llibres sencers dedicats a ajudar a realitzar aquestes migracions “sense problemes”.
 - Perill que la discontinuïtat d'una eina faci fracassar la inversió en desenvolupament realitzada. Qui s'enrecorda del RAD d'Oracle, Power Objects? [7].

D'altra banda varem tenir en compte els següents factors que l'entorn ens posava a l'abast:

- Abaratiment del cost de les comunicacions per internet i possibilitat de connexió a qualsevol lloc i per tant també accés a les aplicacions.
- Maduresa dels sistemes operatius Linux, bases de dades relacionals potents com Postgresql [1] i altres eines lliures que ens alliberaven del cost de les llicències a nosaltres com a desenvolupadors i també la seva repercussió en el cost de la solució final ofertada.

Tenint en compte tots aquests factors varem decidir com havien de ser les aplicacions a desenvolupar en el futur per tal que ens permetessin superar les vulnerabilitats exposades i aprofitar tots els avantatges tecnològics que l'entorn posava al nostre abast.

Així va néixer el projecte de servidor d'aplicacions i entorn de desenvolupament Dgp21.

Aquest article presenta els objectius assolits amb el projecte, els components del sistema, la funcionalitat de les aplicacions desenvolupades, l'estat actual del projecte i les idees que tenim per al futur.

2. Objectius assolits pel projecte.

- Independència de la interfície visual i de la lògica de l'aplicació.
- Independència del servidor de base de dades, sempre que segueixi el model relacional estàndard SQL
- Escalable en aprofitament de recursos de servidor i d'ample de banda. El sistema pot incorporar múltiples servidors d'aplicacions i connexions a internet.
- Mínim consum d'ample de banda, aplicacions utilitzables a qualsevol lloc i amb qualsevol tipus de connexió (ADSL, modem, 3G...)
- Entorn segur, xifrat de les comunicacions amb clau incorporat.
- Desenvolupament ràpid.
- Fàcil manteniment
- No utilització de components amb llicències propietàries.
- Orientat al desenvolupament d'aplicacions de gestió per a PIMES (ERPs)

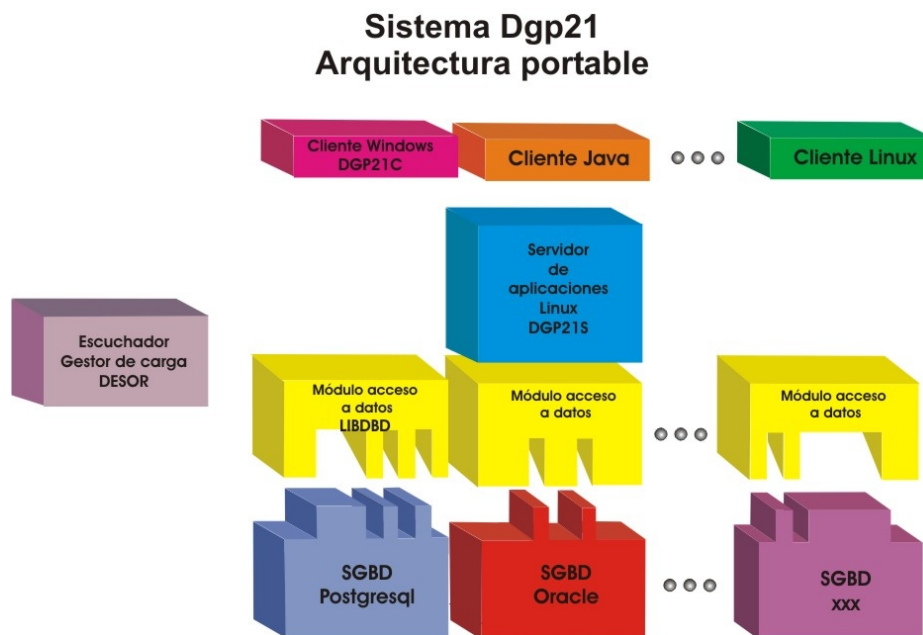


Figura1: Arquitectura del sistema Dgp21

3. Components del sistema.

3.1. Escoltador (DESOR).

És la porta d'entrada al sistema. És l'encarregat d'atendre les peticions de connexió dels usuaris. Gestiona una taula de càrrega configurable dels diferents servidors operatius del sistema Dgp21 i de les diferents connexions a internet que té assignades. Quan rep una petició de connexió d'un usuari la redirigeix al servidor i a la línia de connexió amb menys càrrega del sistema.

A continuació es mostra un exemple de la taula de treball de càrrega que gestiona l'escoltador.

001dims1	gestio	1192.168.116.4	003748008000021
002dims1	gestio	r80.80.161.68	003748001500001
003dims1	program	1192.168.216.4	003748008000001
004dims1	program	1192.168.216.4	003748008000011
005dims5	demo	1192.168.216.5	013748008000021
006dims5	demo	r61.149.16.182	013748002000051

Aquesta taula conté informació dels servidors que atenen al sistema.
Per exemple, la línia 1 informa del següent:

001:	Número d'ordre.
dims1:	Nom de servidor.
gestio:	Grup d'usuaris a qui atendra.
l:	Admet connexions locals.
192.168.116.4:	ip de connexió
3748:	Port de connexió
80:	Número d'usuaris màxim
2:	Número d'usuaris connectats
1:	Servidor actiu

Funcionament.

La taula de càrrega de treball (dgpsap) la crea l'escoltador a l'arrencar a partir d'un fitxer de configuració en text pla (dgpsap.cfg). Quan arrenca un servidor d'aplicacions s'activa el seu estat a la taula de càrrega a fi i efecte de que l'escoltador el tingui en compte per gestionar el balanceig. Si el servidor d'aplicacions no es troba a la mateixa màquina que l'escoltador, l'activació es realitza amb un missatge remot a l'escoltador.

Quan una aplicació client demana una connexió a l'escoltador, aquest busca la millor opció en funció del grup d'usuaris al qual pertany, del tipus de connexió: local o remota i segons la càrrega relativa de cada servidor, retornant a l'aplicació client la nova ip i el nou port de connexió.

Les figures 2 i 3 il.lustren els conceptes exposats.

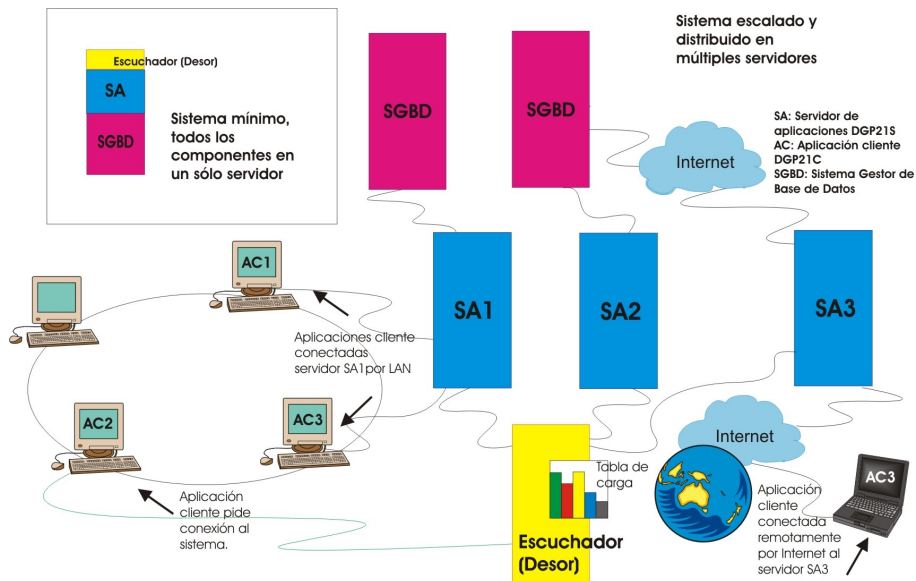


Figura 2: Escalabilidad i sistema de balanceig de càrrega entre servidors

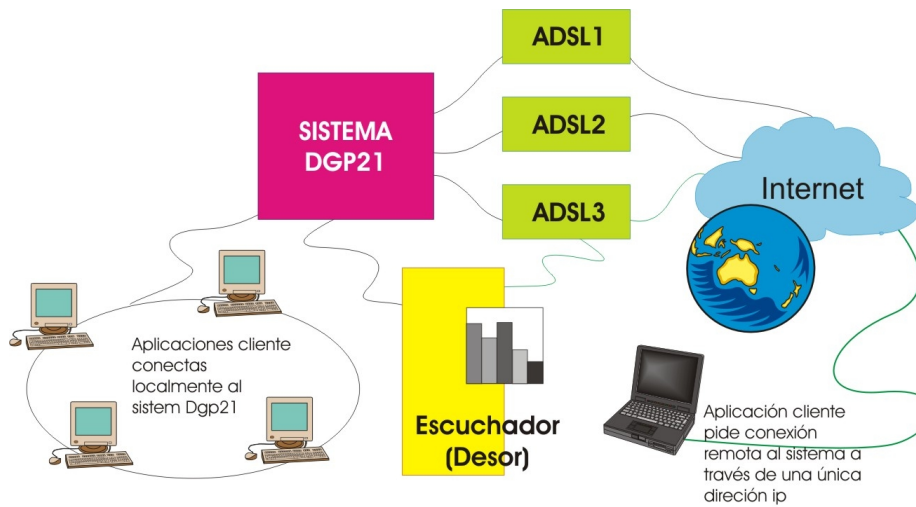


Figura 3: Escalabilidad en línies de comunicacions i balanceig de càrrega entre elles

3.2. Servidor d'aplicacions Dgp21.

És l'encarregat d'executar les aplicacions emmagatzemades a la base de dades de programació, interaccionar amb la base de dades de l'usuari i atendre el diàleg amb l'aplicació client. Ha estat desenvolupat partint de zero en llenguatge C.

3.2.1. Codi de les aplicacions desat en base de dades.

El servidor d'aplicacions treballa amb dues bases de dades diferenciades:

- Base de dades del usuari.
- Base de dades de programació.

Base de dades de l'usuari.

Com el seu nom indica és la que gestiona les dades introduïdes pels usuaris amb l'aplicació. També conté informació que fa servir el servidor d'aplicacions per a personalitzar els aplicatius a diferents nivells: Per sector (alimentació, construcció, tèxtil, transport, etc), per empresa o base de dades, per perfil d'usuari i per usuari.

Base de dades de programació.

A la base de dades de programació es guarden les definicions dels formularis de l'aplicació i el codi que ha d'executar el servidor d'aplicacions.

Un formulari es compon d'un conjunt de controls. El codi que executa el servidor d'aplicacions sempre està associat a un control d'un formulari. Aquest codi s'executarà quan el formulari sigui mostrat per l'aplicació client i l'usuari cliqui el control, si és un control clicable, o sinó, quan el control perdi el focus. El codi a executar pel servidor es compon d'una o més funcions que s'executaran de forma seqüencial o bé seguint el fluxe especificat en llenguatge LCF (Llenguatge de control de funcions, veure apartat 3.2.6).

3.2.2. Mòdul d'accés a la base de dades.

Per a fer l'accés a la base de dades del servidor d'aplicacions Dgp21 s'ha fet un mòdul diferenciat. Aquest mòdul és l'encarregat de fer l'enllaç amb la interfície de programació proporcionada pel motor de base de dades. Això s'ha fet així per facilitar la portabilitat del servidor d'aplicacions a diferents sistemes gestors de bases de dades (veure figura 1). Per fer que les aplicacions desenvolupades en el sistema Dgp21 funcionin amb un altre gestor de base de dades només cal desenvolupar el mòdul d'accés específic. Tot el codi de les

aplicacions desenvolupades es pot aprofitar. El mòdul d'accés a dades que hem desenvolupat actualment és per a Postgresql [1]

3.2.3. IDE integrat.

La mateixa aplicació serveix per a desenvolupar-se ella mateixa. Per a fer-ho només cal connectar-se a la base de dades de programació i accedir al formulari que permet la creació de nous formularis i la programació de codi.

3.2.4. Exemple de desenvolupament d'aplicacions.

La figura 4 ens mostra un exemple de formulari creat amb l'entorn de desenvolupament. Concretament és el formulari que permet mantenir la taula de lliterals del sistema tal i com es mostra a l'usuari de l'aplicació. Veiem que l'usuari ha fet una selecció de lliterals i es disposa a modificar un d'ells.

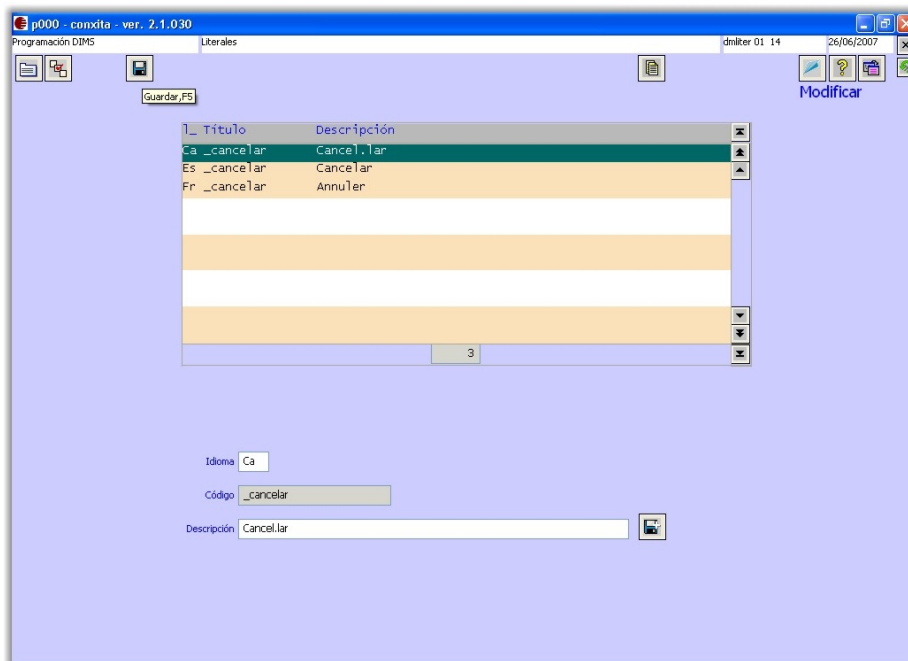


Figura 4: Exemple de formulari

p000 - conxita - ver. 2.1.030
 Programación DIMS Gestor de pantallas de DGP21 dgmpan02.12 26/06/2007

Programa dmliter 01 14 Num. control 11 Tabulador 0 Pestaña 0

Etiqueta _guardar Tabla destino Pos. etiq M Núm. dígitos 0

Posic 1 101 17 Referencia 8 1 Campo destino Tipo H Pertinencia B

Posic 2 5 11 Enlazar ctrl 0 Observaciones Contenido S Atrib. altas 00 00

Programación //codigo guardar
 modo char (10) @50;
 if (modo == "_altas")
 exef(1); // insert
 else
 exef(2); // update
 exef(3); // refresca

Signo 0 Atrib. modif 00 00

Editado 0 Atrib. consulta 20 10

Passar ctrl 1 Atrib. buscar 20 10

Valor def.

Orden Función Fichero func. Tipo Parám. func. Enlaces

1	q_ejecuta		B	1,r,0	
---	-----------	--	---	-------	--

Sentencia sql
 insert into dgliter values ('#051','#052','#053')

Pares de campos o descripción listado

Observaciones

X = 709, Y = 104

Orden Función Fichero func. Tipo Parám. func. Enlaces

2	q_ejecuta		B	1,r,0	
---	-----------	--	---	-------	--

Sentencia sql
 update dgliter set l_idioma='#051',l_codi='#052',l_desc='#053' where aid=#024

Pares de campos o descripción listado

Observaciones

X = 749, Y = 011

Orden Función Fichero func. Tipo Parám. func. Enlaces

3	l_refresca		B	10,1	
---	------------	--	---	------	--

Sentencia sql

Pares de campos o descripción listado

Observaciones

X = 761, Y = 263

Figura 6: Formulari de definició de controls II

Figura 7: Formulari de definició de controls III

La figura 5 ens mostra la pantalla de creació i manteniment de formularis. Veiem la definició del control “Desar” de la pantalla de manteniment de la figura 4. A la part superior veiem les dades que defineixen l'aparença del control i el codi de programació en llenguatge LCF (Llenguatge de Control de Funcions, veure apartat 3.2.6.). A la part baixa observem les funcions associades al control. En aquest cas n'hi ha 3, a la figura 5 veiem la primera i a les figures 6 i 7 la segona i tercera funció respectivament. Si no hi hagués res a la casella “Programación”, al prémer el botó “Desar” s'executarien seqüencialment totes les funcions associades. En aquest cas hi ha codi de programació i s'executen segons el fluxe especificat en el mateix.

Codi de programació del botó “Desar”:

```
1. //codigo guardar
2. modo char(10)@50;
3. if (modo == "_altas")
4.     exef(1); // insert
5. else
6.     exef(2); // update
7. exef(3); // refresca
```

Explicació del codi:

1. Comentari.
2. Associem a la variable “modo” de tipus caràcter, de 10 dígits de capacitat, el contingut del control 50 que en aquest cas és el control que conté la informació del mode en que està l'aplicació: altes, baixes, modificacions, etc.
3. Si s'acompleix la condició llavors
4. Executa la funció número 1. (insert)
5. en qualsevol altre cas
6. Executa la funció número 2. (update)
7. Executa la funció 3. (refresca les dades del formulari)

3.2.5. Funcions.

Les funcions són les que duen a terme les accions concretes al servidor. Hi ha funcions que permeten modelar el comportament de l'aplicació i d'altres que interactuen amb la base de dades. Les funcions tenen accés al contingut dels controls dels formularis i l'utilitzen per a construir sentències SQL o crides a procediments emmagatzemats. També poden actualitzar els valors dels

controls amb el resultat d'una sentència SQL.

Exemple d'una funció.

A la figura 5 veiem la funció “g_ejecuta” amb la següent sentència SQL:

```
insert into dgliter values ('#051', '#052', '#053')
```

Quan el servidor d'aplicacions executa aquesta funció, pren la sentència SQL especificada, substitueix cada cadena del tipus #nnn pel valor contingut al control amb referència nnn i executa la sentència resultant a la sessió de la base de dades que correspon a l'aplicació client.

3.2.6. Llenguatge de control de funcions (LCF).

Aquest llenguatge de programació, com ja s'ha avançat a l'apartat 3.2.4, permet manipular, operar i interrogar els valors dels controls dels formularis i controlar el fluxe d'execució de les funcions associades a un control. Té una notació semblant al llenguatge C i disposa d'instruccions de control: if-else i while, assignacions, operadors lògics, operadors matemàtics i altres instruccions pròpies del llenguatge. El codi font és compilat en el moment de ser desat a la base de dades. Aquest codi compilat és executat per una màquina virtual “libmvd” creada desde zero pels autors del projecte.

Per a crear el llenguatge i el compilador s'ha fet a partir de la base de l'excel.lent tutorial sobre Lex i Yacc de Tom Newman. [2]. També ha estat de molta utilitat per aclarir conceptes, el quadern pràctic sobre compiladors de Juan Manuel Cueva Lobelle de la Universitat d'Oviedo[3]

3.2.7. Tipus de controls.

El sistema està pensat per a desenvolupar aplicacions de gestió empresarial i els controls que s'han implementat fins ara han estat els que hem anat necessitant per a construir aquest tipus d'aplicació. Els enumerem a continuació:

- Caixa de text alfanumèric
- Caixa de text numèric amb format
- Caixa de text tipus data
- Caixa de text extens amb scroll vertical
- Botó

- Botó tipus heina
- Botó d'ajuda
- Recuadre
- Finestra sobreposada
- Control tipus “menú”
- Pestanya
- Checkbox
- Text interactiu per a graelles de text

3.2.8. Protocol de comunicacions entre client i servidor.

El protocol de comunicacions entre l'aplicació servidora i l'aplicació client té bàsicament la següent estructura:

- Formulari:
 - Referencia
 - Idioma
 - Versió
- Vector amb les dades que contenen els controls del formulari
- Control que ha originat el retorn al servidor

El paquet de dades que viatja de client a servidor i viceversa , és xifrat en origen i desxifrat en destí.

Visió general funcionament.

Quan l'usuari obre l'aplicació client en el seu ordinador, el primer que fa és mostrar el formulari de connexió on introduirà les següents dades.

- Identificació de l'usuari
- Paraula de pas
- Base de dades d'usuari
- Idioma .

Aquesta informació, convenientment xifrada amb una clau, arribarà a l'escoltador que la desxifrarà, i d'acord amb el mecanisme de balanceig de càrrega entre servidors i entre línies de comunicació explicat a l'apartat 3.1., indicarà a l'aplicació client a quin servidor d'aplicacions ha de connectar-se i a través de quina ip. L'aplicació client iniciarà la nova connexió, el servidor obrirà una sessió a la base de dades per a aquest client i s'iniciarà el diàleg

entre servidor d'aplicacions i aplicació client.

3.2.9. Diàleg entre aplicació servidora i aplicació client.

A continuació mostrem el diàleg típic entre aplicació servidora i aplicació client a partir de l'establiment de la connexió.

A.S.: Aplicació servidora.

A.C.: Aplicació client.

A.S: Envia un paquet de dades al client segons el protocol de comunicacions esmentat..

A.C.: Rep el paquet de dades. Llegeix la definició del formulari en el fitxer de pantalles local i procedeix a construir la pantalla que veurà l'usuari. Si l'aplicació client no té guardat el formulari, bé perquè és nou o bé té una versió obsoleta, sol·licita al servidor l'enviament de la definició del formulari que actualitza en el seu fitxer de pantalles i després procedeix a construir la pantalla i omplir els controls amb les dades subministrades pel servidor en el vector de dades.

En aquest moment el control de l'aplicació està a la part client: l'usuari introdueix dades en els diferents controls del formulari. Quan l'usuari clica un botó o surt d'un control que té la propietat de “retornar control al servidor”, l'aplicació client prepara el paquet per enviar al servidor amb les dades dels controls del formulari així com la informació sobre quin control ha estat el que ha originat la transferència al servidor.

A.S.: Rep el paquet del client, carrega les dades dels diferents controls del formulari a la seva àrea de memòria, executa el codi associat al control que ha originat la crida i respon al client indicant altre cop un formulari que podrà ser el mateix o un altre en funció del codi executat, juntament amb el vector de dades actualitzat.

Aquest diàleg s'acabarà quan l'usuari tanqui l'aplicació client. El servidor per la seva part tancarà la sessió de la base de dades, recalculerà la càrrega i actualitzarà el fitxer dgpsap (veure apartat 3.1.).

3.3. Aplicació client.

Té com a missió dibuixar la interfície que veu l'usuari, interactuar amb aquest i enviar i rebre la informació al servidor. L'aplicació client desenvolupada fins

ara és per a Windows. També és pot executar en Linux fent servir Wine[8]. La aplicació client no conté cap lògica de negoci, d'aquesta forma podrem desenvolupar una altra aplicació client que corri en un altra plataforma sense comprometre les aplicacions desenvolupades (veure figura 1). Està escrita en llenguatge C++ . Pel seu desenvolupament s'ha partit d'un model de programació d'aplicacions Windows creat per Bartosz Milewski [4]. Això ens ha permès utilitzar un model orientat a objectes sense la necessitat d'utilitzar llibreries comercials com les MFC de Microsoft [9]. La aplicació client no necessita cap procés d'instal·lació , fins i tot es pot executar des d'un dispositiu de memòria USB connectat a qualsevol ordinador amb connexió a Internet. L'aplicació client crea els formularis a la pantalla de l'usuari invocant directament les API's natives de Windows seguint les instruccions enviades pel servidor. Existeix un mecanisme caché que guarda la informació necessària per a crear les pantalles en local, de forma que l'aplicació client recorda com ha de pintar una pantalla i per tant aquesta informació no ha de ser transferida del servidor a menys que hi hagi hagut canvis. Aquest mecanisme redueix considerablement el volum de dades que viatja per la xarxa. L'aplicació client s'encarrega de xifrar les dades enviades al servidor i desxifrar les rebudes abans de mostrar-les a l'usuari.

4. Funcionalitat de les aplicacions Dgp21.

4.1 Suport multidioma.

Una aplicació Dgp21 pot treballar simultàniament amb diferents idiomes. Cada usuari indica amb quin idioma vol treballar a la pantalla de connexió al sistema. El servidor d'aplicacions s'encarregarà de mostrar els formularis, les pàgines d'ajuda, els “tooltips” dels controls i els missatges de l'aplicació en l'idioma que ha seleccionat. El sistema proporciona les eines necessàries a nivell d'usuari per a traduir els textos de l'aplicació.

4.2 Mecanismes de personalització.

L'entorn proporciona mecanismes de personalització de les aplicacions que no es perden amb les actualitzacions. Aquests mecanismes actuen per nivells de preferència de més particular a més general. Els nivells de personalització són:

- Usuari
- Perfil o grup d'usuaris
- Sector
- Empresa

Aquests mecanismes permeten:

- Activar comportaments diferents de l'aplicació.
- Establir valors per defecte.
- Restringir l'accés a menús de l'aplicació
- Aplicar restriccions fins al nivell de control de formulari, permetent deshabilitar o ocultar camps o controls.
- Aplicar tot l'anterior en funció del tipus d'usuari i d'accés a l'aplicació: local o remota.

4.3 Ajuda.

L'ajuda de l'aplicació la invoca l'usuari prement el botó “Ajuda” present als formularis de l'aplicació. L'ajuda està en format html i es pot configurar perquè accedeixi a les pàgines de forma local, servidor de xarxa o internet. L'aplicació client s'encarrega de resoldre la pàgina a mostrar en funció de l'idioma de connexió.

4.4 Missatgeria instantània.

L'entorn proporciona un sistema de missatgeria instantània i diferida al servei dels usuaris de l'aplicació i que ofereix la funcionalitat de programació d'alarmes a les aplicacions desenvolupades.

4.5 Gestió documental.

L'entorn proveeix una sèrie de funcions que permeten a les aplicacions desenvolupades treballar amb documents en qualsevol format. Aquests documents estan emmagatzemats en el servidor de forma xifrada en una carpeta d'accés restringit als usuaris.

Aquests documents es poden vincular a les aplicacions desenvolupades. Quan un document és requerit per un usuari a través de l'aplicació, el servidor l'envia a l'aplicació client que se n'encarrega de desxifrar-lo i emmagatzamar-lo localment perquè l'usuari treballi amb ell. Un cop finalitzades les

modificacions, el document torna a ser xifrat i enviat al servidor.

4.6 Generador d'impresos.

Per a generar impresos sofisticats tipus albarans, factures, etc, s'ha utilitzat el paquet jfreereports [5]. Aquest generador d'informes treballa amb una definició d'informe guardat en format xml. Aquests formats es guarden en el servidor com a un document més de la gestió documental.

S'ha creat un mòdul a l'aplicació client encarregat de fer l'enllaç amb aquest paquet i generar els informes.

5. Estat actual del projecte.

En aquests moments tota la funcionalitat del sistema descrit està implementada. S'han desenvolupat i estan en funcionament les següents aplicacions:

- Control de presència, accessos i gestió de sirenes
- CRM
- Gestió Documental
- Gestió d'un SAT de tallers de vehicles
- Gestió d'empreses de transport i duanes
- Un ERP amb els següents mòduls:
 - Gestió de compres
 - Gestió de stocs.
 - Gestió de vendes.
 - Control de tresoreria.
 - Comptabilitat general. (en desenvolupament)
 - Comptabilitat de costos. (en desenvolupament)
 - Gestió de riscos comercials.
 - Gestió de la fabricació. (en desenvolupament)
 - Control de producció.
 - Traçabilitat (en desenvolupament)
 - Missatgeria instantània
 - Enviament de SMS (en desenvolupament)

Tot el sistema desenvolupat així com les aplicacions esmentades están sota llicència lliure i es poden descarregar des de aquests links a sourceforge:

<http://sourceforge.net/projects/dgp21/>
<http://sourceforge.net/projects/i-globalgest/>

6. Futur del projecte.

Aquí deixem anar algunes idees que portem al cap per al futur del nostre projecte.

- Acabar l'ERP i fer més gran la família d'aplicacions disponibles.
- Dotar-nos d'un sistema gràfic de definició de formularis que agilitzi la feina de creació i retoc de la interfície gràfica.
- Enriquir la interfície gràfica actual afegint nous tipus de controls.
- Crear un client multiplataforma (Windows/Linux) possiblement amb wxWidgets [6].

7. Conclusions.

Amb la creació de l'entorn de desenvolupament i execució d'aplicacions Dgp21 hem assolit els objectius que ens havíem proposat i que resumim en els següents punts:

- Desenvolupar aplicacions segons el model client-servidor en 3 capes, capaces de ser utilitzades a qualsevol lloc a través d'internet.
- Dotar-nos d'eines de desenvolupament i manteniment ràpid d'aplicacions pròpies i independents de qualsevol fabricant que assegurí el futur de les aplicacions creades.
- I tot, utilitzant sempre com a base el software lliure, la qual cosa ens aporta dos avantatges competitius fonamentals a l'eliminar el cost de llicències:
 - Reducció del cost de desenvolupament.
 - Reducció del cost de la solució final ofertada al client.

Referències.

- [1] Sistema Gestor de Base de Dades Postgresql: <http://www.postgresql.org/>
- [2] “A Compact Guide to Lex & Yacc”, Tomas Niemann, <http://epaperpress.com/lexandyacc/download/lexyacc.pdf>
- [3] Cuaderno didáctico nº 10, Conceptos básicos de procesadores de lenguaje, Juan Manuel Cueva Lovelle, Universidad de Oviedo http://www.di.uniovi.es/procesadores/Apuntes/ConceptosBasicos/10_Conceptos_Basicos_Procesadores_Lenguaje.pdf
- [4] Bartosz Milewski., “C++ *In Action. Industria-Strength Programming Techniques*”. Addison-Wesley
- [5] JfreeReport project, <http://sourceforge.net/projects/jfreereport/>
- [6] wxWidgets, Cross-Platform GUI Library, <http://www.wxwidgets.org/>
- [7] Oracle Power Objects , RAD d'Oracle similar a Visual Basic Basic que va sortir al mercat cap a l'any 1996 i pocs anys més tard es va abandonar el seu desenvolupament.
- [8] Wine, emulador d'aplicacions Windows per a Linux: <http://www.winehq.org/>
- [9] MFC, Microsoft Foundation classes. http://en.wikipedia.org/wiki/Microsoft_Foundation_Classes