

**Treball Final de Carrera**

*Gestió de pressupostos per a la construcció*

Ignasi Àlvarez Sanuy

**Enginyeria Tècnica d'Informàtica de Gestió i  
Enginyeria Tècnica d'Informàtica de Sistemes**

Directora: M. Dolors Anton Solà  
Vic, setembre de 2007

## Índex

0. Resums .....	3
0.1. Resum en Català .....	3
0.2. English Summary .....	4
1. Proposta i estudi del problema .....	5
2. Anàlisi del sistema .....	7
2.1. Model de dades .....	7
2.2. Model funcional .....	10
3. Disseny del sistema .....	34
3.1. Disseny d'interfícies .....	34
3.2. Disseny de fitxers .....	40
3.3. Disseny de programes .....	46
3.4. Disseny tecnològic .....	66
4. Millores i conclusions .....	72
4.1. Millores .....	72
4.2. Conclusions .....	73
5. Bibliografia .....	74

## 0. Resums

### 0.1. Resum en Català

**Eng. Tècnica d'informàtica de Gestió / Eng. Tècnica d'informàtica de Sistemes**

**Títol:** GESTIÓ DE PRESSUPOSTOS PER A LA CONSTRUCCIÓ

**Paraules clau:** Gestió, pressupostos, construcció, FDS, capítols, partides, ITEC.

**Autor:** Ignasi Àlvarez Sanuy

**Direcció:** M.Dolors Anton Solà

**Data:** Setembre de 2007

#### Resum

En els últims anys el sector de la construcció ha experimentat un creixement exponencial. Aquest creixement ha repercutit sobre molts aspectes: des de la necessitat de tenir més personal a les obres, la implantació d'unes oficines per a poder gestionar la comptabilitat i portar un control sobre les obres fins a la necessitat d'haver de disposar de programes informàtics específics que ajudin a realitzar la feina de la manera més còmode i àgil possible.

El projecte que s'ha dut a terme consisteix a cobrir una d'aquestes necessitats, que és la de la gestió dels pressupostos en les diferents obres que els constructors realitzen. Utilitza la base de dades de l'ITEC (Institut de Tecnologia de la Construcció de Catalunya) sobre la qual treballen la immensa majoria dels arquitectes quan dissenyen les obres, però també permet entrar les pròpies dades que el constructor vulgui. L'usuari de l'aplicació podrà fer pressupostos per obres de nova construcció, reformes... agrupant cada una d'elles per capítols. Aquests capítols els podem entendre com les diferents fases a dur a terme, per exemple: la construcció dels fonaments, l'aixecament de les parets o fer la teulada. Dins dels capítols hi trobem les partides, que és un conjunt de materials i hores de feina i maquinària per a dur a terme una part de l'obra, com per exemple seria fer un envà de separació entre habitacions. En aquest cas hi tindriem els diferents materials que necessitaríem, totxanes, morter; les hores de manobre necessàries per aixecar-la, el transport de tot el material fins a l'obra... Tots aquests paràmetres (materials, hores, transport...) s'anomenen articles i van inclosos a dins de les partides.

Aquesta aplicació està dissenyada per funcionar en un entorn client/servidor, utilitzant com a servidor un Linux OpenSuse 10.2 i com a clients estacions de treball amb Windows XP, tot i que també podríem utilitzar d'altres versions dels sistemes operatius de Microsoft.

L'entorn de desenvolupament utilitzat és el del llenguatge FDS, el qual ja porta integrat un gestor de fitxers que és el que es farà servir.

## 0.2. English Summary

**Eng. Tècnica d'informàtica de Gestió / Eng. Tècnica d'informàtica de Sistemes**

**Title:** GESTIÓ DE PRESSUPOSTOS PER A LA CONSTRUCCIÓ

**Key words:** Gestió, pressupostos, construcció, FDS, capítols, partides, ITEC.

**Author:** Ignasi Àlvarez Sanuy

**Director:** Dolors Anton Solà

**Date:** September 2007

### Summary

In the last years the sector of the construction has experienced an exponential growth. This growth has repelled on many aspects: from the necessity to have more personnel to works, the implantation of offices to be able to manage the accounting and to take a control on works to the needed one to have to have specific computer science programs that they help to make the work of possible the most comfortable and agile way.

The project that has been carried out consists of covering one with these necessities, that she is the one of the management of the budgets in the different works that the constructors make. It uses the data base of the ITEC (Institut de Tecnologia de la Construcció de Catalunya) on which work the immense majority of the architects when they design works, but also allows to enter the own data that the constructor wants. The user of the application will be able to make budgets by works of new constructions, reforms... grouping each one of them by chapters. These chapters we can understand them like the different stages to carry out, for example: the construction of the foundations, the rise of the walls or to make the tile roof. Within the chapters we found the games, that it is a set of materials, working hours and machinery to carry out a part of the work, as for example would be to make a partition of separation between rooms. In this case we would have the different materials that we would need, bricks, mortar; the hours of trowel necessary to raise it, the transport of all the material until the work... All these parameters (material, hours, transport) are denominate articles and go including within the games.

This application is designed to work in surroundings client/server, using like servant a Linux OpenSuse 10.2 and clients workstations with Windows XP, although also we could use other versions of the operating systems of Microsoft.

The surroundings of used development is the FDS language, which already takes integrated a file manager who is what it will be used.

## 1. Proposta i estudi del problema

El sistema ha de permetre la realització i el manteniment de pressupostos valorats per a la construcció. Ha de ser capaç també de realitzar el manteniment dels diferents articles, grups d'articles, mesures... Val a dir que a partir d'aquí es podran treure llistats dels diferents elements que en formen part i els impresos corresponents pel propi constructor i pel client final.

L'estructura dels pressupostos és la següent:

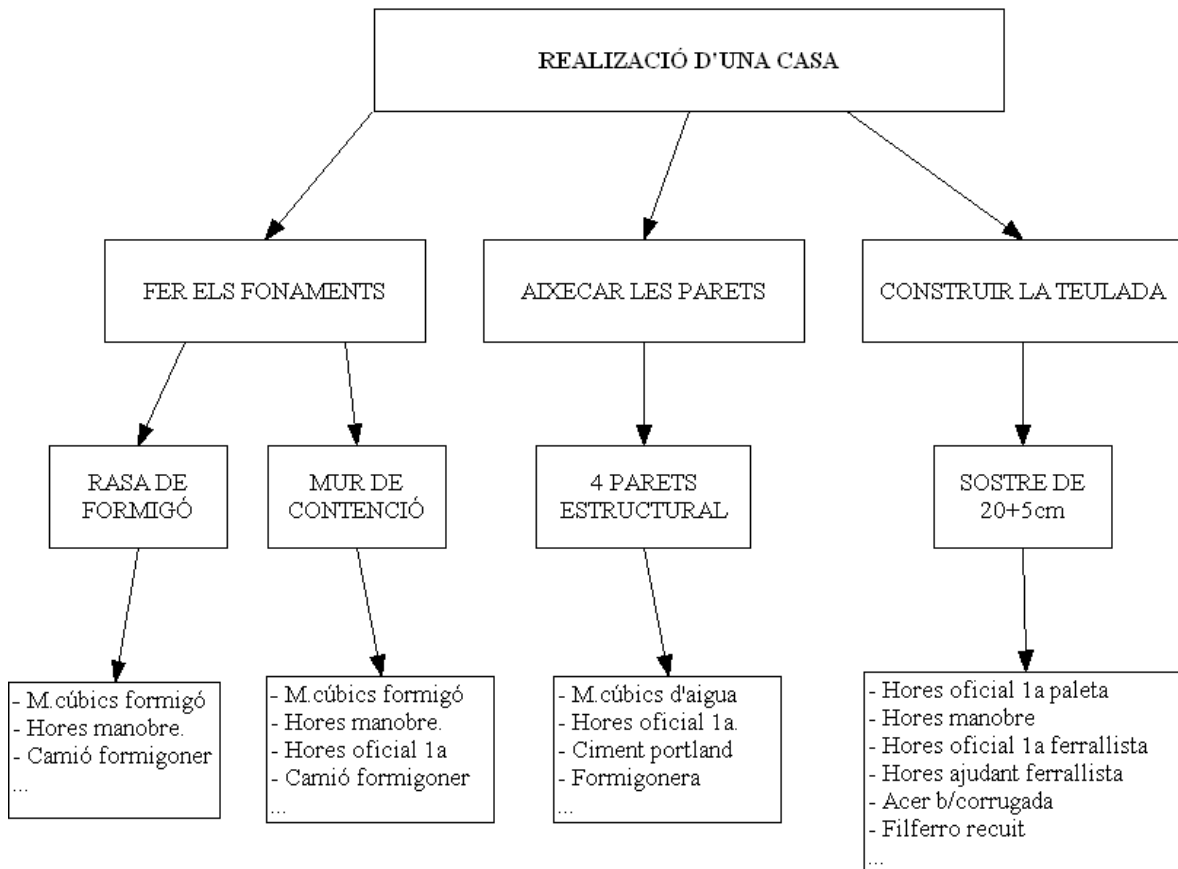
- Un pressupost està format per un o varis capítols
- Cada capítol està format per una o vàries partides.
- Cada partida està formada per una relació d'articles: des de la mà d'obra dels operaris que hi treballen fins a tots els materials necessaris.

Llavors cada material té unes mesures d'una, dues o tres dimensions depenent del tipus. Per exemple, les hores dels operaris tindran una sola dimensió; els metres quadrats de rajola tindran dues dimensions i les metres cúbics d'aigua necessaris en tindran tres de dimensions.

Per exemplificar l'estructura d'un pressupost anem a fer-ne un d'hipotètic:

- El constructor vol realitzar una casa sobre un terreny erm. El primer que haurà de fer seran els fonaments on assentar tota l'estructura; seguidament aixecarà les parets i formarà la distribució de la casa; i per últim es realitzarà la teulada. Aquest conjunt d'accions serien els diversos capítols.  
Per realitzar els fonaments s'ha de realitzar una rasa de formigó armat abocat amb bomba (partida 135138A1) i un mur de contenció també de formigó armat (partida 135238A1). Aquestes dues estructures són el què anomenem partides, les quals estan formades per articles. En el cas de la rasa de formigó hi ha implícit els metres cúbics de formigó necessaris, les hores de manobre per la seva realització, el camió bomba de formigonar... i en el cas del mur de contenció es necessita la mà d'obra d'un oficial de 1a paleta, de manobre, el formigó pertinent, el camió bomba de formigonar...  
Per aixecar les parets es necessiten, entre d'altres coses, 4 parets estructurals de 25 cm de gruix (partida 14H2F665), les quals comprenen les corresponents hores d'oficial de 1a paleta, de manobre, l'aigua, els blocs, el ciment portland... Finalment, per a la realització de la teulada s'haurà de realitzar un sostre de 20+5cm per a una sobrecàrrega de 4000 a 5000 N/m<sup>2</sup> (partida 14LFL68C). El desgloss de la partida comprèn les hores corresponents d'oficial de 1a de paleta, de manobre, d'ajudant de ferrallista, d'oficial de 1a de ferrallista, l'acer necessari, el filferro...

A continuació representem gràficament l'estructura perquè quedi més clar:



A continuació explicarem els diferents requeriments de l'aplicació desenvolupada. Aquests poden ésser de dos tipus diferents: FUNCIONALS i NO FUNCIONALS.

Pel què fa referència als requeriments funcionals especificats anteriorment, s'han obtingut parlant amb els futurs usuaris de l'aplicació sobre la manera que tenen de treballar i com ho organitzen.

En quan als requeriments no funcionals podem esmentar els següents:

- L'aplicació ha estat implementada amb un llenguatge FDS que ja disposa d'un entorn de treball fet i que treballa amb fitxers. Aquest llenguatge està enfocat a la gestió, permet l'ús de diferents empreses i de diferents anys. En quan al tema visual s'ha de realitzar manualment, establint per codi la posició relativa a la pantalla dels textos, dades... a mostrar. És un entorn de pagament i ha estat marcat per l'empresa.
- És un projecte real, demanat per una empresa constructora.
- Aquest projecte s'ha desenvolupat per l'interès econòmic per part de l'empresa desenvolupadora i, per tant, no se n'ha buscat un d'existent.

## 2. Anàlisi del sistema

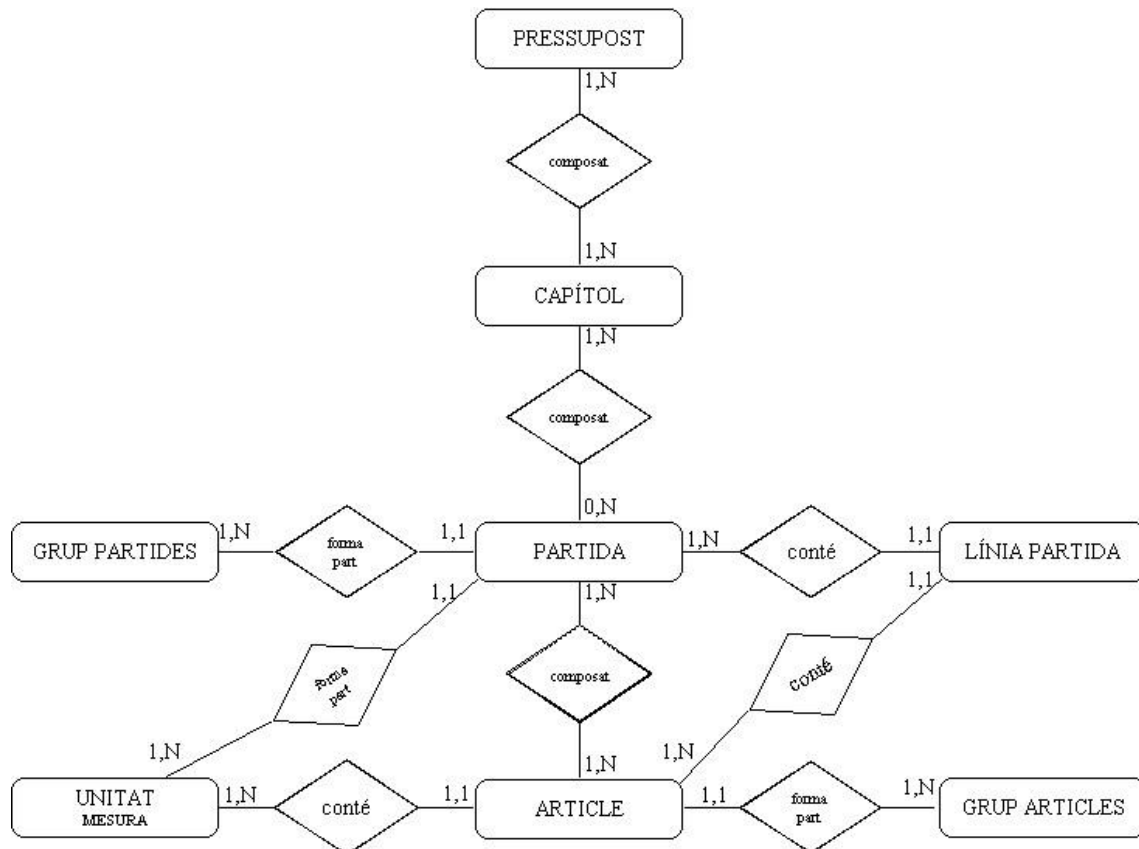
L'objectiu de l'anàlisi és saber quins són els elements que formaran part de la nostra aplicació i la interconnexió entre ells, així com les entitats participants que interactuaran amb el projecte informàtic.

El què pretenem fer en aquest apartat és descriure de manera detallada el què ha de fer el projecte informàtic però sense lligar-ho a cap tecnologia en concret.

Consta de dues parts: el model de dades i el model funcional.

### 2.1. Model de dades

Aquest model s'utilitza per representar la informació que al sistema li interessa guardar i la relació que hi ha entre ella. Per a la realització d'aquest apartat mostrarem un diagrama entitat-relació i en farem la seva corresponent descripció.



A continuació descriurem el model de dades. Cada element que participa en el diagrama Entitat-Relació s'ha de descriure:

ENTITAT: Pressupost  
Codi: numèric  
Data: data  
Client: string  
Adreça Client: string  
Població Client: string  
Província Client: string  
Telèfon client: string  
Descripció 1: string  
Descripció 2: string  
Descripció 3: string  
Descripció 4: string  
Descripció 5: string  
Import: numèric  
Número de Capítols: numèric  
Text Portada: string  
Text Contraportada: string  
Històric: Caràcter  
IDENTIFICADOR: Codi  
RELACIONS:  
Pressupost <composat per> Capítols

ENTITAT: Capítol  
Obra: numèric  
Línia: numèric  
Descripció 1: string  
Descripció 2: string  
Descripció 3: string  
Descripció 4: string  
Import: numèric  
Número de Partides: numèric  
IDENTIFICADORS: Obra, Línia  
RELACIONS:  
Capítol <composat per> Partides

ENTITAT: Partida  
Codi: numèric  
Descripció General: string  
Descripció 1: string  
Descripció 2: string  
Descripció 3: string  
Descripció 4: string  
Descripció 5: string  
Unitat de Mesura: string  
Preu: numèric  
Grup de Partida: string  
IDENTIFICADOR: Codi  
RELACIONS:  
Partida <forma part> Grup de Partides  
Partida <conté> Línies de Partida  
Partida <conté> Unitat de Mesura  
Partida <està composada> Articles



ENTITAT: Línia de Partida  
Codi: string  
Línia: numèric  
Article: string  
Quantitat: numèric  
Preu: numèric  
Import: numèric  
IDENTIFICADORS: Codi, Línia  
RELACIONS:  
Línia de Partida <conté> Article  
Línia de Partida <conté> Partida

Aquesta entitat "Línia de Partida" conté la relació d'articles que formen part de cada partida, amb el seu corresponent preu, la quantitat necessària i l'import total.

ENTITAT: Grup de Partides  
Codi: string  
Descripció: string  
IDENTIFICADOR: Codi  
RELACIONS:  
Grup de Partides <conté> Partides

L'entitat Grup de Partides conté el codi i la descripció dels diferents grups en els quals es poden classificar les partides.

ENTITAT: Grup d'Articles  
Codi: string  
Descripció: string  
IDENTIFICADOR: Codi  
RELACIONS:  
Grup d'Articles <conté> Articles

En aquesta entitat hi ha representats els diferents grups d'articles que podem trobat, des dels diferents acabats i tipus de materials fins a les hores d'operari, manobre, ajudants, peons, oficials...

ENTITAT: Article  
Codi: string  
Descripció: string  
Unitat de mesura: string  
Preu unitari: numèric  
Preu de cost: numèric  
Grup d'Article: string  
Data d'última revisió: data  
IDENTIFICADOR: Codi  
RELACIONS:  
Article <forma part> Partida  
Article <pertany> Grup d'Articles  
Article <conté> Unitat de Mesura  
Article <forma part> Línia de Partida

ENTITAT: Unitat de Mesura  
Codi: string  
Descripció: string  
Alçada: caràcter  
Amplada: caràcter  
Fondària: caràcter  
IDENTIFICADOR: Codi  
RELACIONS:  
Unitat de Mesura <forma part> Partida  
Unitat de Mesura <forma part> Article

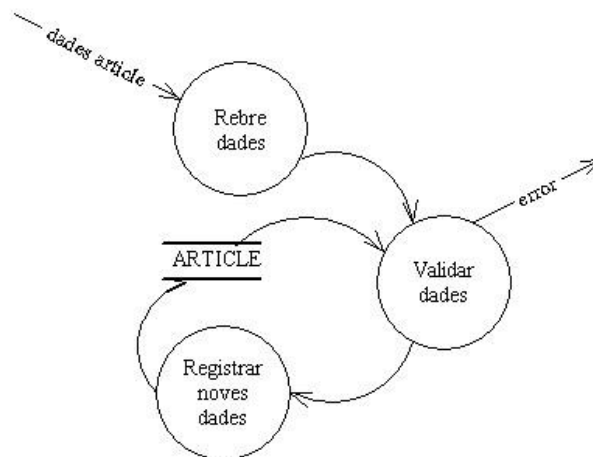
## 2.2. Model funcional

Aquest apartat ens permetrà veure de manera detallada quines són les tasques que ha de dur a terme el sistema. Per a la seva realització farem una llista dels esdeveniments que ha de detectar el futur sistema informàtic i en farem la seva corresponent descripció.

**Nom esdeveniment:** Alta d'un article

**Resposta:** Es registren les dades de l'article

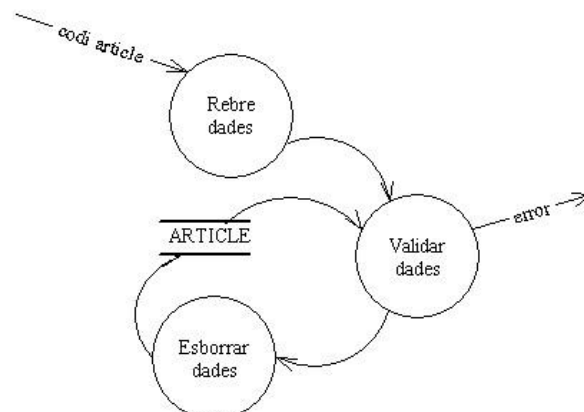
**Procediment:**



**Nom esdeveniment:** Baixa d'un article

**Resposta:** S'esborren les dades de l'article

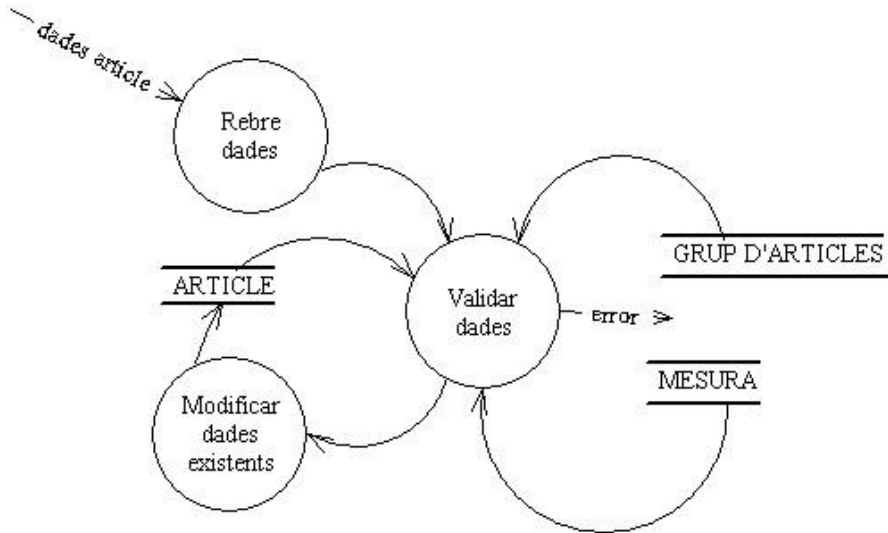
**Procediment:**



**Nom esdeveniment:** Modificar dades d'un article

**Resposta:** Es canvien les dades corresponents de l'article

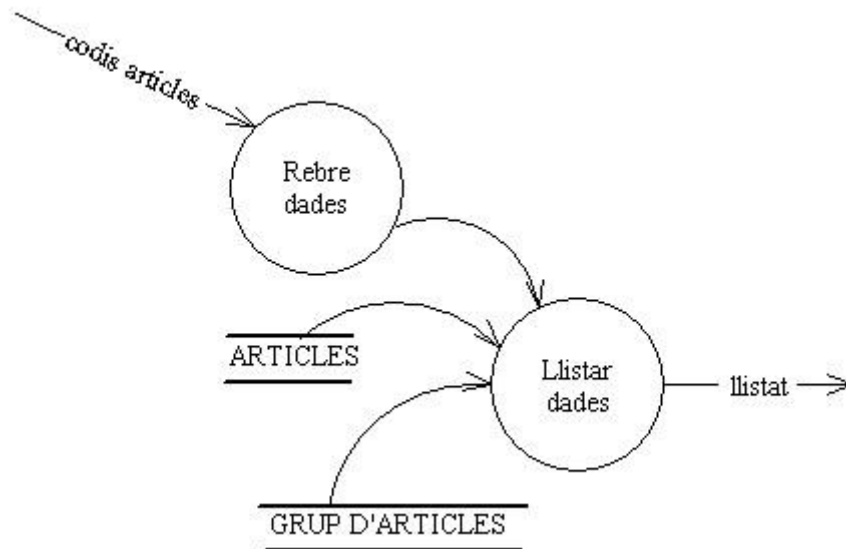
**Procediment:**



**Nom esdeveniment:** Llistat d'articles

**Resposta:** Es llisten els articles i les seves característiques

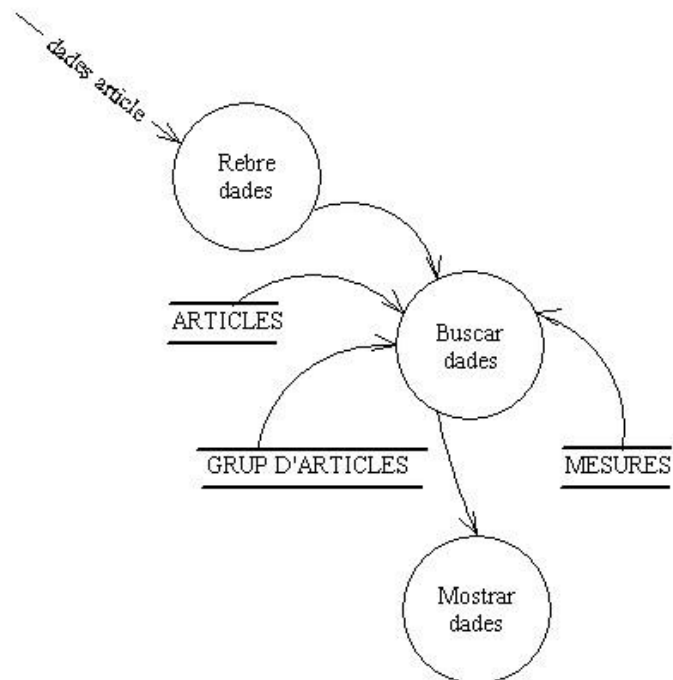
**Procediment:**



**Nom esdeveniment:** Consulta d'un article

**Resposta:** Es busquen i es mostren les dades de l'article

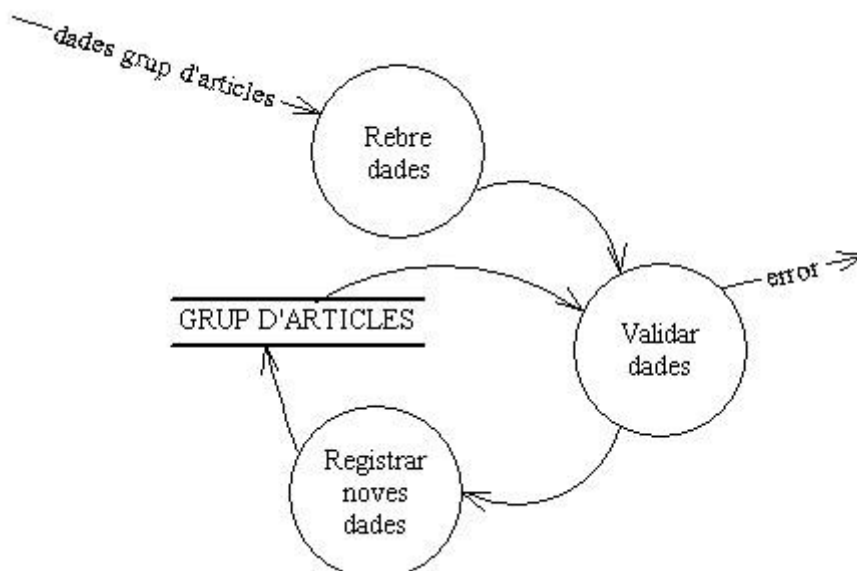
**Procediment:**



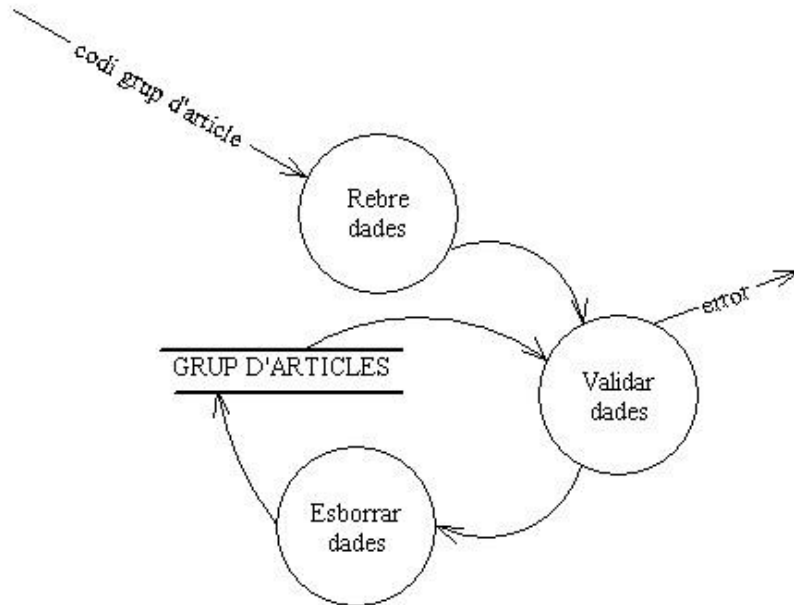
**Nom esdeveniment:** Alta d'un grup d'articles

**Resposta:** Es registren les dades del grup d'articles

**Procediment:**

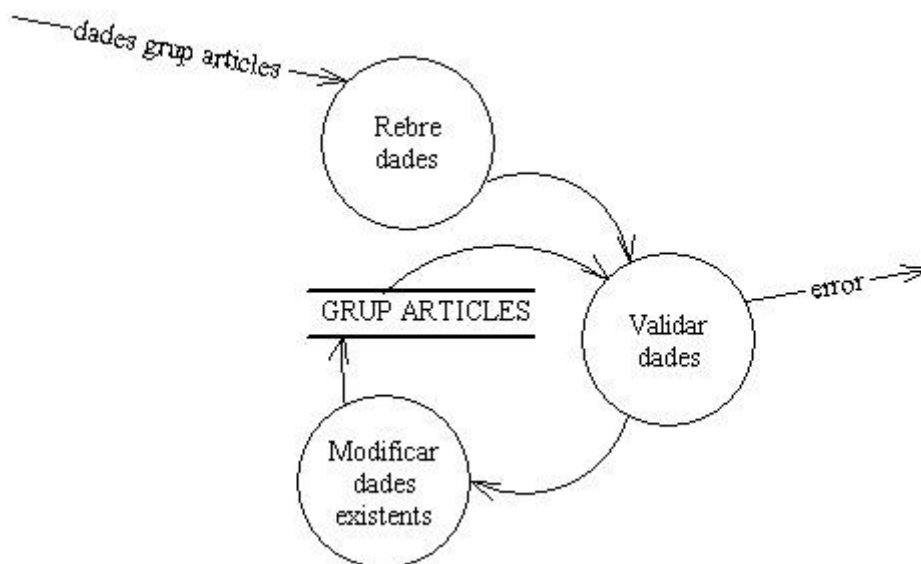


**Nom esdeveniment:** Baixa d'un grup d'articles  
**Resposta:** S'esborren les dades del grup d'articles  
**Procediment:**



Podem donar de baixar un grup d'articles sense comprovar si hi ha algun article que hi pertanyi.

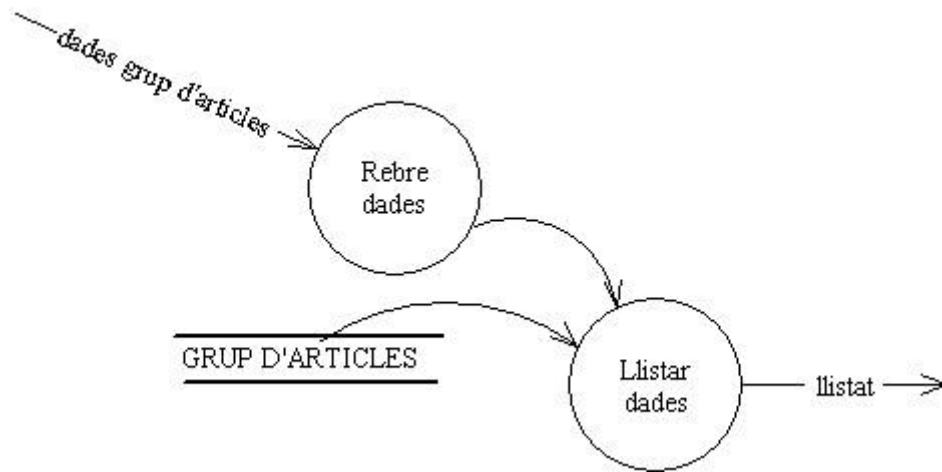
**Nom esdeveniment:** Modificar dades d'un grup d'articles  
**Resposta:** Es canvien les dades corresponents del grup d'articles  
**Procediment:**



**Nom esdeveniment:** Llistat de grups d'articles

**Resposta:** Es llisten els grups d'articles i les seves característiques

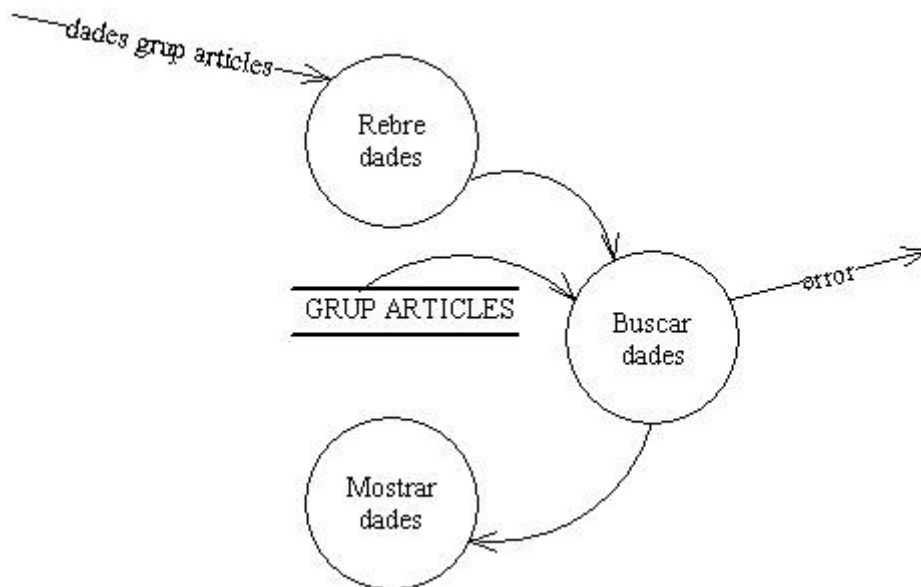
**Procediment:**



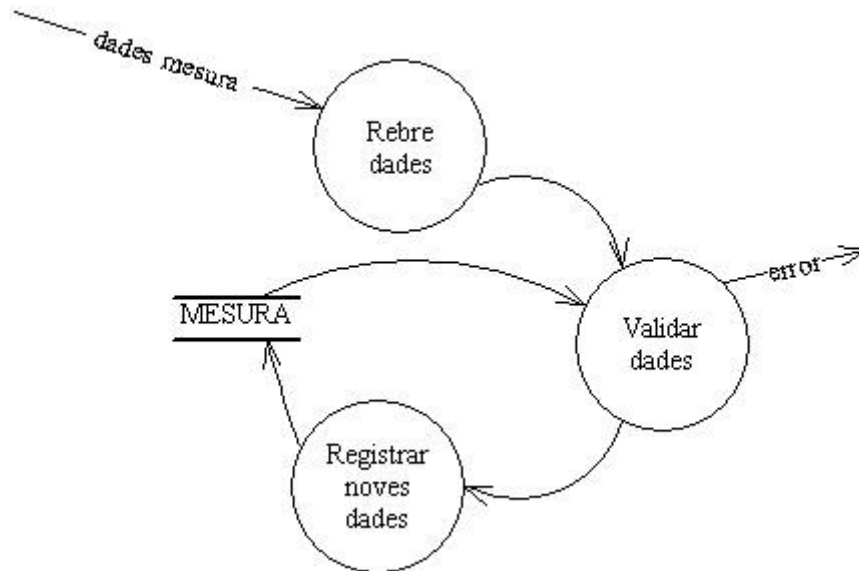
**Nom esdeveniment:** Consulta d'un grup d'articles

**Resposta:** Es busquen i es mostren les dades del grup d'articles

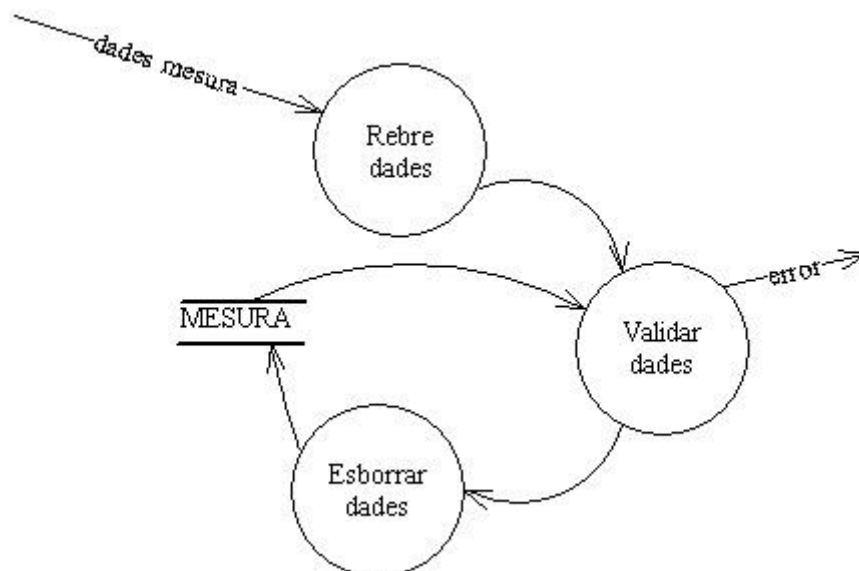
**Procediment:**



**Nom esdeveniment:** Alta d'una mesura  
**Resposta:** Es registren les dades de la mesura  
**Procediment:**



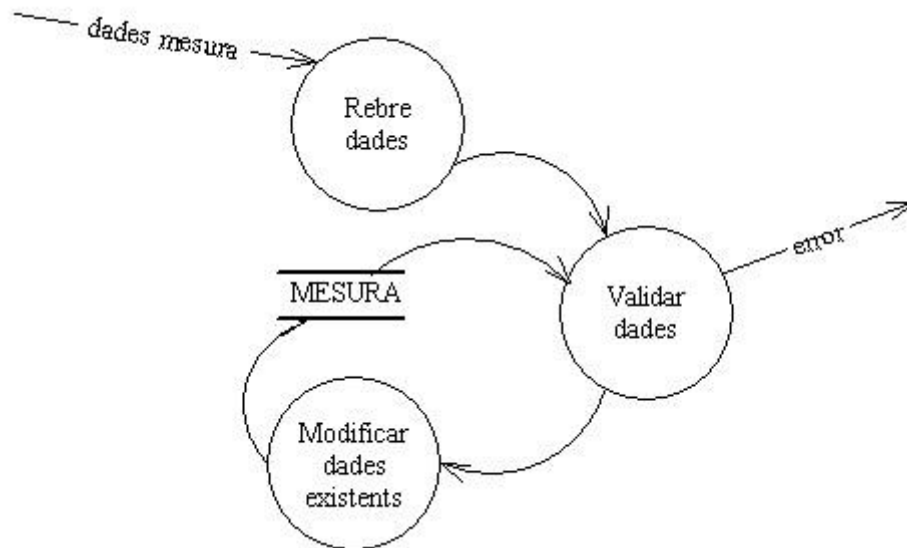
**Nom esdeveniment:** Baixa d'una mesura  
**Resposta:** S'esborren les dades de la mesura  
**Procediment:**



**Nom esdeveniment:** Modificar dades d'una mesura

**Resposta:** Es canvien les dades corresponents de la mesura

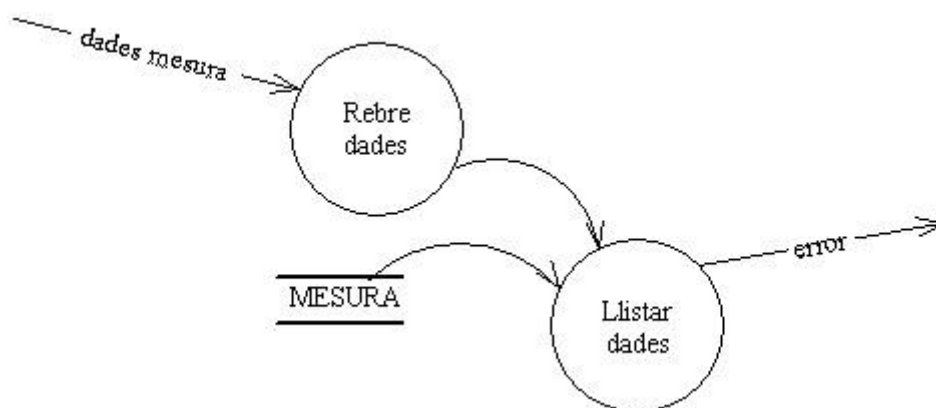
**Procediment:**



**Nom esdeveniment:** Llistat de mesures

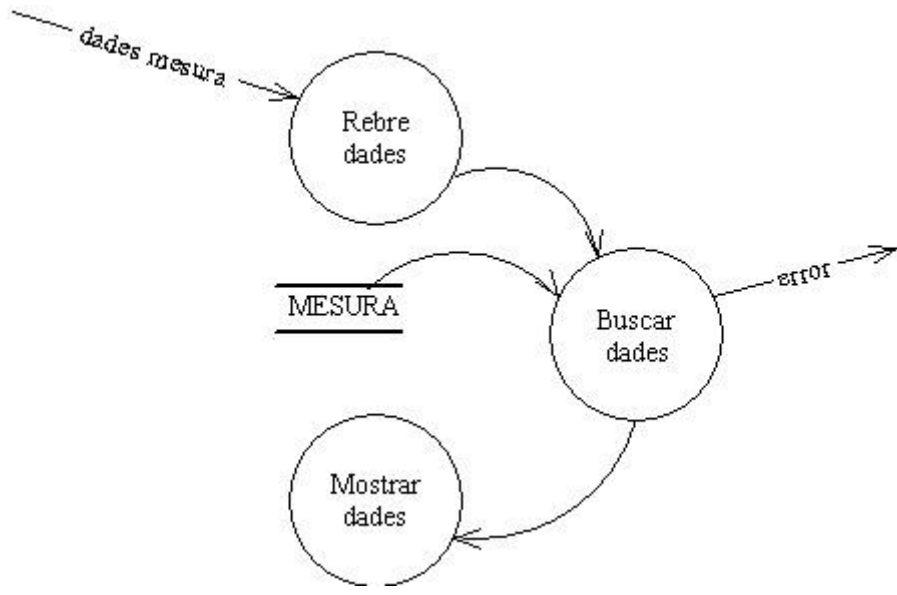
**Resposta:** Es llisten les mesures i les seves característiques

**Procediment:**

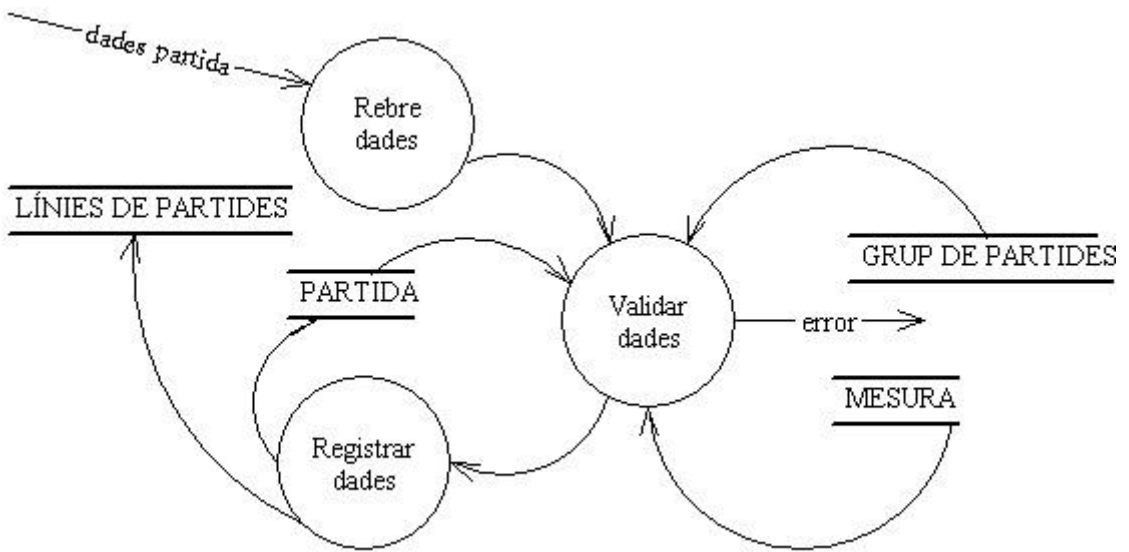




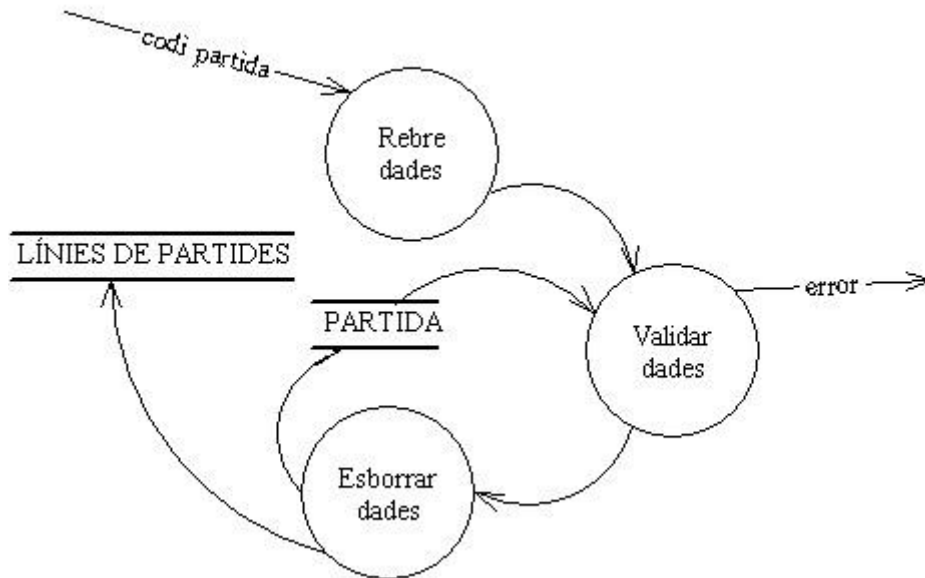
**Nom esdeveniment:** Consulta d'una mesura  
**Resposta:** Es busquen i es mostren les dades de la mesura  
**Procediment:**



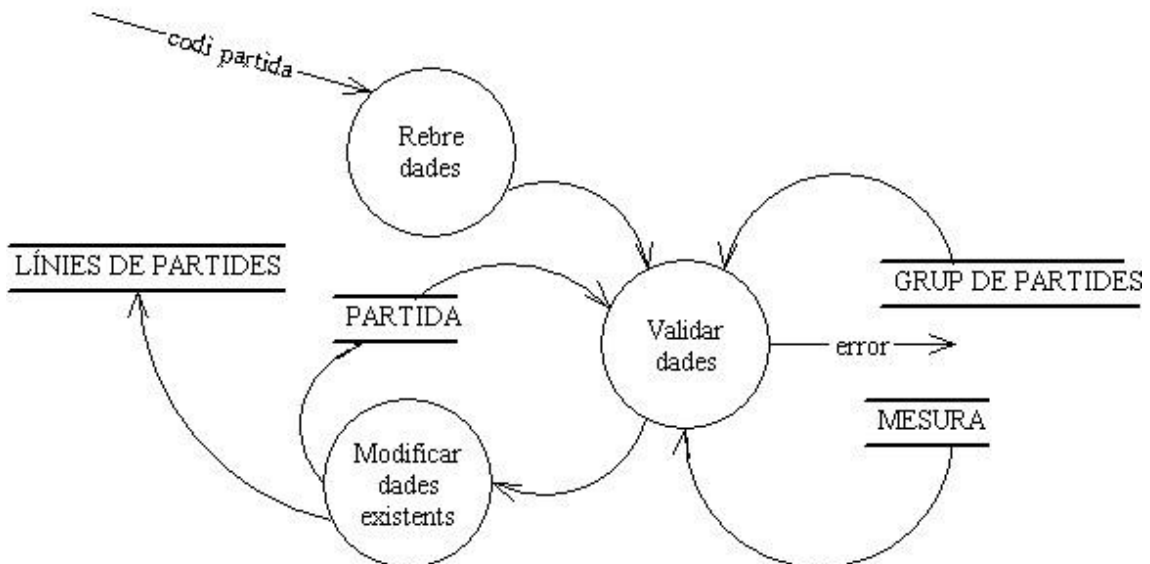
**Nom esdeveniment:** Alta d'una partida  
**Resposta:** Es registren les dades de la partida  
**Procediment:**



**Nom esdeveniment:** Baixa d'una partida  
**Resposta:** S'esborren les dades de la partida  
**Procediment:**



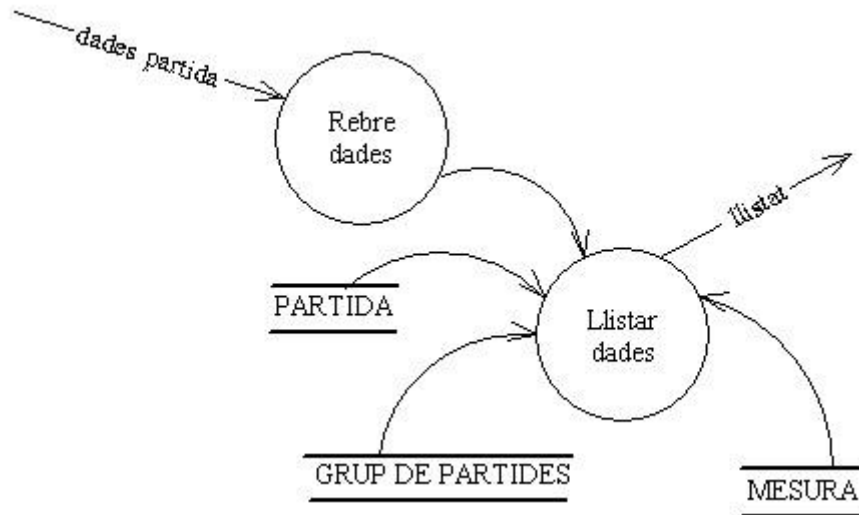
**Nom esdeveniment:** Modificar dades d'una partida  
**Resposta:** Es canvien les dades corresponents de la partida  
**Procediment:**



**Nom esdeveniment:** Llistat de partides

**Resposta:** Es llisten les partides i les seves característiques

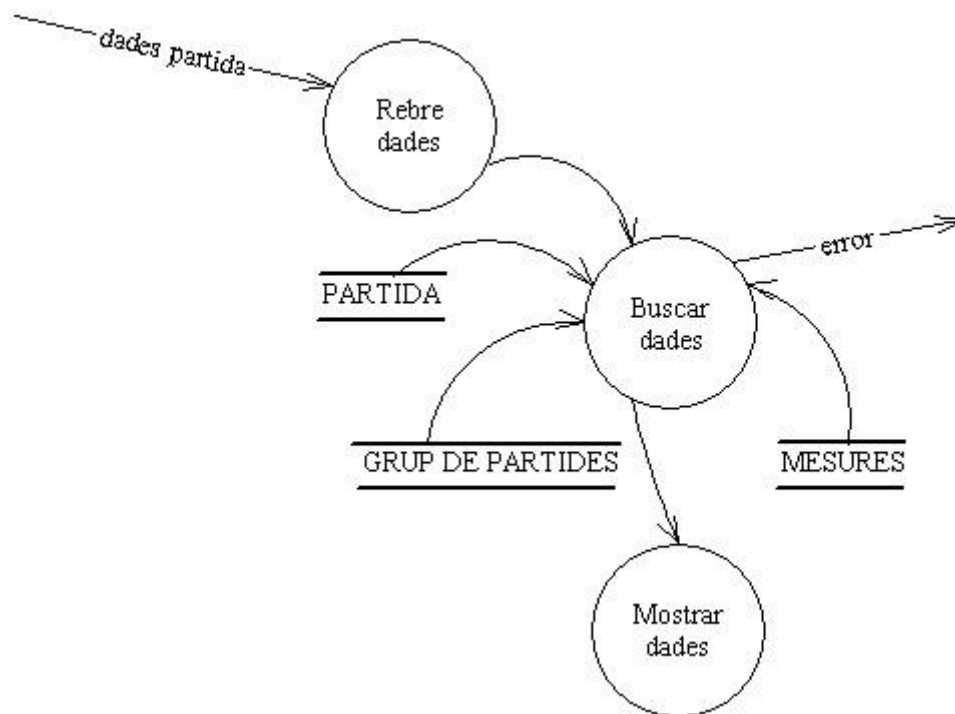
**Procediment:**



**Nom esdeveniment:** Consulta d'una partida

**Resposta:** Es busquen i es mostren les dades de la partida

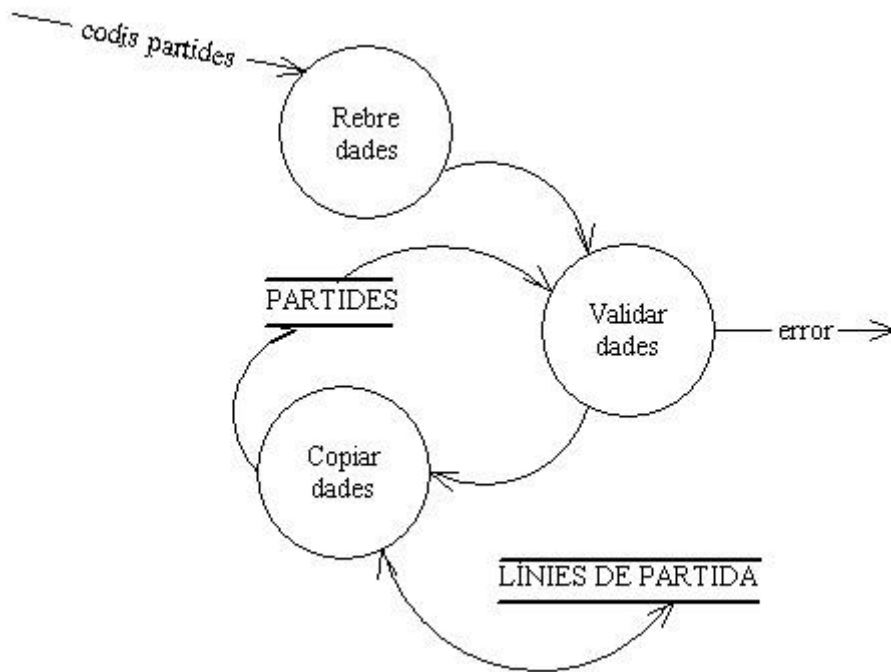
**Procediment:**



**Nom esdeveniment:** Còpia d'una partida

**Resposta:** Copiar tota la partida i els articles que la componen amb un altre codi

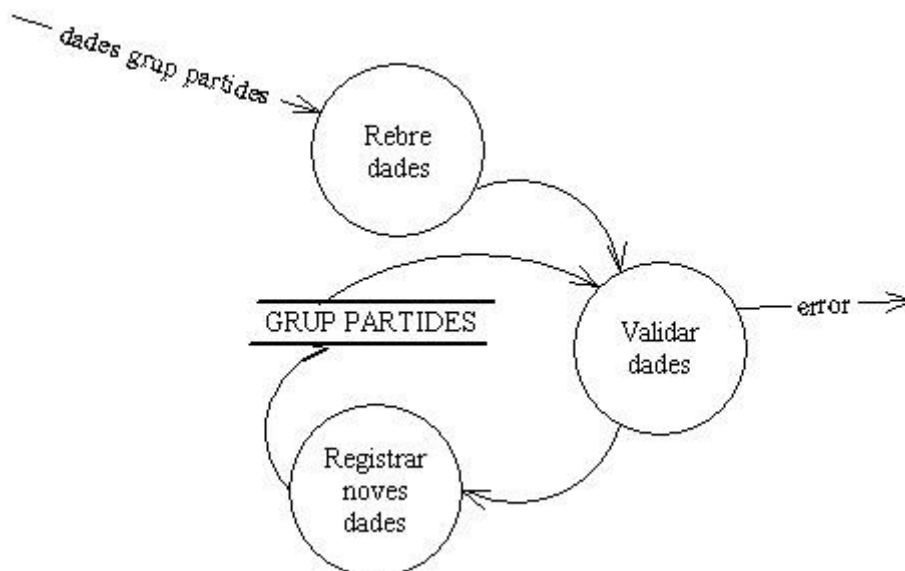
**Procediment:**



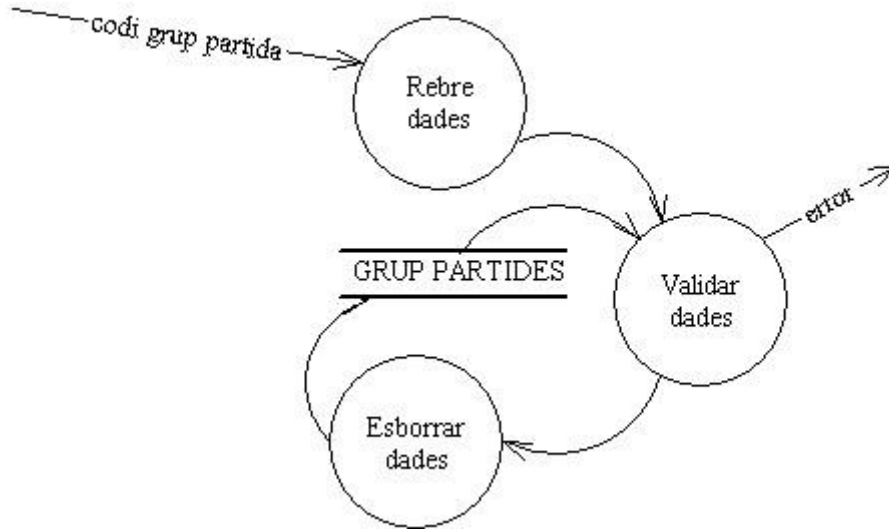
**Nom esdeveniment:** Alta d'un grup de partides

**Resposta:** Es registren les dades del grup de partides

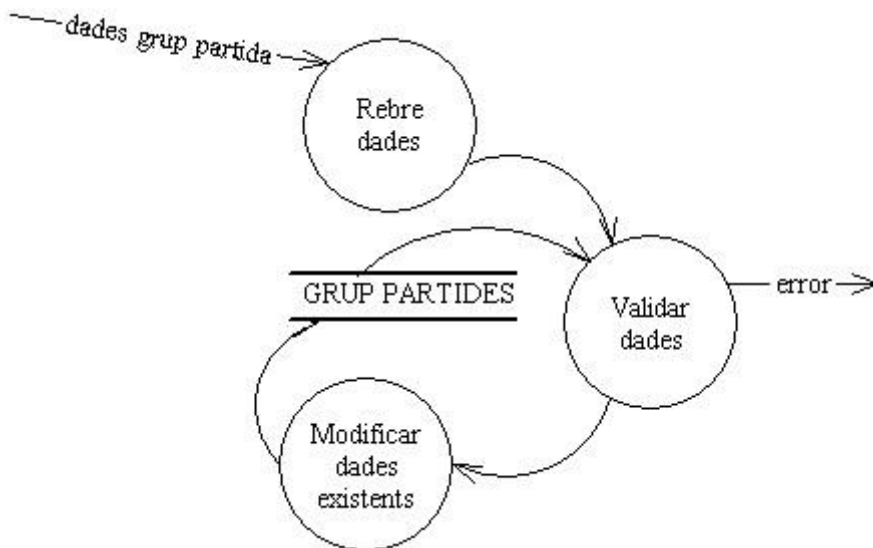
**Procediment:**



**Nom esdeveniment:** Baixa d'un grup de partides  
**Resposta:** S'esborren les dades del grup de partides  
**Procediment:**



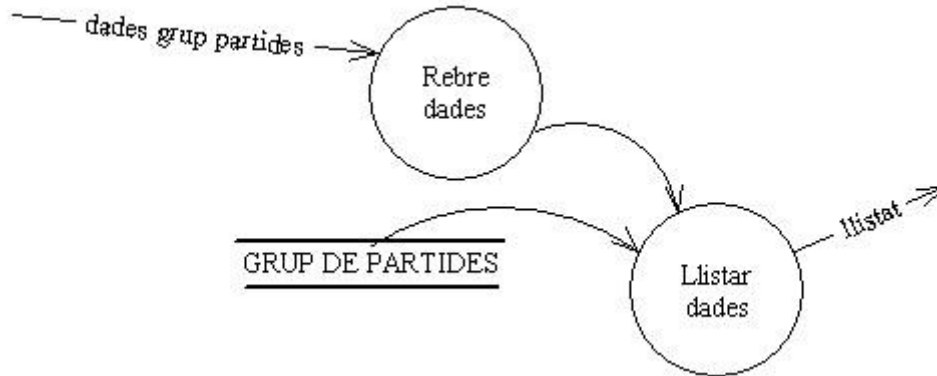
**Nom esdeveniment:** Modificar dades d'un grup de partides  
**Resposta:** Es canvien les dades corresponents del grup de partides  
**Procediment:**



**Nom esdeveniment:** Llistat de grups de partides

**Resposta:** Es llisten els grups de partides i les seves característiques

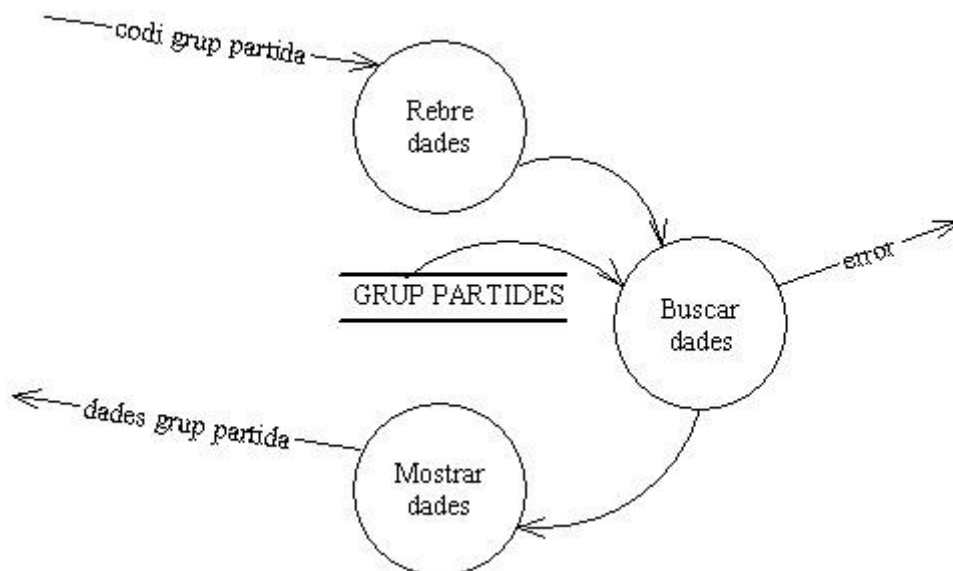
**Procediment:**



**Nom esdeveniment:** Consulta d'un grup de partides

**Resposta:** Es busquen i es mostren les dades del grup de partides

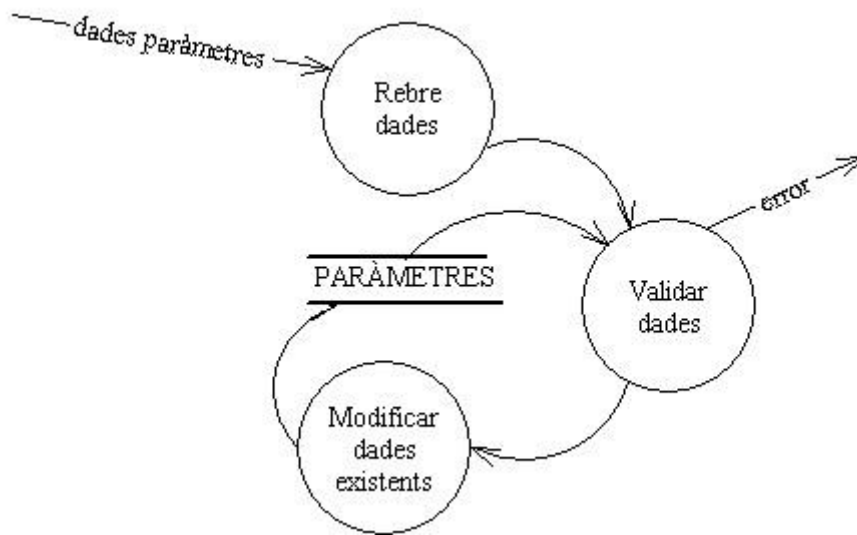
**Procediment:**



**Nom esdeveniment:** Modificar paràmetres generals

**Resposta:** Es canvien les dades genèriques de l'aplicació

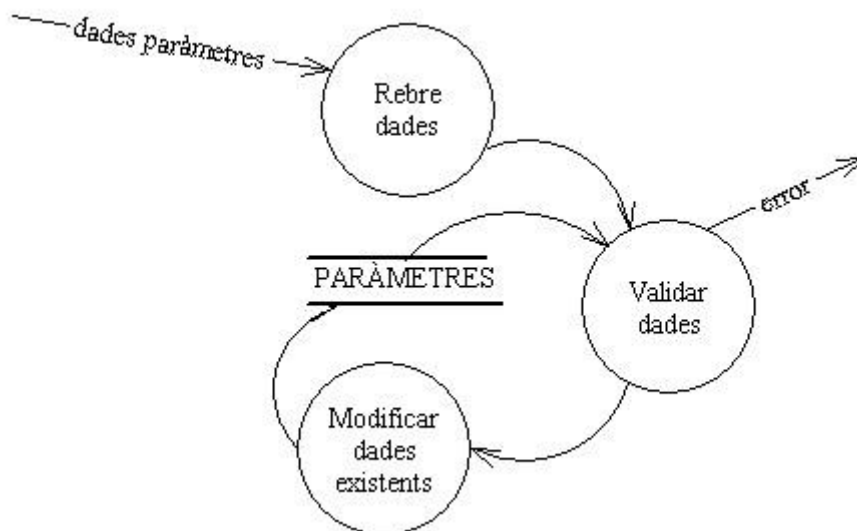
**Procediment:**



**Nom esdeveniment:** Establir text estàndard de la portada

**Resposta:** Editar el text estàndard que sortirà a la portada impresa del pressupost.

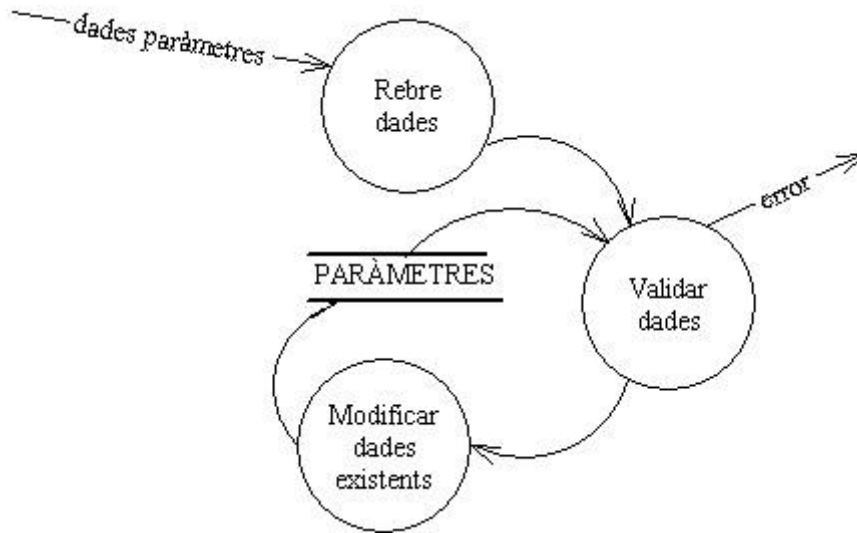
**Procediment:**



**Nom esdeveniment:** Establir text estàndard de la contraportada

**Resposta:** Editar el text estàndard que sortirà a la contraportada impresa del pressupost.

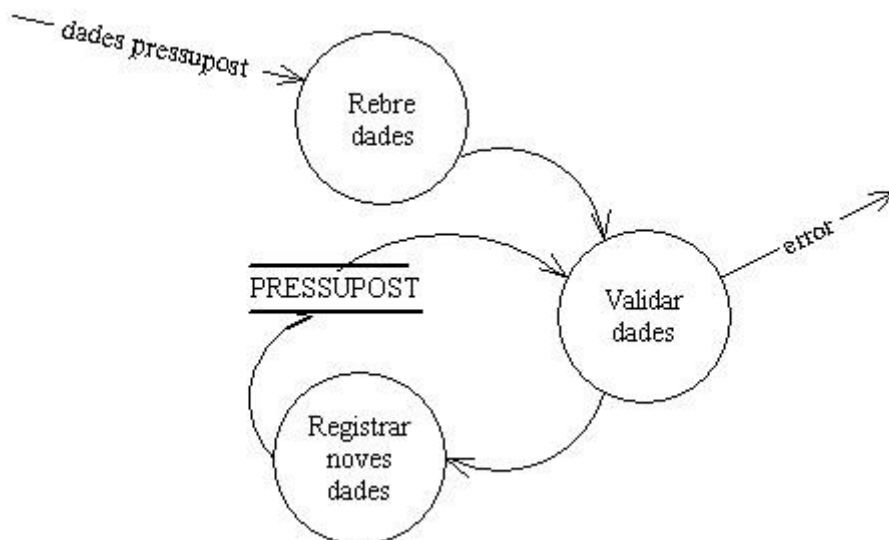
**Procediment:**



**Nom esdeveniment:** Alta d'un pressupost

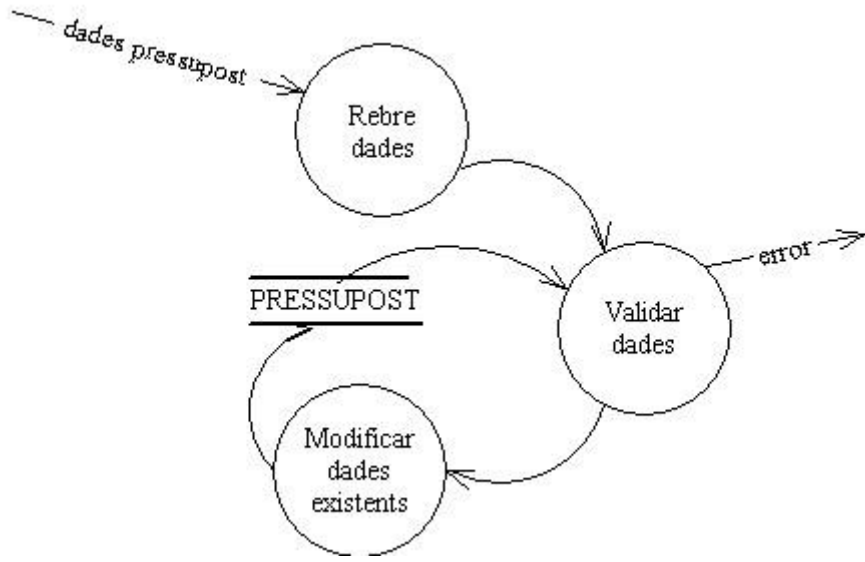
**Resposta:** Es registren les dades del pressupost.

**Procediment:**

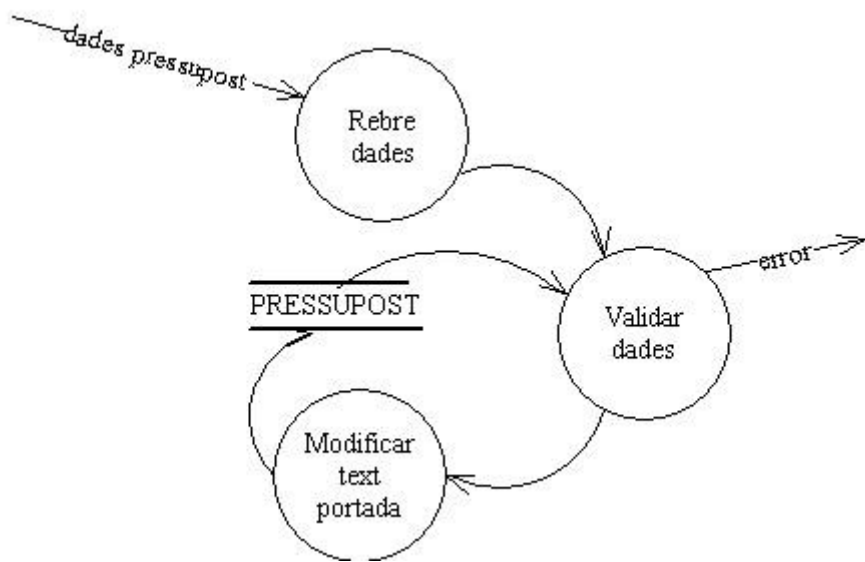




**Nom esdeveniment:** Modificació de les dades principals d'un pressupost  
**Resposta:** Es permeten modificar les dades principals referents al pressupost.  
**Procediment:**



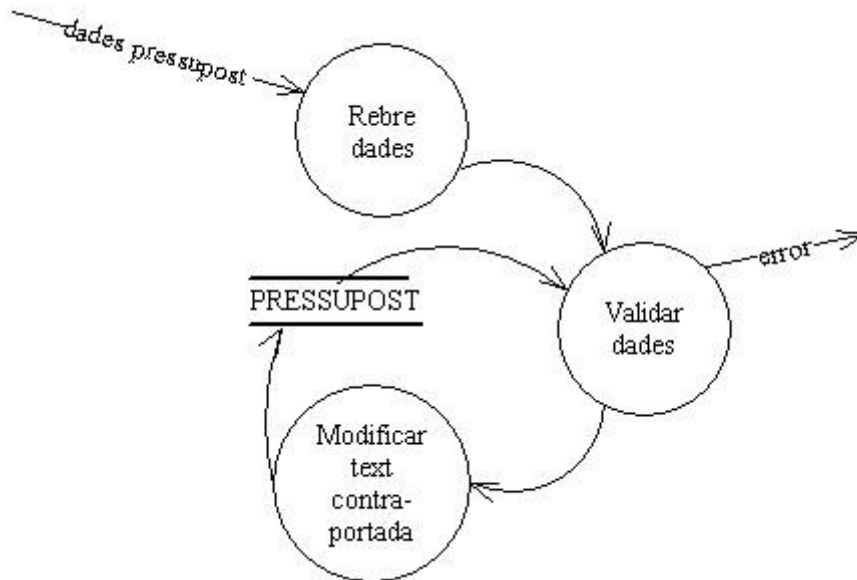
**Nom esdeveniment:** Establir text personalitzat de la portada  
**Resposta:** Editar el text personalitzat que sortirà a la portada impresa del pressupost.  
**Procediment:**



**Nom esdeveniment:** Establir text personalitzat de la contraportada

**Resposta:** Editar el text personalitzat que sortirà a la contraportada impresa del pressupost.

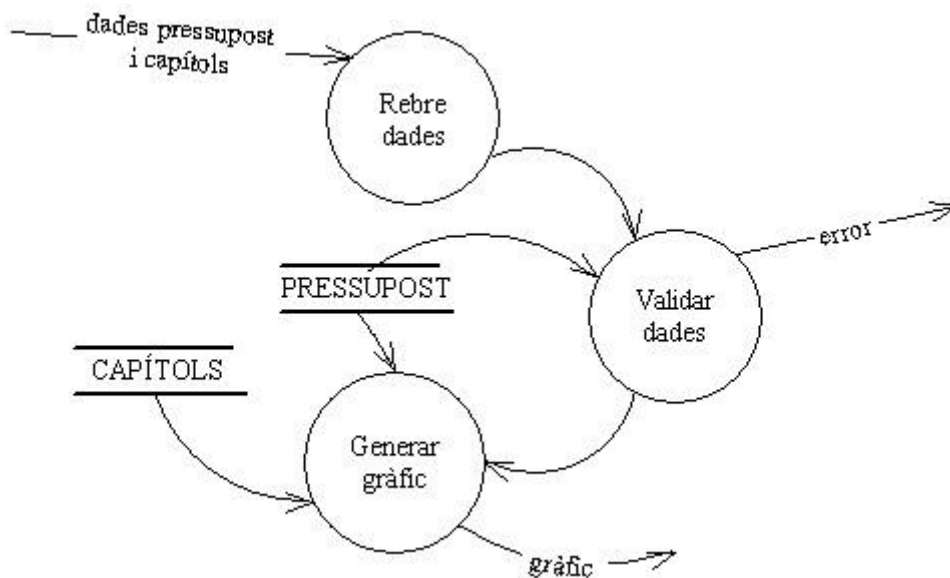
**Procediment:**



**Nom esdeveniment:** Generar gràfic dels imports dels capítols

**Resposta:** Es genera i es mostra un gràfic amb els percentatges d'imports de cada capítol del pressupost.

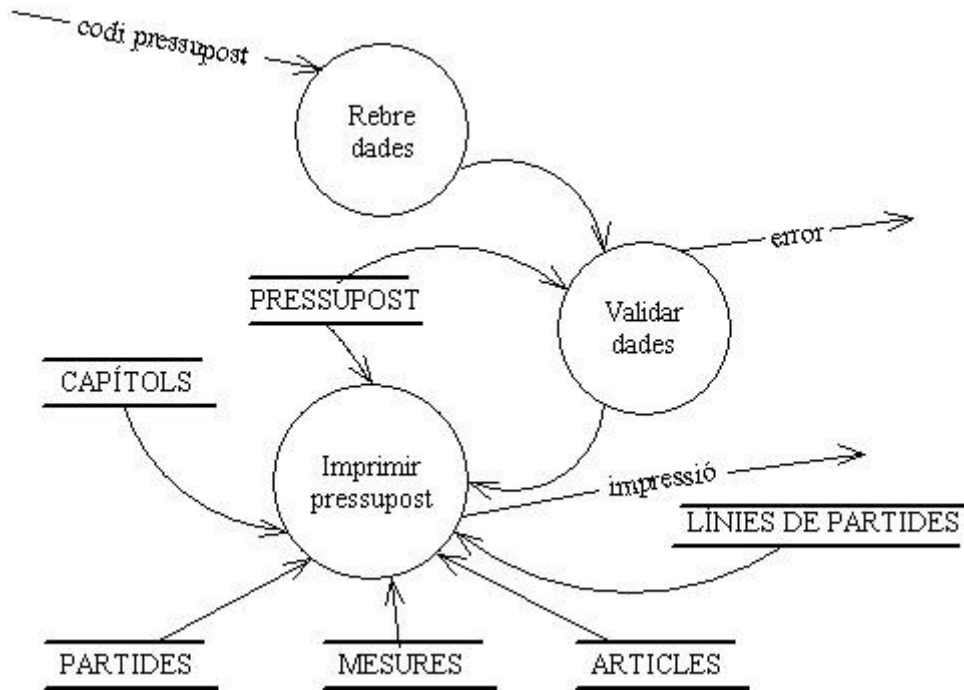
**Procediment:**



**Nom esdeveniment:** Imprimir un pressupost

**Resposta:** S'imprimeix el pressupost en concret amb les seves corresponents dades.

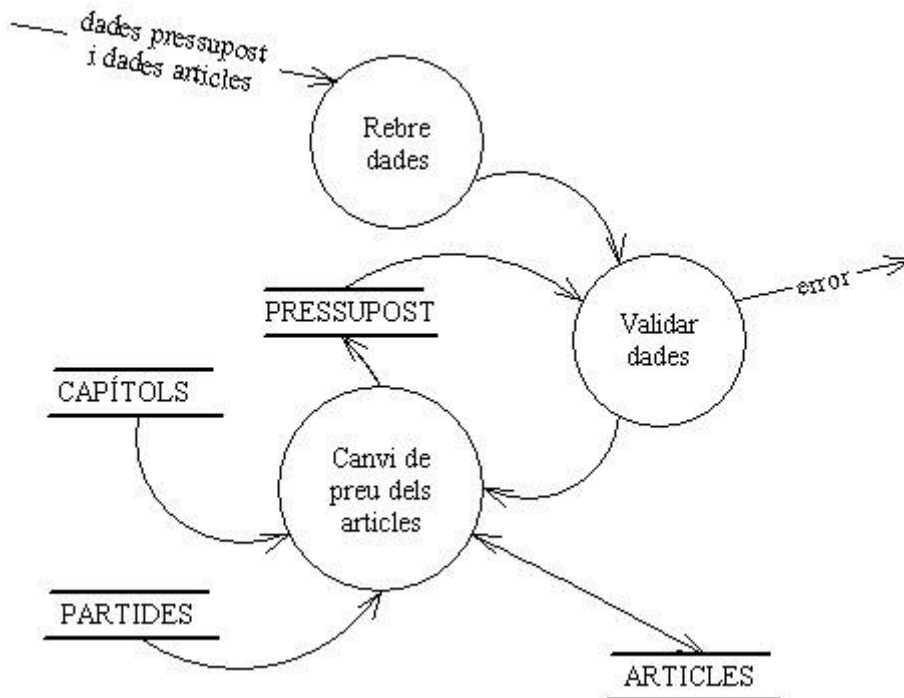
**Procediment:**



**Nom esdeveniment:** Canvi de preu dels articles

**Resposta:** Es canvia el preu dels articles seleccionats.

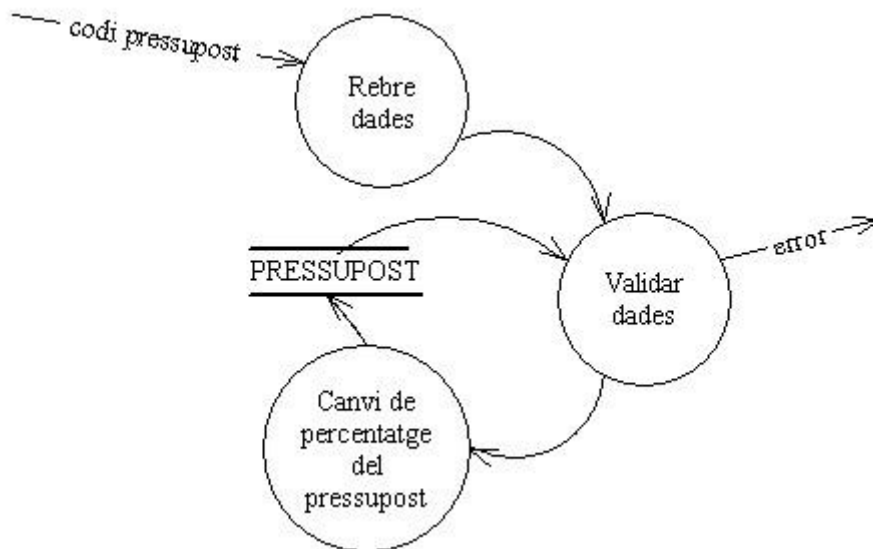
**Procediment:**



**Nom esdeveniment:** Canvi de percentatge del pressupost

**Resposta:** Es canvia el preu total del pressupost segons un percentatge entrat.

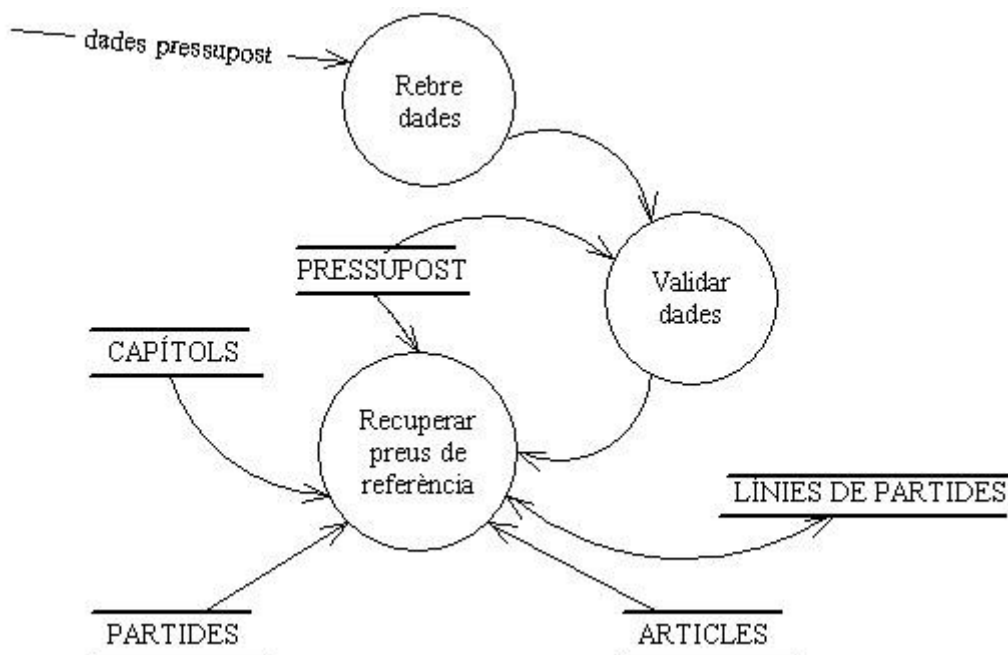
**Procediment:**



**Nom esdeveniment:** Recuperar els preus de referència dels articles

**Resposta:** Es recuperen els preus de referència dels articles dels capítols i partides seleccionats.

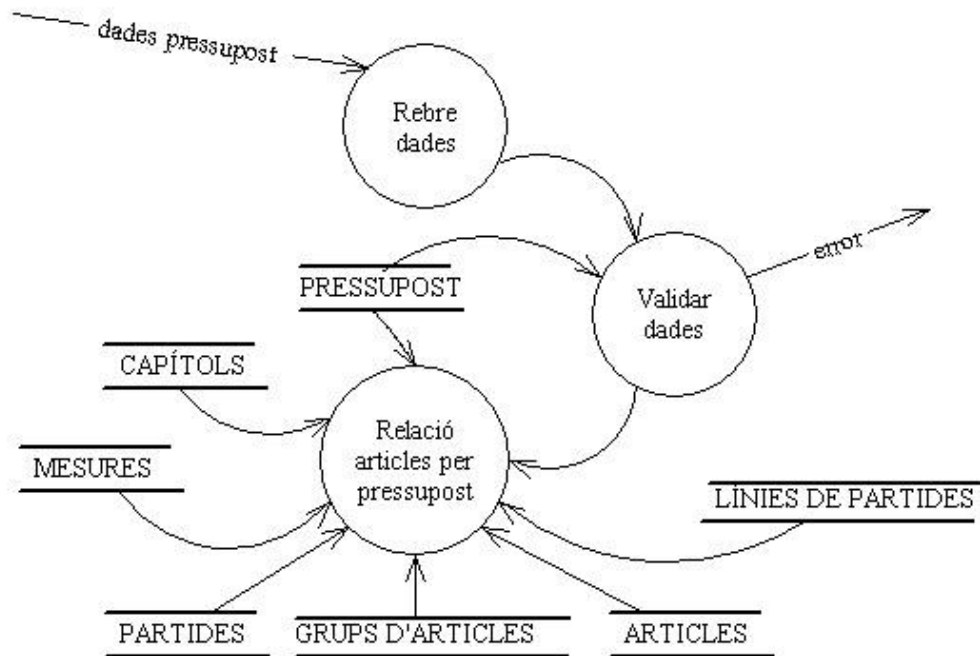
**Procediment:**



**Nom esdeveniment:** Llistat d'articles per pressupost

**Resposta:** Es llisten els articles corresponents al pressupost.

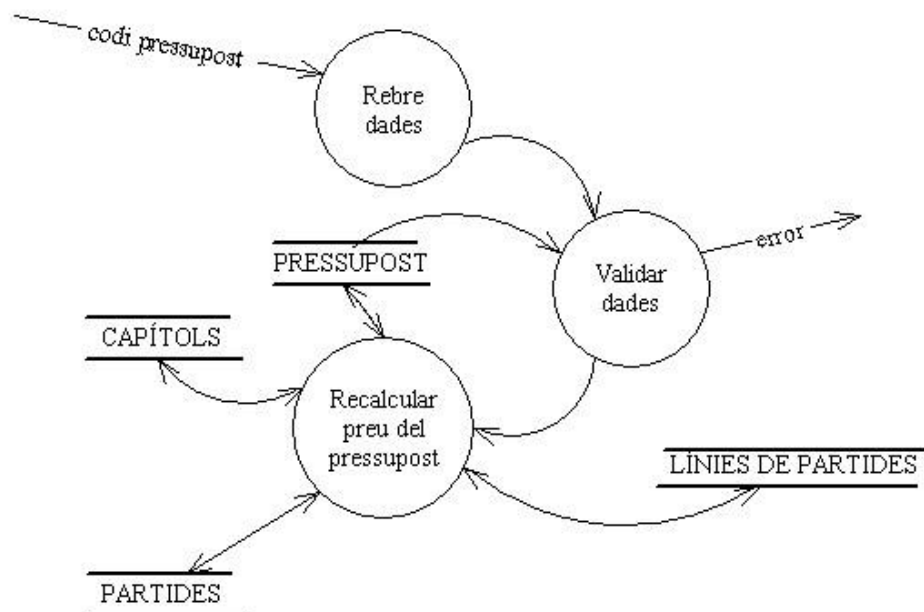
**Procediment:**



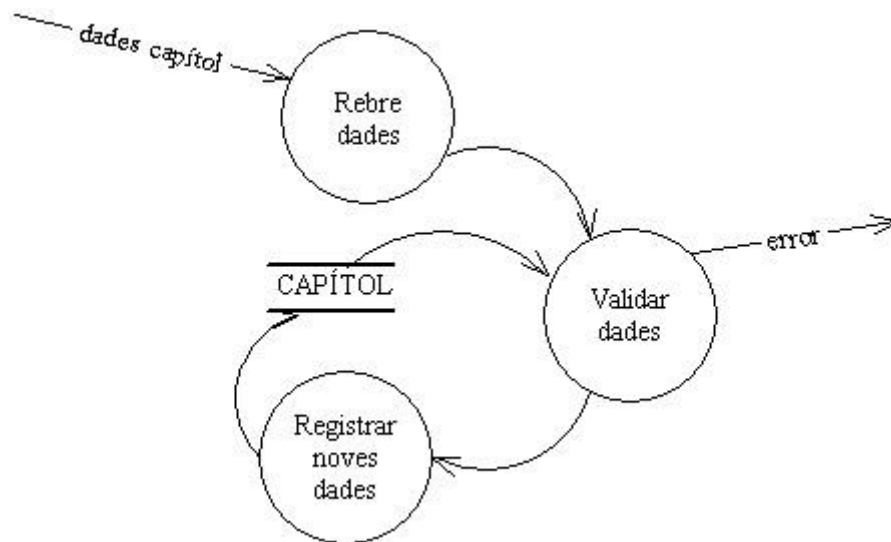
**Nom esdeveniment:** Recàlcul del preu del pressupost

**Resposta:** Es recalcula el preu de tot el pressupost, els articles, les partides i els capítols.

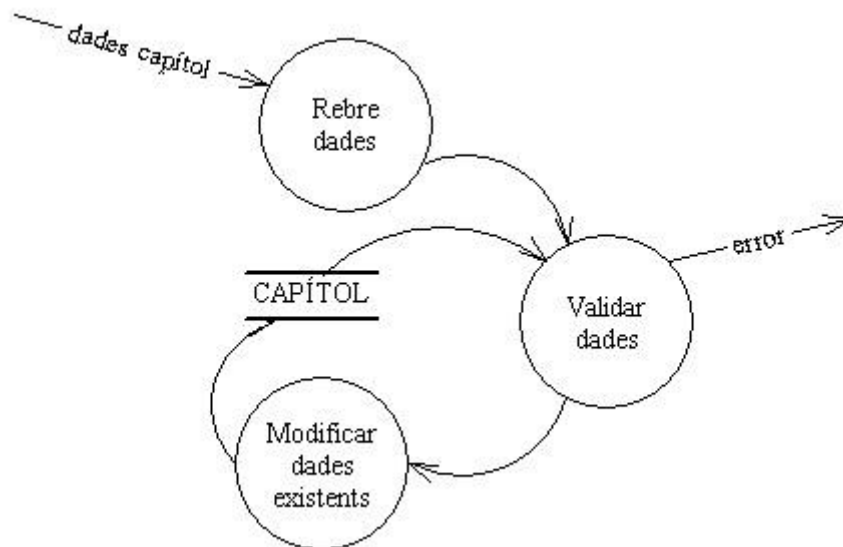
**Procediment:**



**Nom esdeveniment:** Alta d'un capítol  
**Resposta:** Es registren les dades del capítol.  
**Procediment:**



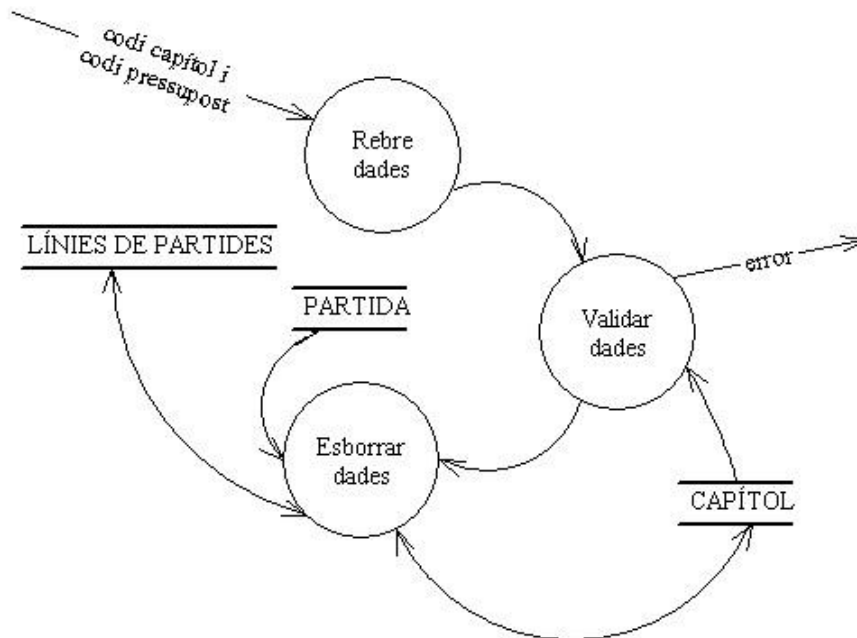
**Nom esdeveniment:** Modificació d'un capítol  
**Resposta:** Es modifiquen les dades del capítol.  
**Procediment:**



**Nom esdeveniment:** Baixa d'un capítol

**Resposta:** S'esborren les dades del capítol i del detall de partides i articles corresponent.

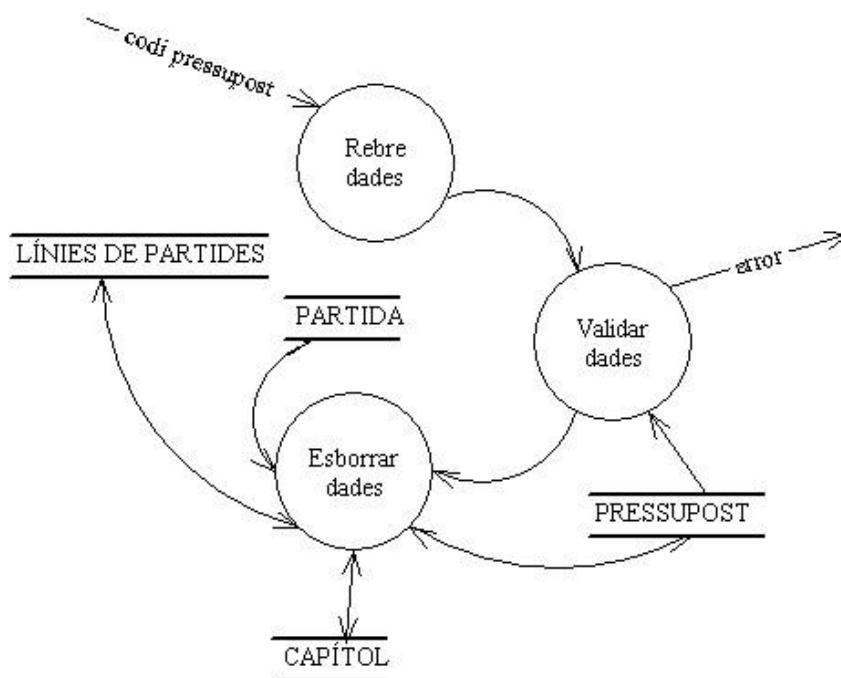
**Procediment:**



**Nom esdeveniment:** Baixa d'un pressupost

**Resposta:** Esborrar totes les dades corresponents al pressupost.

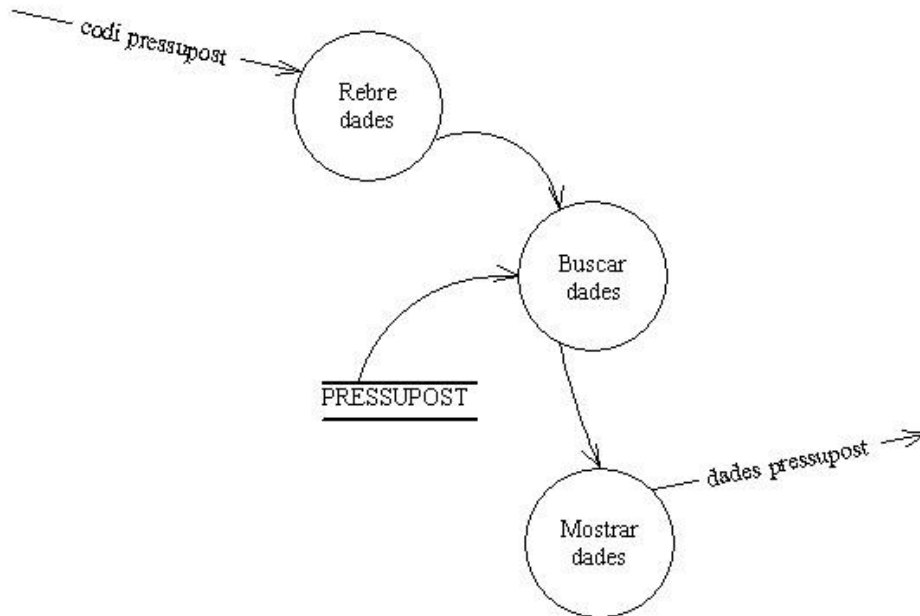
**Procediment:**



**Nom esdeveniment:** Consulta d'un pressupost

**Resposta:** Es busquen i es mostren les dades del pressupost.

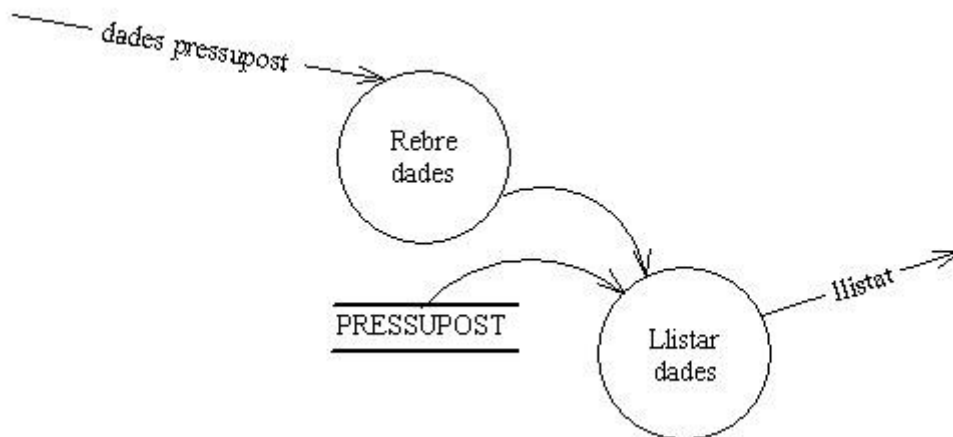
**Procediment:**



**Nom esdeveniment:** Llistat de pressupostos

**Resposta:** Es llisten els pressupostos i les seves dades principals.

**Procediment:**

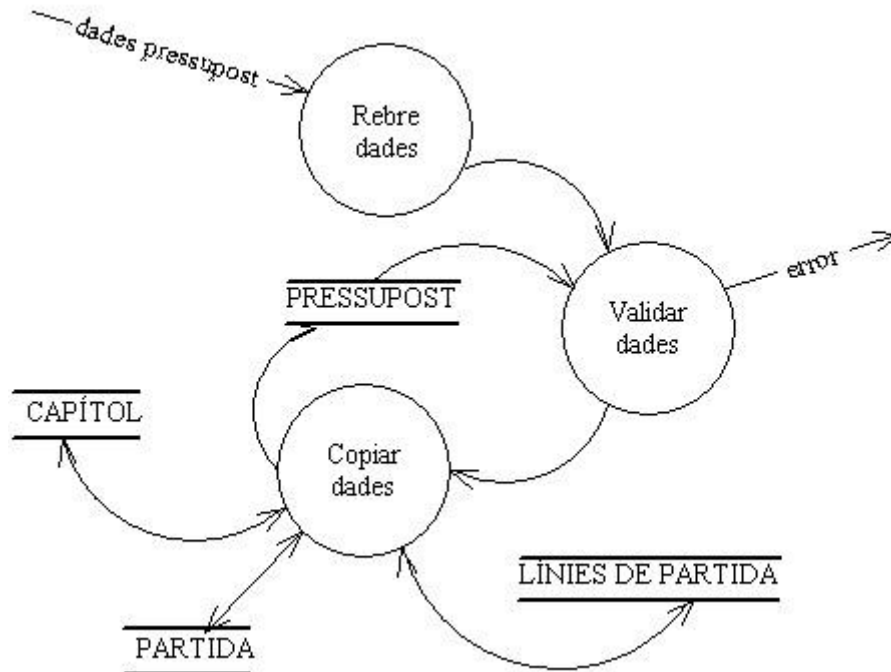




**Nom esdeveniment:** Còpia d'un pressupost

**Resposta:** Copiar totes les dades del pressupost, els seus capítols, les seves partides i les línies amb un altre codi de pressupost.

**Procediment:**



### 3. Disseny del sistema

L'etapa de disseny consta de diverses fases per tal de lligar el projecte a una tecnologia concreta. Per a la seva realització, ho distribuïrem en tres grans apartats:

1. *Disseny d'interfícies*: Detallarem la part d'interacció amb l'usuari i explicarem les característiques que han de complir les interfícies per assegurar-ne un bon disseny.
2. *Disseny de fitxers*: Especificarem els fitxers utilitzats i que han sortit després del diagrama Entitat – Relació.
3. *Disseny de programes*: Explicarem el llenguatge utilitzat, les característiques que té i en posarem exemples.

#### 3.1. Disseny d'interfícies

Per tal d'assegurar un bon disseny i mantenir una coherència visual en les interfícies amb l'usuari, hem tingut en compte una sèrie de directrius.

A nivell general hem realitzat les interfícies amb les següents característiques:

- *Mínim esforç (realimentació)*: Hem seguit el mateix model per a totes les interfícies per facilitar la feina tan al programador com a l'usuari final per adaptar-se al programa.
- *Mínima memòria*: Hem utilitzat els botons, camps i títols justos i necessaris per a poder realitzar totes les funcions associades a la interfície.
- *Ús de patrons de conducta*: Hem col·locat els botons sempre en les mateixes posicions i amb els mateixos gràfics, els títols tenen la mateixa mida, el mateix color, la disposició dels camps i les graelles és equivalent també
- *Notificació de problemes d'immediat*: Si hi ha alguna acció no permesa o algun error a l'hora d'entrar algun camp es notifica immediatament a l'usuari mitjançant un missatge en pantalla i no deixem que segueixi el flux fins que no s'hagi solventat l'error.
- *Tolerància en errors*: En cas que hi hagi algun error permetem corregir-lo o bé abandonar l'operació en curs si és necessari.
- *Confirmació en accions destructives*: Sempre que es realitza alguna baixa d'algun registre i/o registres es mostra un missatge de confirmació per a poder continuar.

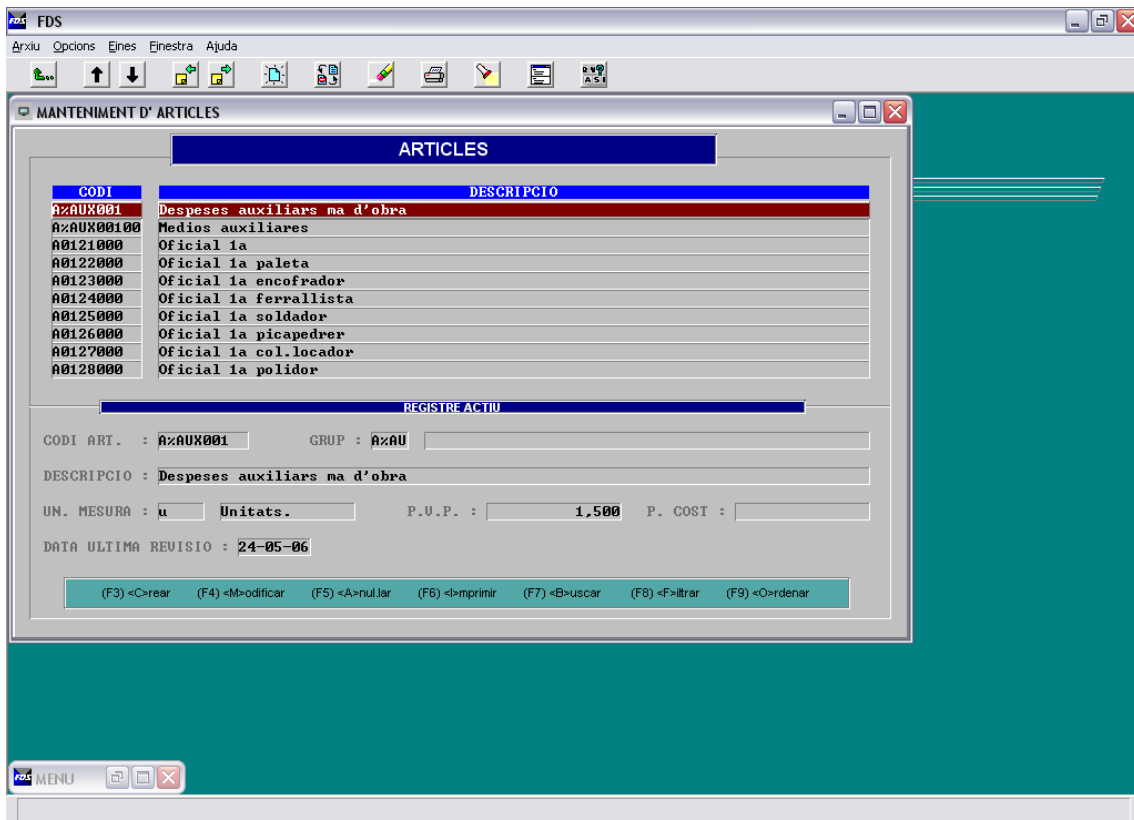
Pel què fa referència a les interfícies de sortida, hem tingut en compte els següents criteris:

- *Només visualitzar la informació rellevant*: En el cas dels llistats, impressions i gràfics, per norma general no se solen mostrar els codis interns d'articles o mesures i només se'n mostra la corresponent descripció.
- *Bon ús de la superfície*: Hem utilitzat tot l'ample i llargària possibles en els llistats posant-hi el màxim de dades possibles sense que la llegibilitat es veigués compromesa.
- *Missatges d'errors significatius*: Quan hem mostrat algun error d'execució per part de l'usuari, s'informa del motiu que l'ha portat a terme.

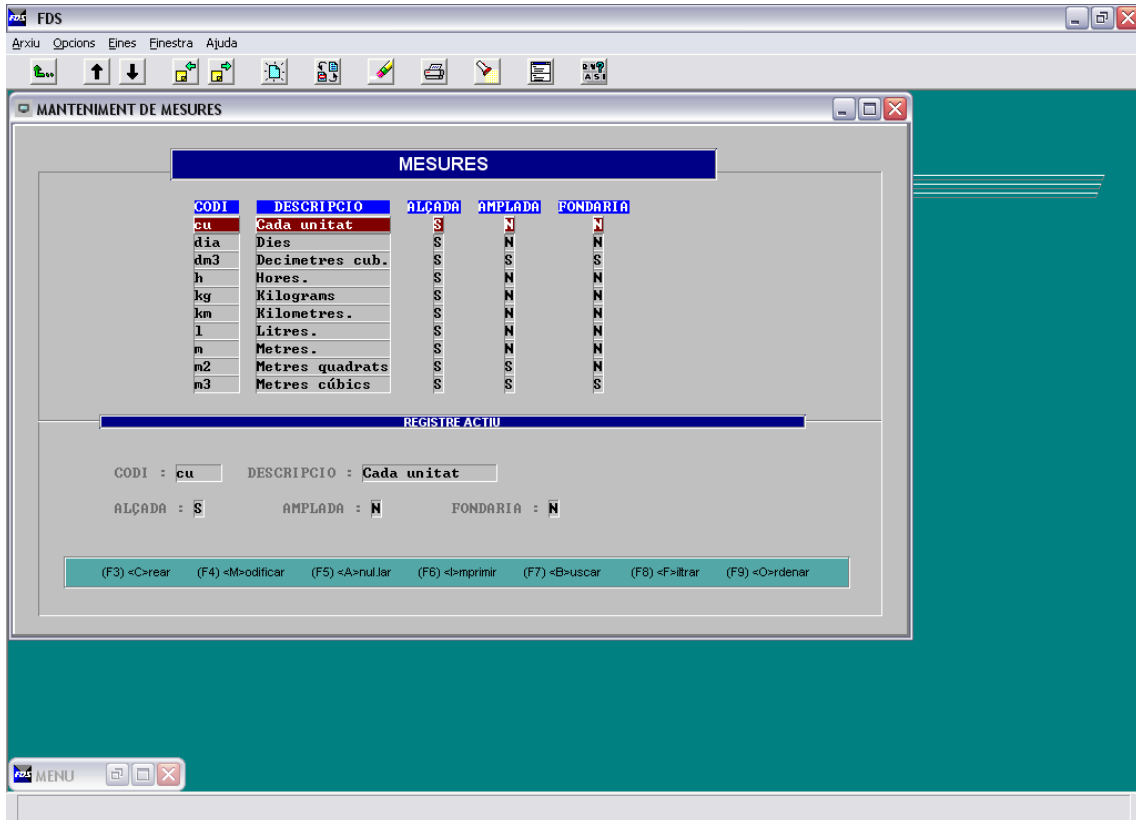
En el cas de l'entrada de dades, hem considerat que haviem de seguir unes conductes també:

- *Desactivar les ordres no pròpies del context:* Quan l'usuari, per exemple, està realitzant una alta d'un article, no pot realitzar alhora una altra acció sobre un altre article. D'aquesta manera evitem errors d'integritat
- *Minimització d'entrada de dades:* Sempre que l'usuari ha d'entrar unes dades, només ha de modificar les dades corresponents a l'apartat sobre el qual s'ha fet la modificació.

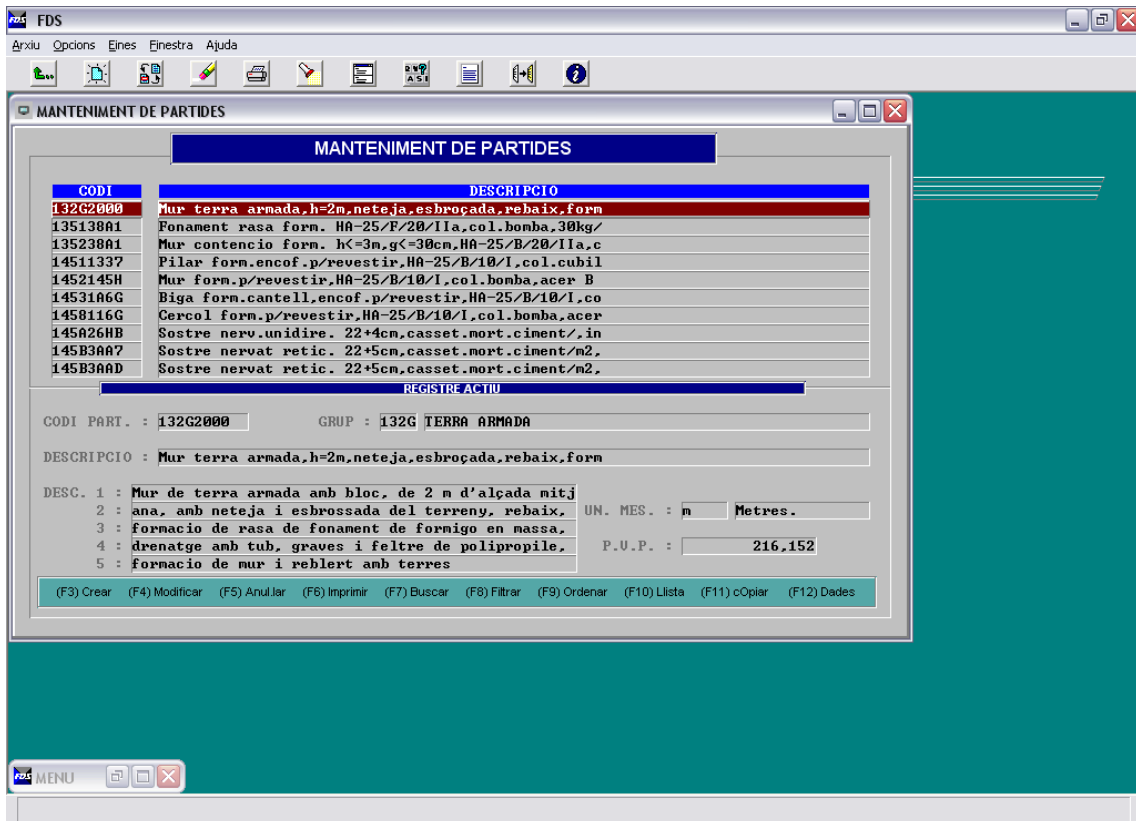
A continuació mostrem alguns exemples d'interfícies, exemplificant els criteris anteriors:



*Manteniment d'articles*

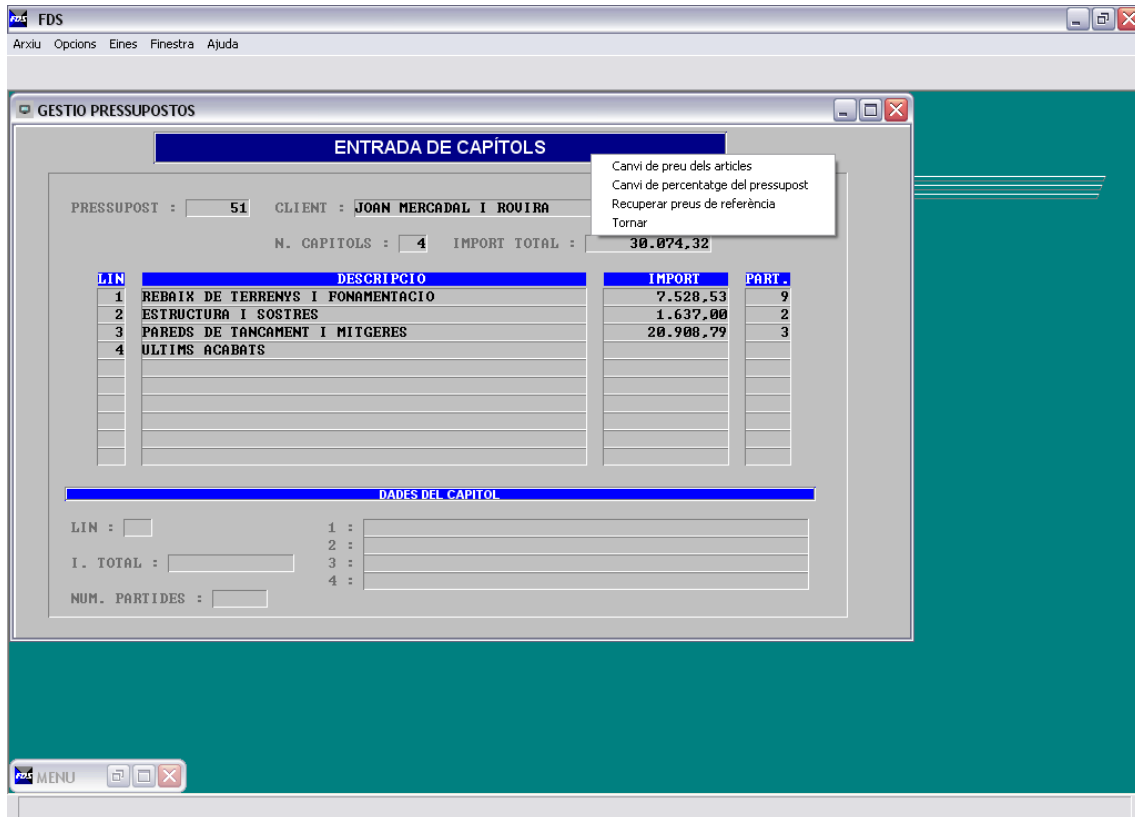


Manteniment de mesures

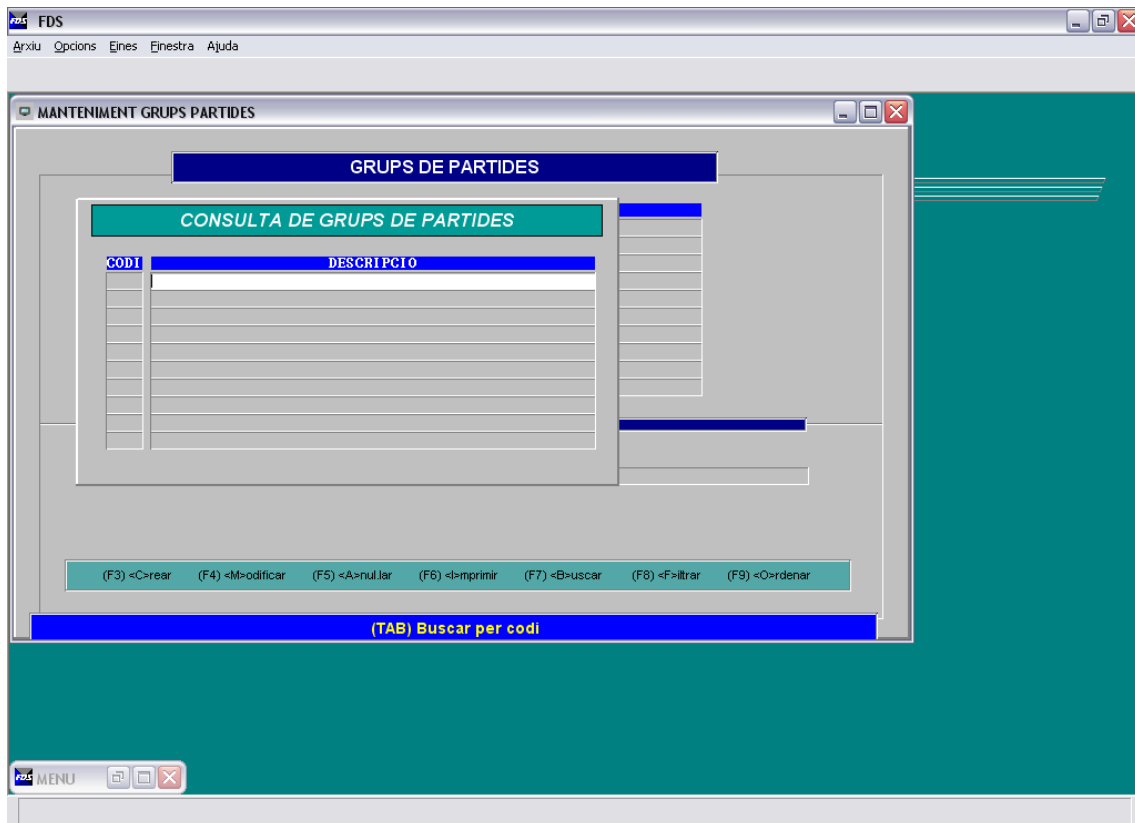


Manteniment de partides estandaritzades.

En aquests tres casos de manteniments podem veure com se segueixen els mateixos patrons de conducta a l'hora de dissenyar la interfície, tal com els botons, títols i posició i estructura dels camps.

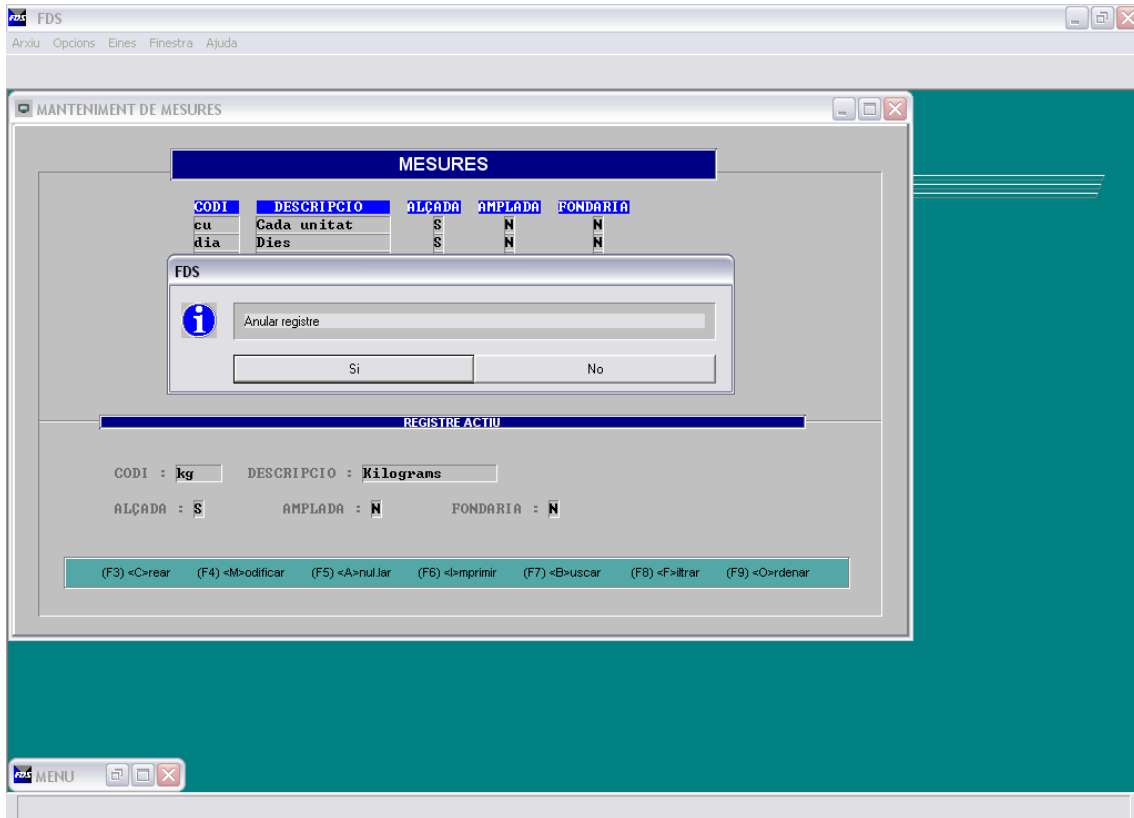


*Diferents opcions d'actualització de preus dels articles*



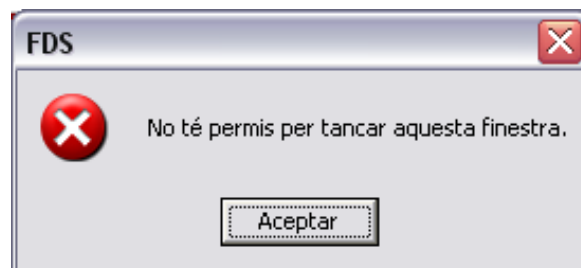
*Consulta de grups de partides*

En aquests dos casos podem veure com l'aplicació només permet seleccionar les opcions que ens interessin. A més a més, en el cas de la consulta de grups de partides utilitzem la mínima memòria possible, ja que no carreguem els grups d'immediat sinó que esperem que l'usuari ens entri una descripció per la qual començar a buscar.



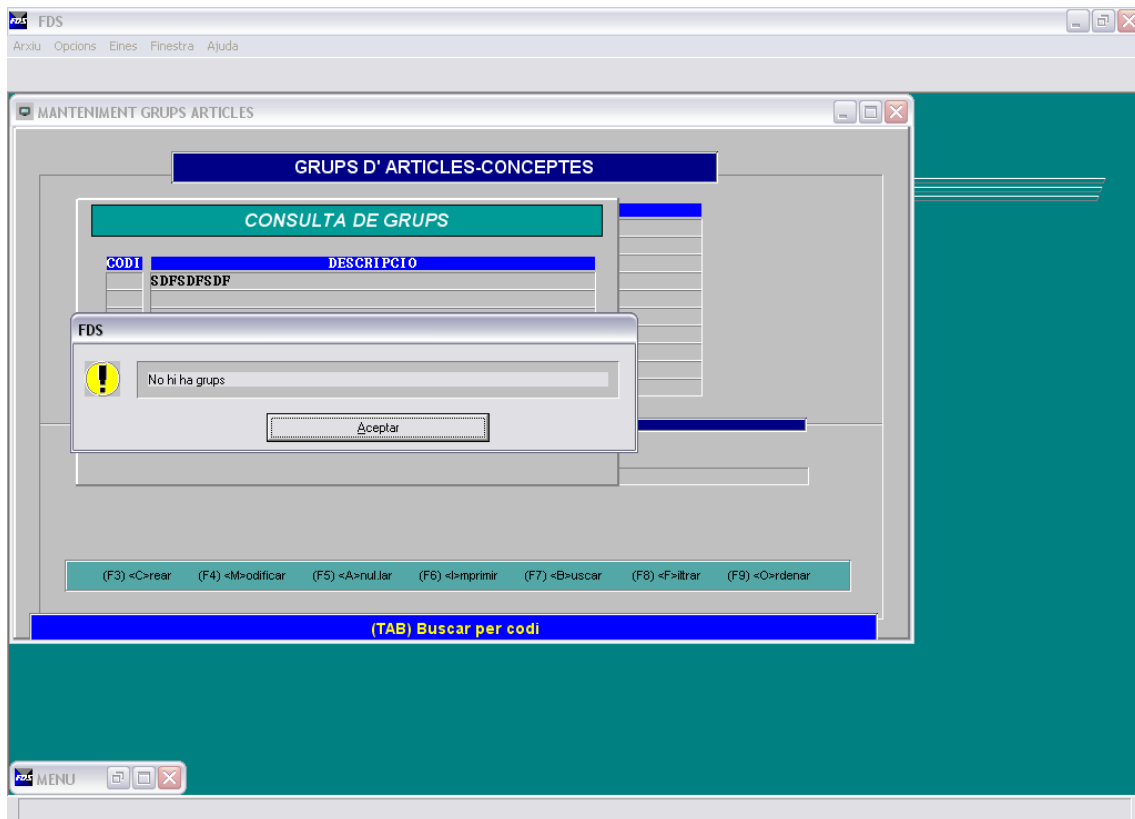
*Confirmació de l'eliminació d'una mesura.*

Podem veure com existeixen confirmacions en les accions destructives.



*Error mostrat per impossibilitat de tancar la finestra en curs*

Es mostra aquest error quan l'usuari intenta tancar una finestra quan no li està permès i podria afectar en l'execució del programa.



*Error que mostra l'aplicació en el manteniment de grups d'articles*

En aquest cas mostrem un error quan en la consulta de grups d'articles no s'ha trobat cap coincidència amb els caràcters entrats.

### 3.2. Disseny de fitxers

Treballarem amb fitxers ja que el mateix entorn està optimitzat per fer-ho i ja incorpora un sistema gestor de fitxers que en permet la gestió fàcil i còmode.

- Traducció del model E/R a fitxers:

Un cop obtingut el model conceptual (representat en el diagrama E/R) l'hem de transformar en un model lògic. Els passos que hem de seguir per aplicar aquesta transformació són els següents:

1. Cada entitat es transforma en un fitxer i els atributs de l'esmentada entitat en atributs del fitxer.
2. Les relacions molts a molts es transformen en fitxers la clau del qual estarà formada per la clau primària de les entitats relacionades.
3. En el cas de la relacions 1 a molts s'incorpora la clau principal de l'entitat principal a la clau principal de l'entitat secundària

Seguint aquests criteris hem creat els corresponents fitxers segons el model E/R (descrits a la pàgina 42).



- Índexs:

A més a més de crear els índex principals per codi de tots els fitxers per a la seva utilització, hem creat una sèrie d'índexs per facilitar-ne l'accés en accions de l'aplicació.

Per exemple, hem creat un índex *alfabètic* en els fitxers que precisen de consultes alfabètiques, tal i com hem vist en una interfície anterior. Per l'usuari és molt més còmode entrar part de la descripció de, per exemple, un grup d'articles que no pas el seu corresponent codi a l'hora de buscar-lo.

Aquests índexs creats tenen l'estructura [ *descripció\_ppal,codi* ]

Cal comentar també que en el cas dels fitxers relatius al pressupost, cada vegada que ens endinsem un nivell per sota (recordant l'estructura Pressupost → Capítol → Partida → Article) els corresponents índexs dels fitxers per nivell contenen el codi corresponent del fitxer i el codi dels fitxers de nivell superior.

- Altres fitxers:

A part dels ja mencionats fitxers anteriorment, hem creat uns altres fitxers, amb informació duplicada. Aquests fitxers són:

- Partides per pressupost: Fitxer on hi guardem el pressupost al qual correspon, el capítol i la informació relativa a la partida corresponent.
- Articles per partida dins d'un pressupost: En aquest fitxer hi emmagatzemem l'article amb les seves característiques, el pressupost del qual es tracta, el capítol al qual pertany i la partida d'aquest article.

Aquests fitxers inicialment s'omplen amb la informació provinent dels estàndards, però en qualsevol moment es pot variar degut a, per exemple, la dificultat d'accés a la ubicació de l'obra; llavors el transport dels materials incrementarà. D'aquesta manera els articles, partides estàndards no pateixen cap modificació.

També hem creat un fitxer de paràmetres amb la següent estructura:

**Paràmetres:**

*Registre*: numèric

*Últim pressupost*: numèric

*Empresa*: string

*Localitat de la firma*: string

*Logotip*: string

*Filigrana*: string

*Text Portada*: string

*Text Contraportada*: string

*Comptador de partides manuals*: numèric

Aquest fitxer de paràmetres s'ha creat per establir les dades de l'empresa propietària del programa, establir els textos de la portada i la contraportada estàndards en la impressió dels pressupostos en el cas que no se n'especifiquin de concrets en el mateix pressupost i un comptador de partides entrades per l'usuari ja que per defecte l'aplicació porta integrades les partides estandarditzades de l'ITEC, com s'ha comentat anteriorment.

En el cas del logotip i de la filigrana, són els noms de les imatges que es volen que surtin en la impressió. Si mai l'empresa canvia el logotip només s'haurà de tocar aquest paràmetre perquè surti imprès.

La filigrana és un dibuix en marca d'aigua que surt de fons en la impressió per tal de fer encara més visual la impressió.

Això ens permet implantar el programa en diferents clients i només establint les dades d'aquest fitxer podem començar a utilitzar-lo a ple rendiment.

A continuació descrivim el contingut de cada fitxer:

**NOM FITXER:** @Earticles

**Camps:**

ar\_cod: alfabètic  
ar\_des: alfabètic  
ar\_mes: alfabètic  
ar\_pvp: numèric  
ar\_pst: numèric  
ar\_gru: alfabètic  
ar\_datrev: data

**ÍNDEX PRINCIPAL:** i\_codi (ar\_cod)

**ÍNDEXS ALTERNATIUS:** i\_alfa (ar\_des,ar\_cod)  
i\_grup (ar\_gru,ar\_cod)

**NOM FITXER:** @Egruppar

**Camps:**

gp\_cod: alfabètic  
gp\_des: alfabètic

**ÍNDEX PRINCIPAL:** index (gp\_cod)

**ÍNDEX ALTERNATIU:** i\_alfa (gp\_des,gp\_cod)

**NOM FITXER:** @Egrups

**Camps:**

gr\_cod: alfabètic  
gr\_des: alfabètic

**ÍNDEX PRINCIPAL:** i\_codi (gr\_cod)

**ÍNDEX ALTERNATIU:** i\_alfa (gr\_des,gr\_cod)

**NOM FITXER:** @Emasures

**Camps:**

me\_cod: alfabètic  
me\_des: alfabètic  
me\_H: alfabètic  
me\_A: alfabètic  
me\_F: alfabètic

**ÍNDIX PRINCIPAL:** i\_codi (gr\_cod)

**NOM FITXER:** @Eparametres

**Camps:**

pa\_reg: numèric  
pa\_1stobra: numèric  
pa\_empresa: alfabètic  
pa\_diesrev: numèric  
pa\_localitat: alfabètic  
pa\_logotip: alfabètic  
pa\_filigrana: alfabètic  
pa\_texpor: alfabètic  
pa\_texcon: alfabètic  
pa\_fitprov: alfabètic  
pa\_1stalb: numèric  
pa\_perman: numèric

**ÍNDIX PRINCIPAL:** i\_codi (pa\_reg)

**NOM FITXER:** @Epartidac

**Camps:**

pc\_cod: alfabètic  
pc\_des: alfabètic  
pc\_des1: alfabètic  
pc\_des2: alfabètic  
pc\_des3: alfabètic  
pc\_des4: alfabètic  
pc\_des5: alfabètic  
pc\_mesu: alfabètic  
pc\_preu: numèric  
pc\_grup: alfabètic

**ÍNDIX PRINCIPAL:** i\_codi (pc\_cod)

**ÍNDIXS ALTERNATIUS:** i\_alfa (pc\_des,pc\_cod)  
i\_grup (pc\_grup,pc\_cod)

**NOM FITXER:** @Epartidal

**Camps:**

pl\_cod: alfabètic  
pl\_lin: numèric  
pl\_art: alfabètic  
pl\_qua: numèric

**ÍNDIX PRINCIPAL:** i\_codi (pl\_cod,pl\_lin)

**ÍNDIXS ALTERNATIUS:** i\_art (pl\_art,pl\_cod,pl\_lin)

**NOM FITXER:** @Epresart

**Camps:**

pr\_obra: numèric  
pr\_capitol: numèric  
pr\_linpart: numèric  
pl\_lin: numèric  
pr\_art: alfabètic  
pr\_qua: numèric  
pr\_preu: numèric  
pr\_import: numèric

**ÍNDIX PRINCIPAL:** i\_codi (pr\_obra,pr\_capitol,pr\_linpart,pr\_lin)

**ÍNDIX ALTERNATIU:** i\_art (pr\_art,pr\_obra,pr\_capitol,pr\_linpart,pr\_lin)

**NOM FITXER:** @Eprescap

**Camps:**

ca\_obra: numèric  
ca\_lin: numèric  
ca\_desc1: alfabètic  
ca\_desc2: alfabètic  
ca\_desc3: alfabètic  
ca\_desc4: alfabètic  
ca\_imp: numèric  
ca\_par: numèric

**ÍNDIX PRINCIPAL:** i\_obra (ca\_obra,ca\_lin)

**NOM FITXER:** @Epresdad

**Camps:**

da\_cod: numèric  
da\_data: data  
da\_cli: alfabètic  
da\_dir: alfabètic  
da\_cp: alfabètic  
da\_pob: alfabètic  
da\_pro: alfabètic  
da\_tel: alfabètic  
da\_desc1: alfabètic  
da\_desc2: alfabètic  
da\_desc3: alfabètic  
da\_desc4: alfabètic  
da\_desc5: alfabètic  
da\_import: numèric  
da\_capitols: numèric  
da\_certif: numèric  
da\_texpot: alfabètic  
da\_texcon: alfabètic  
da\_historic: alfabètic

**ÍNDIX PRINCIPAL:** i\_codi (da\_cod)

**ÍNDIXS ALTERNATIUS:** i\_cli (da\_cli,da\_cod)

i\_hist (da\_historic,da\_cod)

**NOM FITXER:** @Eprespar

**Camps:**

pa\_obra: numèric  
pa\_capitol: numèric  
pa\_lin: numèric  
pa\_part: alfabètic  
pa\_des1: alfabètic  
pa\_des2: alfabètic  
pa\_des3: alfabètic  
pa\_des4: alfabètic  
pa\_des5: alfabètic  
pa\_preu: numèric  
pa\_H: numèric  
pa\_A: numèric  
pa\_F: numèric  
pa\_mesu: alfabètic  
pa\_cant: numèric  
pa\_import: numèric

**ÍNDEX PRINCIPAL:** i\_codi (pa\_obra,pa\_capitol,pa\_lin)

**NOM FITXER:** @Eprovedor

**Camps:**

pr\_codi: numèric  
pr\_rao: alfabètic  
pr\_dni: alfabètic  
pr\_dir1: alfabètic  
pr\_pob1: alfabètic  
pr\_pro1: alfabètic  
pr\_cop1: alfabètic  
pr\_pais1: alfabètic  
pr\_tel1: alfabètic  
pr\_tel2: alfabètic  
pr\_tel3: alfabètic  
pr\_tel4: alfabètic  
pr\_ccl: alfabètic  
pr\_pcn: alfabètic  
pr\_ctc: alfabètic  
pr\_fp: numèric  
pr\_rec: alfabètic  
pr\_dp1: numèric  
pr\_dp2: numèric  
pr\_dtp: numèric  
pr\_seg: alfabètic  
pr\_obs: alfabètic  
pr\_ucom: data  
pr\_data1: data  
pr\_data2: data  
pr\_ga: alfabètic  
pr\_data: data  
pr\_lock: alfabètic  
pr\_cpa: alfabètic  
pr\_ipf: numèric

pr\_iva: numèric  
pr\_ba: numèric  
pr\_idioma: numèric  
**ÍNDIX PRINCIPAL:** prov\_codi (pr\_codi)  
**ÍNDIX ALTERNATIU:** alfa (pr\_rao,pr\_codi)

**NOM FITXER:** @Etmpparti  
**Camps:**

tp\_cod: alfabètic  
tp\_lin: numèric  
tp\_art: alfabètic  
tp\_qua: numèric  
**ÍNDIX PRINCIPAL:** i\_codi (tp\_cod,tp\_lin)  
**ÍNDIX ALTERNATIU:** i\_art (tp\_art,tp\_cod,tp\_lin)

### 3.3. Disseny de programes:

La tecnologia utilitzada per implementar l'aplicació és el llenguatge FDS. Aquest llenguatge ja porta integrat un entorn per a la programació (FDS Developer) i un *Runtime* (FDS Windows) per l'execució dels programes. Tanmateix, també incorpora un gestor de fitxers des del qual podem definir els fitxers, els seus respectius índexs, exportar-los i importar-los (molt útil a l'hora de realitzar actualitzacions de l'aplicació).

A part dels fitxers que nosaltres podem definir en l'aplicació, l'entorn porta integrats una sèrie de fitxers interns, com són, per exemple, la definició dels índexs i els fitxers, un fitxer per pas de paràmetres entre aplicacions (que més endavant comentarem i que el farem servir), un fitxer de definicions d'empreses (que també comentarem), fitxer d'usuaris de l'aplicació amb els seus privilegis...  
Aquests fitxers sempre tenen el mateix prototipus de nom: *fds\_NomDelFitxer*.

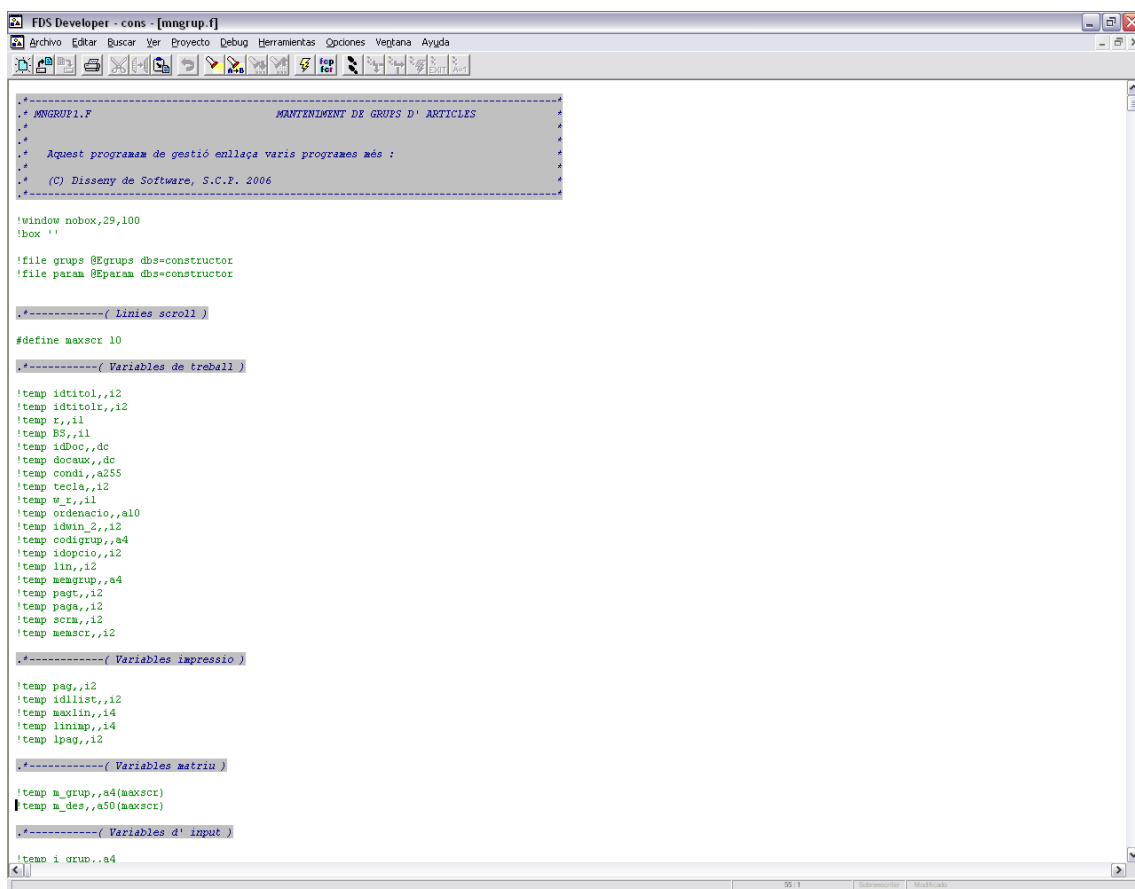
Una altra característica que té aquest llenguatge a l'hora de definir els fitxers és que podem posar "@E" i/o "@Y" a davant del nom del fitxer que definim. Si posem @E, li indiquem que aquell fitxer estarà disponible només per l'empresa en curs que estiguem. I si hi posem @Y indiquem que aquell fitxer pertany a un any en concret. Internament, l'FDS guarda aquests fitxers en carpetes diferents, el què fa és substituir els valors @E i @Y per un directori que li hem definit.  
Val a dir que en la nostra aplicació hem utilitzat l'@E ja que ens és útil per si mai un constructor vol tenir més d'una empresa per fer les promocions, cosa força freqüent en les promotores immobiliàries.

A part d'aquests fitxers, hi ha un fitxer *config.dbs* que es troba a la carpeta *Config* del directori d'instal·lació de l'FDS i a on s'hi indiquen les rutes dels directoris de dades, programes compilats (\*.prn i \*.pnt) i programes font (\*.f i \*.r). Aquest fitxer pot contenir la ruta directament de la ubicació dels fitxers o bé pot contenir l'adreça IP del servidor on estan allotjats. En aquest cas, en el servidor sí que haurem de tenir un fitxer *config.dbs* on indiqui la ruta local dels directoris abans esmentats.

Cal comentar també que aquesta plataforma està pensada per treball multi-usuari i que les comunicacions TCP/IP entre els clients i el servidor les gestiona l'FDServer, un gestor de connexions propi de l'entorn i que funciona a través del port 7500 TCP. Per tant, si no ens interessa tenir recursos compartits en el servidor i que gestioni tot el trànsit de dades l'FDServer aquesta és la millor manera.

Com a sistemes operatius per l'execució del software per part dels usuaris podem utilitzar tots els sistemes operatius de Microsoft des del Windows 98 fins al Vista, al igual que de servidor (en el cas que es tractin de molt pocs ordinadors). Per a varis clients és àmpliament recomanada la utilització d'un servidor dedicat que treballi amb un sistema operatiu UNIX o LINUX, encara que també podríem tenir un Windows 2000 Server o superior. Com a sistema Unix es recomana SCO OpenServer i com a Linux podem utilitzar Red Hat, Fedora Core i Suse Linux. Òbviament el rendiment amb un sistema Linux és superior al d'un Microsoft, ja que està optimitzat per al seu ús.

Es mostra a continuació, mitjançant impresos de pantalla, l'entorn de treball del llenguatge FDS.



```
* MNGRUP1.F                                MANTENIMENT DE GRUPS D' ARTICLES
*
*
* Aquest program de gestió enllaça varis programes més :
*
* (C) Disseny de Software, S.C.F. 2006
*
!window nobox,29,100
!box ''
!file groups @Egrups dbs=constructor
!file param @Eparam dbs=constructor

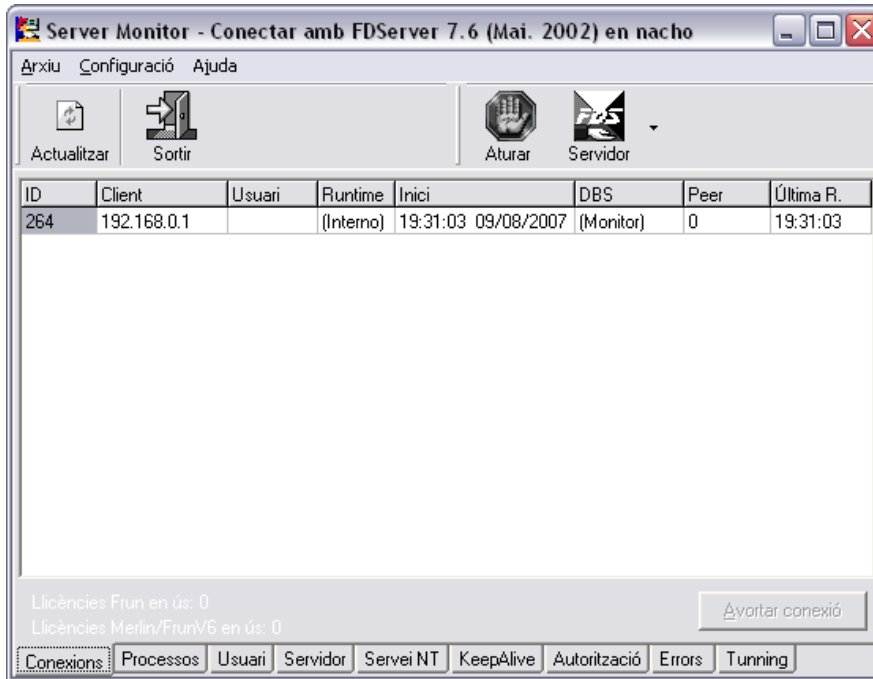
#-----( Linies scroll )
#define maxscr 10
#-----( Variables de treball )
!temp idtitol,,12
!temp idstitol,,12
!temp r,,11
!temp ES,,11
!temp idDoc,,dc
!temp docAux,,dc
!temp cond1,,s255
!temp tecla,,12
!temp w_r,,11
!temp ordenacio,,a10
!temp idwin_2,,12
!temp codigrup,,a4
!temp idopcio,,12
!temp lin,,12
!temp mnggrup,,a4
!temp pagt,,12
!temp paga,,12
!temp scrm,,12
!temp menact,,12

#-----( Variables impressio )
!temp pag,,12
!temp idllist,,12
!temp maxlin,,14
!temp linimp,,14
!temp lpag,,12

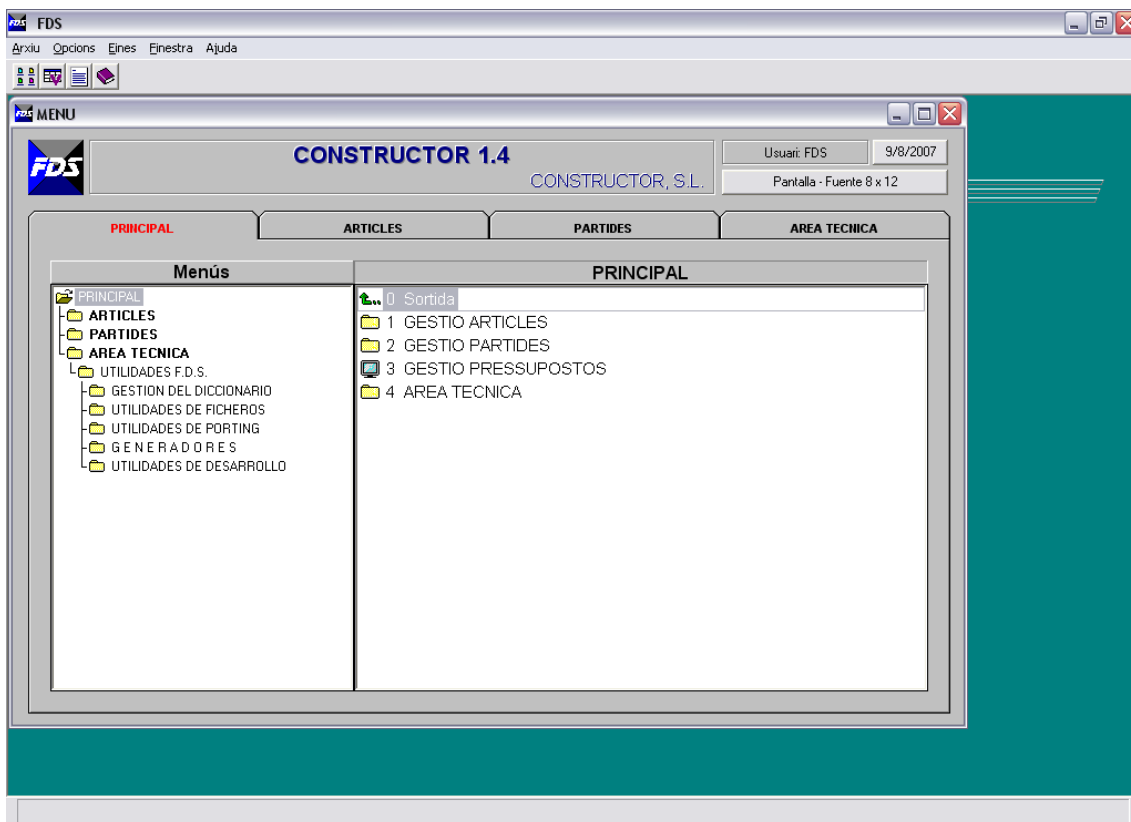
#-----( Variables matrix )
!temp m_grup,,a4(maxscr)
!temp m_des,,a50(maxscr)

#-----( Variables d' input )
!temp i gruw..a4
```

Aquí podem veure una mostra de l'FDS Developer, amb la declaració de la finestra, els fitxers i les variables de treball.

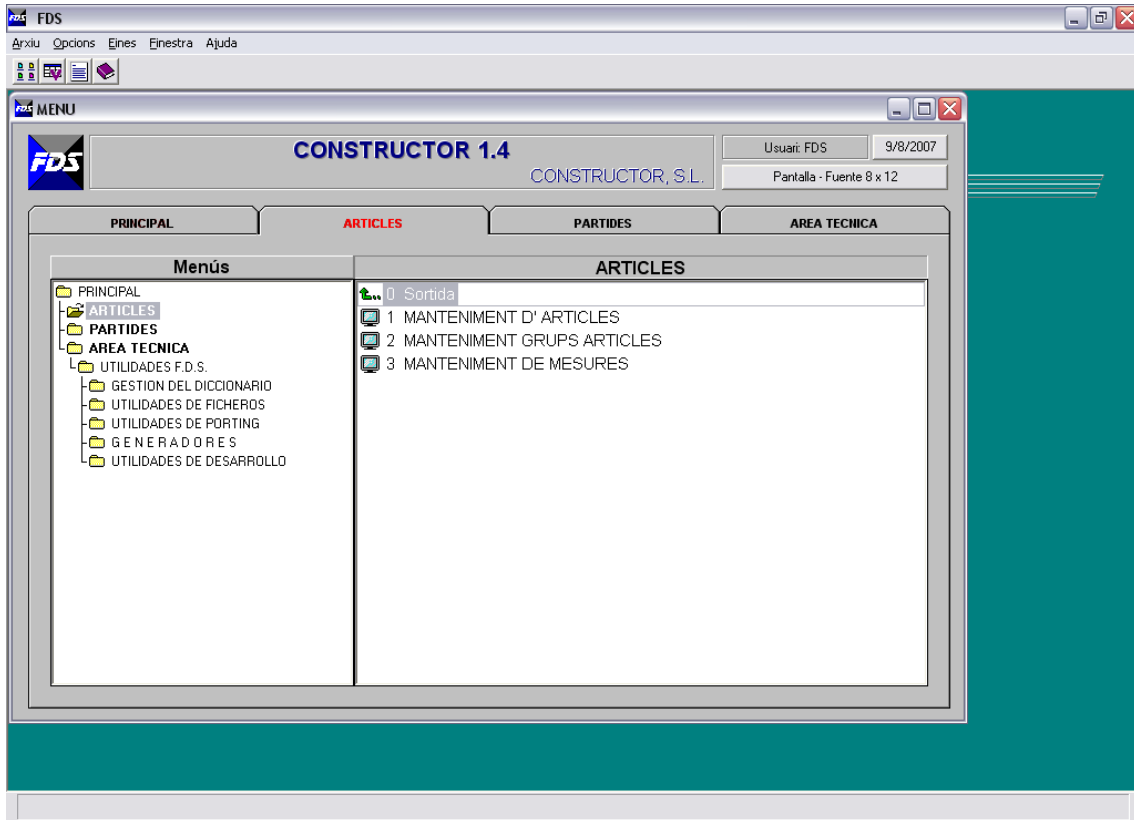


En aquesta figura tenim l'FDS Server Monitor, que és el monitor d'estat de les connexions de l'FDS Server del client amb el servidor. Des d'aquí podem engegar-lo i parar-lo, d'aquesta manera tallaríem tots els processos en curs del client.

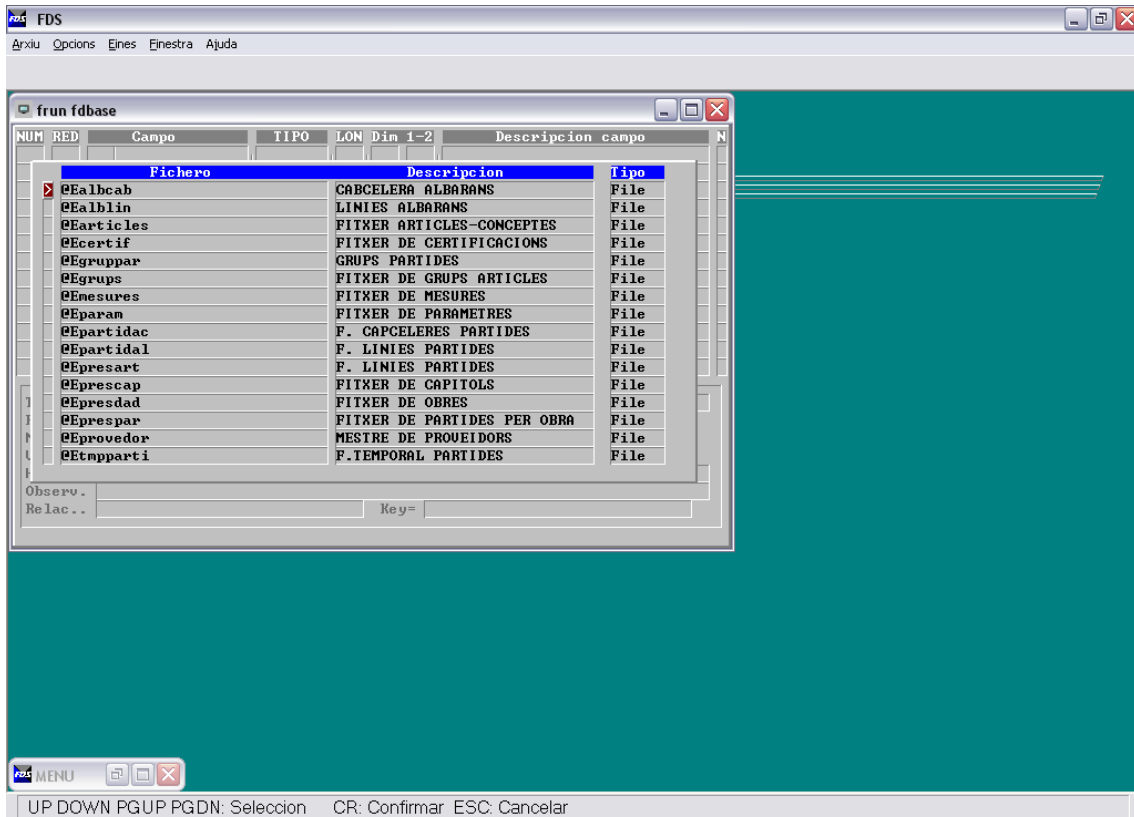


Finestra principal del programa, on podem accedir a totes les opcions.

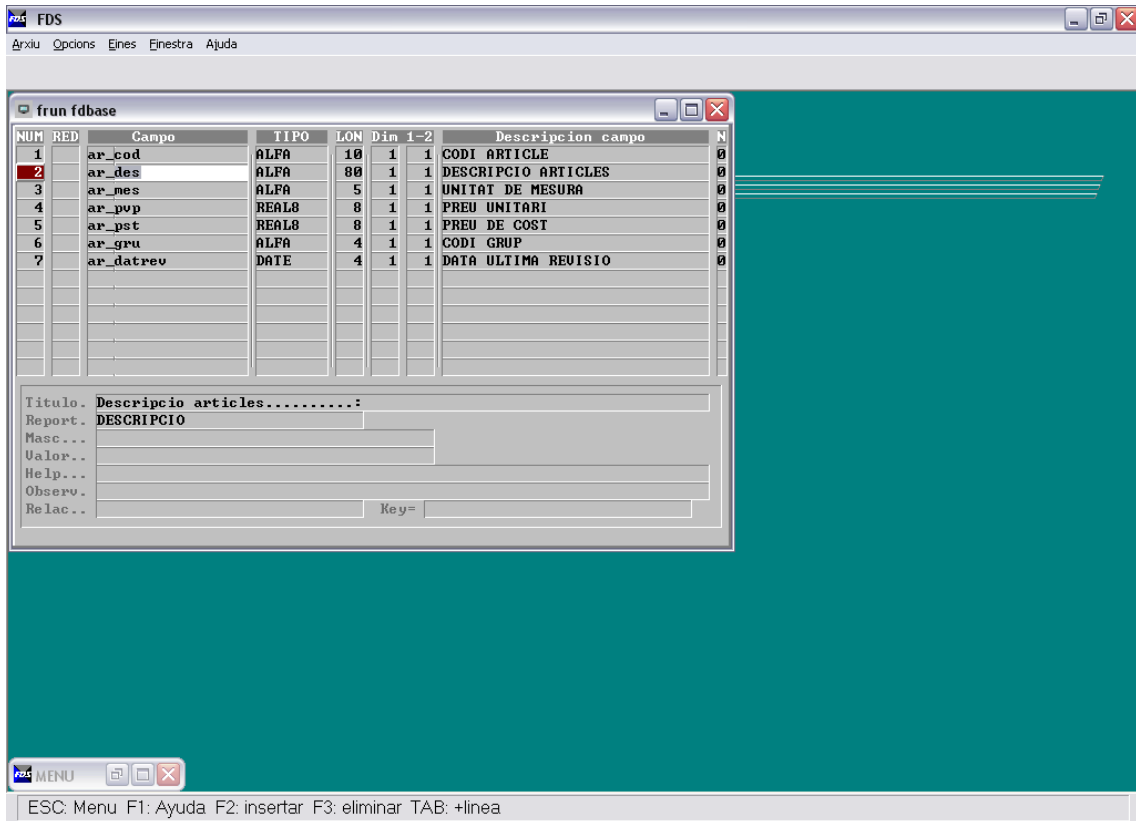




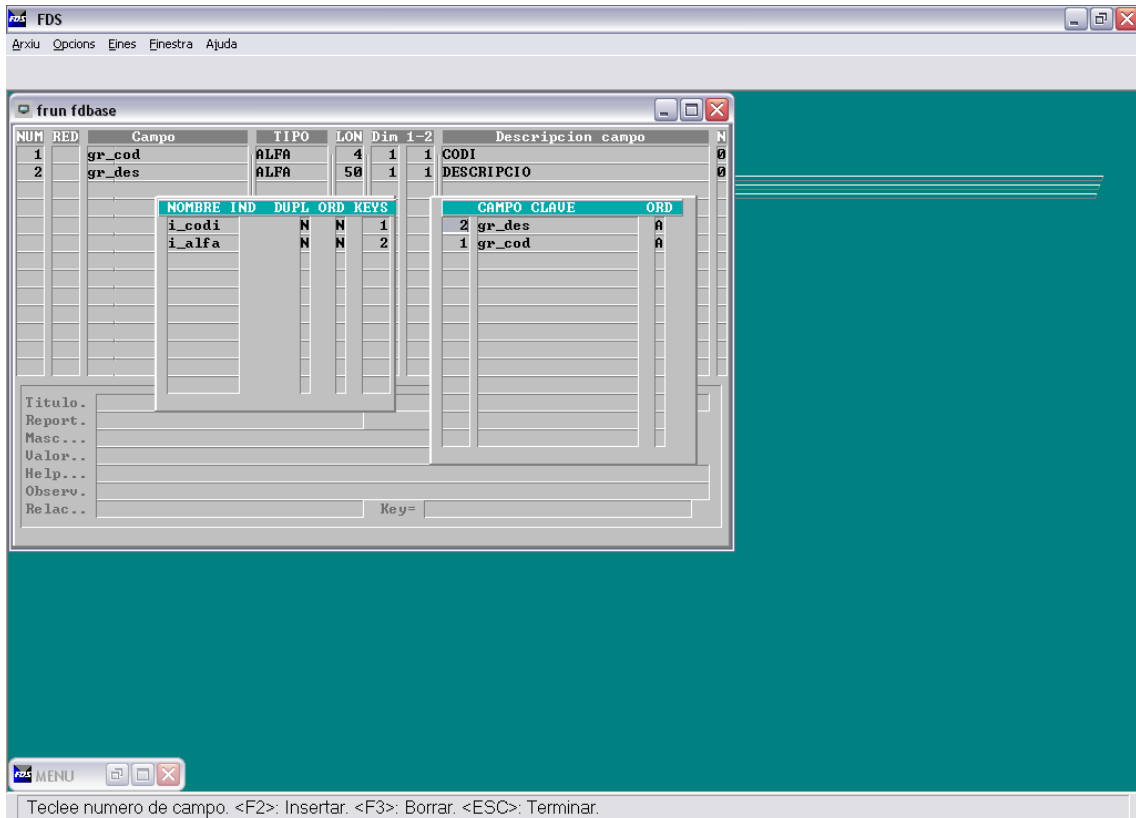
Gestió d'articles amb els seus corresponents manteniments.



Gestor de fitxers amb els corresponents fitxers de l'aplicació.



Conjunt de camps del fitxer d'articles.



Definició d'índexs del fitxer de grups d'articles. Com podem observar, l'índex alfabètic conté com a claus la descripció i el codi; tal i com hem comentat abans això ho hem realitzat per agilitzar les cerques per descripció.

- Característiques de programació:

A nivell de definició de dades, el llenguatge FDS permet la definició dels següents tipus de dades:

- **Alfabètic o string:** de 0 a 255 caràcters.
- **Enter d'1 byte:** Nombre enter entre 0 i 255.
- **Enter de 2 bytes:** Nombre enter entre -32797 i +32768.
- **Enter de 4 bytes:** Nombre enter entre  $-2^{10}$  i  $+2^{10}$ .
- **Data:** Enter de 4 bytes que conté els segons passats des de l'01/01/70.
- **Real de 4 bytes:** Real de simple precisió.
- **Real de 8 bytes:** Real de doble precisió.
- **Serial:** Comptador intern d'insercions.
- **Money:** Real de doble precisió. Utilitzat per la moneda.
- **Blob:** Dades binàries com ara imatges.

Ara anem a veure com declarem i accedim als fitxers i a les variables des de codi.

La declaració dels fitxers es fa de la següent manera:

```
!file NomFitxerLogic NomFitxerFisic [dbs=NomAplicació]
```

Com a exemple:

```
!file articles @Earticles dbs=constructor
```

La sentència *dbs=constructor* fa referència al nom de l'aplicació sobre la qual està el fitxer. Si no s'especifica pren per defecte l'aplicació en curs.

Aquesta definició és útil, per exemple, quan tenim una facturació i una comptabilitat i hem de comptabilitzar les factures, llavors podem accedir als fitxers de la comptabilitat des d'un procés de traspàs de la facturació.

En quan a l'accés de les dades del fitxer tenim una sèrie d'instruccions:

- **clearbuf** NomFitxer: Esborra el buffer de variables del fitxer.
- **access** NomFitxer **by** index: Es prepara per accedir al fitxer per l'índex que li hem dit. Si no usem aquesta sentència el programa accedirà al fitxer pel primer índex que haguem declarat en el gestor de fitxers.
- **read** NomFitxer: Llegeix un registre en concret del fitxer.
- **insert** NomFitxer: Insereix un nou registre al fitxer.
- **write** NomFitxer: Modifiquem un registre del fitxer.
- **delete** NomFitxer: Esborrar un registre del fitxer, previ posicionament en ell.
- **start** NomFitxer: Posiciona el punter just abans del primer registre que compleix les condicions que li hem dit. Usat per a inici de bucles.
- **next** NomFitxer: Llegeix el següent registre segons l'índex que li hem dit.
- **prev** NomFitxer: Llegeix el registre anterior segons l'índex que li hem dit.

Les instruccions **read**, **start**, **next** i **prev** poden anar precedides d'un "\_" ( \_read, \_start...). Si posem aquest caràcter al davant estem indicant que no bloquegem el registre, per tant, sempre que podem ho utilitzem. En els únics casos en els quals hem de bloquejar obligatòriament són en el cas d'escriure (**write**) i en el cas d'esborrar (**delete**).

A més a més tenim variables de retorn de certes instruccions que ens indiquen l'estat en què hem quedat després de l'execució de la instrucció. A continuació mostrem les diferents clàusules que podem tenir en funció de la instrucció:

- En el cas de la instrucció **read**:

<u>Define</u>	<u>Clàusula</u>	<u>Valor</u>	<u>Descripció</u>
_NOERROR		0	Registre llegit
_NSR	nsr=	1	Registre no existeix
_RIU	riu=	3	Registre ocupat

En el cas de les instruccions **prev** i **next** les clàusules són les mateixes.

L'ús és el següent:

```
_read NomFitxer nsr=TRAP
```

El TRAP pot ser una etiqueta a anar si succeix l'esdeveniment o bé poden ser una sèrie de clàusules que desvien el flux del programa predefinides.

Aquestes clàusules són:

- **NEXT**: Es desvia el flux cap a la següent línia del programa.
- **BREAK**: Bifurca el flux del programa a la línia que segueix al done del mateix nivell. En el cas d'estar en un bucle, en finalitza l'execució del mateix.
- **CONTINUE**: Bifurca el flux del programa al done del mateix nivell. En el cas d'estar en un bucle, en continua l'execució des de l'inici del bucle, obviant les línies posteriors.

Exemple:

```
clearbuf grups
gr_cod=codigrup
read grups nsr=NEXT
if _NSR do
    r=1
else
    r=0
endif
return
```

- En el cas de la instrucció **insert**:

<u>Define</u>	<u>Clàusula</u>	<u>Valor</u>	<u>Descripció</u>
_NOERROR		0	Operació exitosa
_RE	re=	2	Registre ja existeix

Exemple:

```
clearbuf grups
gr_cod=i_grup
gr_des=i_des
insert grups re=NEXT
if _RE do
    error "Registre ja existeix !"
    return
endif
```

- En el cas de la instrucció **write**:

<u>Define</u>	<u>Clàusula</u>	<u>Valor</u>	<u>Descripció</u>
_NOERROR		0	Registre existosa
_RE	re=	2	Registre ja existeix
_NRS	nrs=	4	Registre no seleccionat

Exemple:

```
gr_des=i_des
write grups nrs=NEXT
if _NRS do
    error "Registre no seleccionat, No s'actualitza !"
    gosub DISPLAYDOCU
    scroll memscr
    return
endif
```

Un altre dels processos importants que es realitza en l'aplicació és el recàlcul i actualització de preu del pressupost. Aquest procés és potser el més important de l'aplicació.

La mecànica de funcionament és la següent:

- El programa rep per paràmetre el número de pressupost.
- Llegeix recursivament per cada capítol les seves partides i per cada partida tots els seus articles.
- Un cop arribat al nivell més baix, l'article, en recalcula el preu, multiplicant la quantitat pel preu i n'actualitza l'import.
- Llavors va retornant nivell per nivell realitzant l'actualització fins a tornar al nivell del pressupost i així successivament.

Aquest procés es pot executar manualment des de l'entrada de pressupostos si es desitja però també es fa internament entre les diferents entrades de capítols, partides i articles per assegurar la correctesa de l'import total.

Aquí tenim un tros del programa de càlcul:

```

.*----- ( Recalcula el preu de l' obra )-----*
RECALC_PREOBR
  C_impob=0
  C_ncap=0
  access C.prescap by "i_obra"
  clearbuf C.prescap
  C.ca_obra=C_obra
  C.ca_lin=1
  _start C.prescap
  while ( )
    _next C.prescap nsr=NEXT
    if _NSR do
      break
    endif
    if C.ca_obra<>C_obra do
      break
    endif
    if C_mode="AR" or C_mode="NR" do
      C_cap=C.ca_lin
      gosub RECALC_PRECAP
    endif
    C_impob=C_impob+C.ca_imp
    C_ncap=C_ncap+1
  done
  C_impob=DIS.ROUND(C_impob,2)
  if C_mode="AR" or C_mode="AN" do
    C.da_cod=C_obra
    read C.presdad nsr=NEXT
    if _NSR do
      error " No es pot actualitzar preu pressupost
"+str(C_obra)
      return
    endif
    C.da_import=C_impob
    C.da_capitols=C_ncap
    write C.presdad
  endif
  unlock C.prescap
  unlock C.presdad
  return

```

En aquest exemple hem pogut veure més exemples d'accés als fitxers, així com un bucle per accedir a tots els registres del fitxer.

Pel què fa al tema de declaració de variables, l'estructura és la següent:

```
!temp camp, [títol], tybytes[(dimensió)] [,màscara] [;llista_de_valors]
```

La descripció dels camps és la següent:

**camp**

Nom de la variable amb una longitud màxima de 18 caràcters.

**títol**

Títol automàtic de la variable que apareix si és posicionat en pantalla amb la declaració +camp.

**tybytes**

Tipus i tamany de la variable.

**dimensió**

Nombre d'elements quan es defineixen vectors o matrius. S'admeten dues dimensions separades per comes.

**màscara**

Format d'edició del camp.

**llista\_de\_valors**

Valors permesos per les variables que són automàticament verificats en un input.

- Altres exemples:

**Exemple 1:**

En aquest exemple podem veure un seguit d'instruccions que ens permeten la impressió en formularis pre-dissenyats amb un editor del propi entorn de desenvolupament, l'FDSForm.

```
idllist=FCreate("relartic","forms",sal,FF_MULTIPAG)
```

Creem l'identificador del llistat sobre el qual "inserirem" el conjunt de dades que volem que surtin impreses.

```
FData(idllist,"DESC",ar_des,lpag)
```

Inserim al formulari el valor del camp *ar\_des* en el camp *DESC* del formulari en la posició *lpag*. En aquest cas hem d'especificar la posició del camp ja que en el formulari es tracta d'una taula.

```
lpag=lpag+1
if lpag>41 do
    FNewpage(idllist)
    lpag=1
    gosub HEAD
endif
```

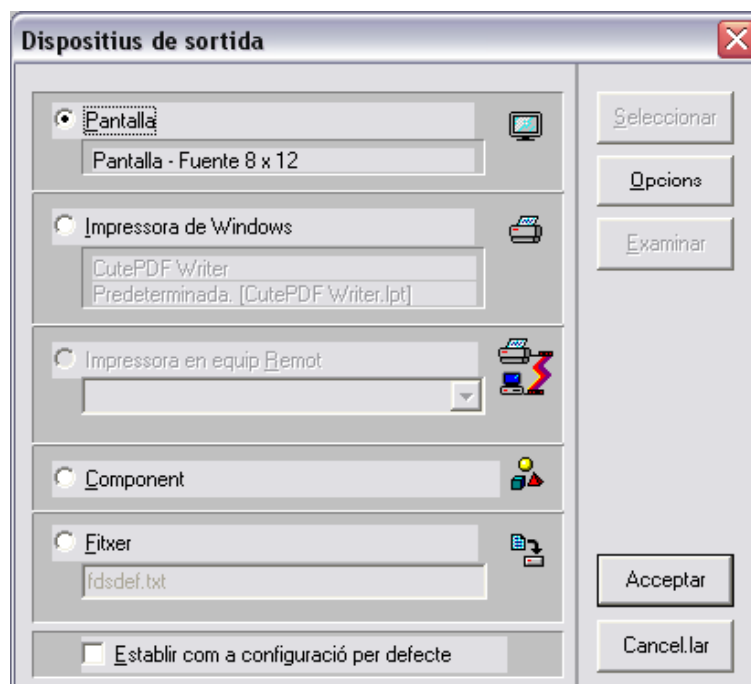
Tros de codi que augmenta la posició a inserir del camp i comprova que no sigui la última. En tal cas es crea una pàgina nova, es posa el comptador de línies a 0 i s'insereixen les dades de la capçalera: títols, data, hora...

```
FRelease(idllist)
```

Un cop finalitzat el llistat, amb l'execució d'aquesta instrucció alliberem el llistat del programa i passa a enviar-se a la impressora seleccionada.

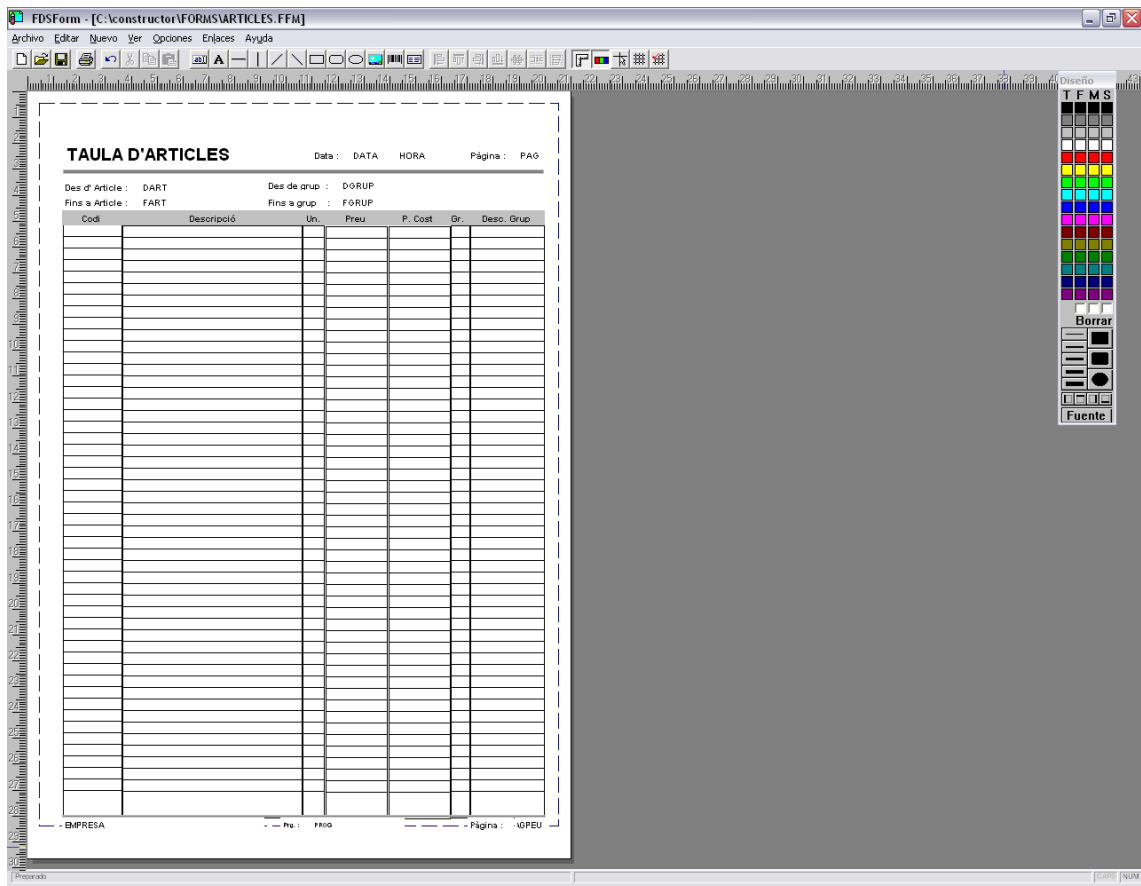
```
sal=getenv("FDS_LPT")
SelPrinter()
switch (GetPrinter())
  case 1:
    sal=""
    break
  case 2:
    sal="LPTWIN"
    break
  case 5:
    sal="NULL"
    break
  default:
    sal="PRINTER"
    break
done
```

En aquí el programa llegeix la impressora que té seleccionada actualment i mostra una finestra de selecció de sortida, per a posteriori recollir-ne el valor.



Finestra de selecció de sortida del llistat.





Vista de l'editor de formularis amb el formulari del llistat d'articles.

Pàgina: 1 de 87 Imprimir

---

**TAULA D'ARTICLES** Data : 10-08-07 10:46 Pàgina : 1

Des de Article : Des de grup :  
 Rn s a Article : Rn s a grup :

Codi	De scripció	Un.	Preu	P. Co st	Cr.	De so. Grup
A%4A0001	Dispositiu aïllant ma d'obra	m	1.500000	0.000000		
A%4A000100	Mòdul aïllant ma d'obra	m	4.000000	0.000000		
A0121000	Oficial 1a	h	19.800000	0.000000		OFICIALS 1A
A0122000	Oficial 1a colista	h	20.000000	0.000000		OFICIALS 1A
A0123000	Oficial 1a encofrador	h	19.800000	0.000000		OFICIALS 1A
A0124000	Oficial 1a formolista	h	19.800000	0.000000		OFICIALS 1A
A0125000	Oficial 1a soldador	h	20.100000	0.000000		OFICIALS 1A
A0126000	Oficial 1a encasador	h	19.800000	0.000000		OFICIALS 1A
A0127000	Oficial 1a collocador	h	19.800000	0.000000		OFICIALS 1A
A0128000	Oficial 1a colista	h	19.800000	0.000000		OFICIALS 1A
A0129000	Oficial 1a quicoms	h	19.800000	0.000000		OFICIALS 1A
A012A000	Oficial 1a fuster	h	20.170000	0.000000		OFICIALS 1A
A012B000	Oficial 1a estucador	h	19.800000	0.000000		OFICIALS 1A
A012C000	Oficial 1a carter	h	19.800000	0.000000		OFICIALS 1A
A012E000	Oficial 1a vedner	h	19.270000	0.000000		OFICIALS 1A
A012F000	Oficial 1a manua	h	20.150000	0.000000		OFICIALS 1A
A012G000	Oficial 1a colador	h	20.400000	0.000000		OFICIALS 1A
A012H000	Oficial 1a electricista	h	20.400000	0.000000		OFICIALS 1A
A012J000	Oficial 1a limesta	h	20.400000	0.000000		OFICIALS 1A
A012L000	Oficial 1a llaner	h	20.400000	0.000000		OFICIALS 1A
A012M000	Oficial 1a muntador	h	20.400000	0.000000		OFICIALS 1A
A012N000	Oficial 1a sardner	h	19.800000	0.000000		OFICIALS 1A
A012P000	Oficial 1a marfesta	h	19.800000	0.000000		OFICIALS 1A
A0132000	Ajudant colista	h	17.870000	0.000000		AJUDANTS
A0133000	Ajudant encofrador	h	17.870000	0.000000		AJUDANTS
A0134000	Ajudant formolista	h	17.870000	0.000000		AJUDANTS
A0135000	Ajudant soldador	h	17.890000	0.000000		AJUDANTS
A0136000	Ajudant encasador	h	17.870000	0.000000		AJUDANTS
A0137000	Ajudant collocador	h	17.870000	0.000000		AJUDANTS
A013A000	Ajudant fuster	h	17.790000	0.000000		AJUDANTS
A013B000	Ajudant estucador	h	17.870000	0.000000		AJUDANTS
A013C000	Ajudant carter	h	17.870000	0.000000		AJUDANTS
A013E000	Ajudant vedner	h	17.490000	0.000000		AJUDANTS
A013F000	Ajudant manua	h	17.890000	0.000000		AJUDANTS
A013G000	Ajudant colador	h	17.990000	0.000000		AJUDANTS
A013H000	Ajudant electricista	h	17.990000	0.000000		AJUDANTS
A013J000	Ajudant limesta	h	17.990000	0.000000		AJUDANTS
A013L000	Ajudant muntador	h	17.870000	0.000000		AJUDANTS
A013P000	Ajudant sardner	h	17.870000	0.000000		AJUDANTS
A0140000	Mandats	m <sup>2</sup>	16.870000	0.000000		MANDATS
A0140000	Mandats quicoms	m	16.870000	0.000000		MANDATS
A0150000	Mandats electricista	m	17.150000	0.000000		MANDATS ESPE
A0160000	Piso	m	17.070000	0.000000		PECES
A016P000	Piso sardner	m	17.070000	0.000000		PECES
B0111000	Arba	m	0.940000	0.000000		NEUTRES
B0117100	Dissolvent desmantox. colista PVC	l	3.160000	0.000000		DISSOLVENTS
B0311070	Soma cadra calc. a'horms.	m	14.900000	0.000000		SOMES
B0312070	Soma cadra quart. a'horms.	m	15.950000	0.000000		SOMES
B0312020	Soma cadra quart. a'mortari	m	16.370000	0.000000		SOMES
B0312080	Soma cadra quart. 0-5 mm	m	16.870000	0.000000		SOMES

Pg: 1 articles Pàgina : 1

Aquí podem veure una mostra del llistat que hem executat mostrat per pantalla.

## Exemple 2:

En aquest exemple mostrarem la definició del contingut visual de les pantalles: títols, icones i definició de les finestres.

Cal destacar que totes aquestes definicions es realitzen des de codi.

```
!window nobox,29,100
!box ''
```

Definició de la pantalla i la seva alçada i la seva amplada.

```
drawbox(3,3,24,94,"") /* Requadre principal
line(17,3,96,"H","O")
idtitol=GCreateText(2,18,2,62,GW_LINE3D,GW_CENTER+GW_VCENTER)
GSetFont(idtitol,"Arial",20,GW_BOLD)
GSetText(idtitol,"MESURES")
GSetTextColor(idtitol,GW_WHITE)
GSetBackColor(idtitol,0,0,135)
```

Definició del requadre principal, una línia horitzontal i opaca i un títol amb tipus de lletra *Arial* amb les seves corresponents mides i colors.

A més a més, a diferència dels llenguatges visuals com podrien ser el C++ o el Delphi en les respectives últimes versions, les graelles que mostren els registres s'han de gestionar manualment les operacions sobre elles, com ara el desplaçament unitari de cada element, l'avançar i el retrocedir pàgina.

Per a poder realitzar aquestes operacions, es va decidir utilitzar un suport del llenguatge FDS anomenat *document*. Un document és una taula en memòria que es pot omplir amb qualsevol valor i/o camp. D'aquesta manera, al document hi tenim carregades les dades que s'han de mostrar a la graella i també tenim un apuntador que ens indica quin és l'element que en aquells moments tenim seleccionat.

En quan a la programació, quan entrem a dins de la graella capturem la tecla que es prem i en funció d'això es realitza una acció o una altra: avançar una línia, una pàgina, accedir a les opcions de modificar, esborrar, crear...

Aquí podem veure un exemple:

```
CONS
gosub ICONS2
scroll 1
while (1)
  gosub CONS_DISPFIG
  tecla=keycode(2)
  gosub CONS_DISPLAY
  if tecla=chr(32) do
    tecla=-#DOWN
  endif
  if tecla=-#BS do
    tecla=-#UP
  endif
  if tecla=-#UP do
    if scrline>1 do
      scroll -1
```



```
        w_r=10
        break

    case -#F4:
    case -#RETURN:
    case asc("M"):
    case asc("m"):
        gosub MODIFICAR
        w_r=10
        break

    case -#F5:
    case -#DEL:
    case asc("A"):
    case asc("a"):
        gosub ANULAR
        w_r=10
        break

    case -#F6:
    case asc("I"):
    case asc("i"):
        gosub IMPRIMIR
        w_r=10
        break

    case -#F7:
    case asc("V"):
    case asc("v"):
        gosub BUSCAR
        w_r=10
        break

    case -#F8:
    case asc("F"):
    case asc("f"):
        gosub FILTRAR
        scroll 1
        w_r=10
        break

    case -#F9:
    case asc("O"):
    case asc("o"):
        gosub ORDENAR
        w_r=10
        break

    default:
        bell
        w_r=10
        break

done
if w_r=10 do
    continue
endif
if w_r<>0 do
    break
endif
done
```

```
gosub ICONSOFF
return
```

Aquesta és la rutina principal de captura de la tecla i l'execució de les instruccions. Podem veure com en cada cas de la tecla premuda es crida a una rutina que executa la funció seleccionada.

```
CONS_DISPFIG
  highlight m_mesures-m_F
  gosub FIT_I_VAR
  display i_mesures-i_F
  return
```

```
CONS_DISPLAY
  display m_mesures-m_F
  return
```

```
CONS_PGDN
  if paga=pagt do
    return
  else
    paga=paga+1
  endif
  gosub CONS_CLEAR
  gosub CONS_SCRM
  gosub CONS_UPMAT
  gosub CONS_DISP
  scroll 1
  return
```

```
CONS_PGUP
  if paga=1 do
    return
  else
    paga=paga-1
  endif
  gosub CONS_CLEAR
  gosub CONS_SCRM
  gosub CONS_UPMAT
  gosub CONS_DISP
  scroll maxscr
  return
```

```
CONS_SCRM
  scrm=maxscr
  if paga=1 and (DcRows(idDoc)<maxscr) then scrm=DcRows(idDoc)
  if paga=pagt do
    scrm=DcRows(idDoc)
    scrm=scrm-((pagt-1)*maxscr)
  endif
  return
```

```
CONS_UPMAT
  for (scroll 1; scripline<=scrm; scroll)
    DcSetRowPos(idDoc, (paga-1)*maxscr+scripline)
    gosub CONS_FITVAR
  done
  return
```

```
CONS_DISP
  for (scroll 1; scrline<=scrm; scroll)
    gosub CONS_DISPLAY
  done
  return

CONS_CLEAR
  clearmat m_mesures
  clearmat m_des
  clearmat m_H
  clearmat m_A
  clearmat m_F
  clscroll m_mesures
  clscroll m_des
  clscroll m_H
  clscroll m_A
  clscroll m_F
  return

.*----- ( Càrrega de document a variables pantalla )-----*

CONS_FITVAR
  m_mesures=idDoc#me_cod
  m_des=idDoc#me_des
  m_H=idDoc#me_H
  m_A=idDoc#me_A
  m_F=idDoc#me_F
  return
```

En les línies de codi superiors hi ha les diferents rutines d'avançar i retrocedir pàgina, lectura dels valors del document, display de variables i el *highlight* de la fila seleccionada.

Una altra opció que tenim és la d'ordenar la graella ascendentment o descendentment i després mostrar-la de nou. Aquesta operació també s'ha de dur a terme a través de codi i ho realitzem de la següent manera:

```

.*----- ( Ordenar ) -----*

ORDENAR
  gosub O_NEWDOC
  DcAddRow (O_doc,1,) : DcSetRowPos(O_doc,1) : O_doc#O_nom="Codi
Ascendent"
  DcAddRow (O_doc,2,) : DcSetRowPos(O_doc,2) : O_doc#O_nom="Codi
Descendent"
  DcAddRow (O_doc,3,) : DcSetRowPos(O_doc,3) :
O_doc#O_nom="Descripció Ascendent"
  DcAddRow (O_doc,4,) : DcSetRowPos(O_doc,4) :
O_doc#O_nom="Descripció Descendent"
  DcAddRow (O_doc,5,) : DcSetRowPos(O_doc,5) :
O_doc#O_nom="Tornar"
  O_x=80
  O_y=0
  gosub O_FLOAT
  switch(O_ordre)
    case 1 :
      ordenacio="gr_cod"
      DcSort(idDoc,"gr_cod")
      break
    case 2 :
      ordenacio="!gr_cod"
      DcSort(idDoc,"!gr_cod")
      break
    case 3 :
      DcSort(idDoc,"gr_des")
      ordenacio="gr_des"
      break
    case 4 :
      DcSort(idDoc,"!gr_des")
      ordenacio="!gr_des"
      break
    case 5:
      scroll 1
      return
      break
  done
  gosub DISPLAYDOCU
  scroll 1
  return

*----- ( Càrrega de les opcions del menu flotant ) -----*

!temp O_ordre,,a15
!temp O_doc,,dc
!temp O_r,,i2
!temp O_nomdoc,,a10
!temp O_x,,i1
!temp O_y,,i1

O_FLOAT
  if O_x=0 then O_x=1
  if O_y=0 then O_y=1

```



```

O_r=DcRows(O_doc)
if O_r<2 do
    error "No hi ha opcions per poder fer un menu "
    goto O_END
endif
createmenu("O_ORDRE")
for (scroll 1;scrline<=DcRows(O_doc);scroll)
    DcSetRowPos(O_doc,scrline)
    insertmenuitem("O_ORDRE",,scrline,O_doc#O_nom,)
done
O_ordre=execfloatmenu ("O_ORDRE",O_y,O_x,"v")
O_END
delmenu ("O_ORDRE")
DcDelete (O_doc)
return

.*----- ( Crea document ) -----*

O_NEWDOC
O_doc=DcNew("OPCIONES",,)
DcAddCol (O_doc, fdcSTRING, "O_nom", "ORDENAR",,)
O_r=DcCols(O_doc)
if O_r<1 do
    error "No s' han creat les columnes "
    goto O_END
endif
return

```

En aquest exemple podem veure com es crea un menú flotant amb les diferents opcions d'ordenació i segons l'opció seleccionada s'ordena la graella i es mostra.

Per últim, podem filtrar les dades de la graella segons els paràmetres que ens interressi i mostrar-la de nou. Per fer-ho realitzem les següents accions:

```

.*----- ( Filtrar les dades ) -----*

FILTRAR
openwin 2
gosub INPUTS
texmess="": gosub MESSAGE
closewin 2
if BS=1 do
    return
endif
condi="me_cod>=:dmesures and me_cod<=:fmesures"
docaux=FQLSelect("fql:@Emasures",,FQL_ALL,,,,condi,ordenacio)
if (DcRows(docaux)<1) do
    error "No hi ha registres per aquestes condicions"
    return
endif
idDoc=DcCopy(docaux)
gosub DISPLAYDOCU /* Display del document per pantalla
DcDelete(docaux)
return

```

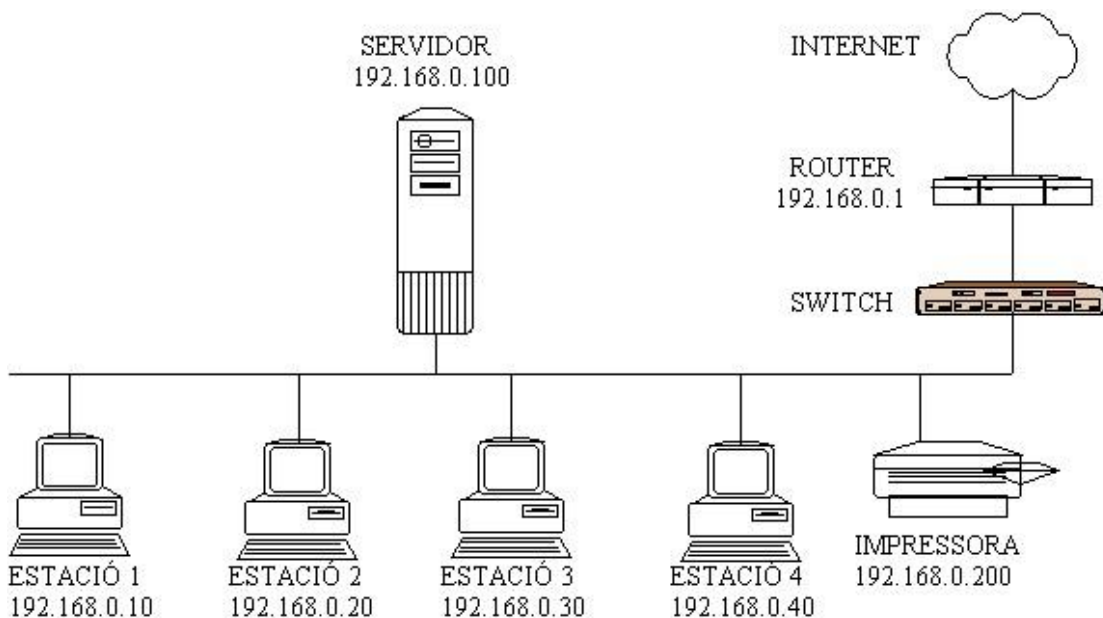
### 3.4. Disseny tecnològic

En aquest apartat veurem totes les necessitats tan a nivell hardware com software per tal de què l'aplicació realitzada tingui un òptim ús i explotació.

❖ Necessitats a nivell Hardware:

En aquest apartat farem una distinció entre les dues possibles solucions en la implementació: un entorn personal on tindrem un sol ordinador i una estructura client/servidor, on disposarem d'una xarxa d'àrea local (LAN) on hi tindrem un servidor, una sèrie d'ordinadors que es connectaran al servidor per tenir accés al programa, les impressores per treure els llistats, un router per l'accés a Internet i un switch per distribuir les connexions internes de la LAN.

A continuació mostrem un exemple gràfic de com podria ser la implantació a nivell hardware en el cas de disposar d'una estructura client/servidor.



En aquest esquema, el servidor allotja tota l'aplicació i les dades i en gestiona les connexions. Llavors cada estació disposa d'un software de connexió per accedir al programa del servidor.

Des de qualsevol ordinador es pot utilitzar la impressora de la xarxa i també es pot accedir a internet a través del router. Cal dir que el router ens el proporciona l'operadora amb la qual tenim contractada el servei d'internet.

En aquest cas el switch seria un que permetés connexions 10/100/1000 a tots els ports i tant els ordinadors com el servidor disposessin de targetes de xarxa Gigabit per a agilitzar al màxim les connexions i els traspassos de dades.

En el cas que tinguéssim una estructura d'entorn personal, només disposaríem d'un ordinador, sense necessitat de tenir una LAN ni un switch. I si a més a més tingués connexió a Internet necessitaríem també un router o mòdem per accedir-hi.

Pressupost d'un ordinador per fer de servidor:

**Ordinador Servidor Linux Suse**



Intel Server 2000 Xeon

- Xassis Intel SC5400BASE
- Procesador Intel Dual Core Xeon 5050 2x2 amb 3,00 Ghz bus a 667 Mhz.
- Placa base Intel S5000PSL
- Video integrat 16 Mb
- Memoria RAM de 2 GB DDR II 667 Mhz
- Disco dur SATA de 400 GB SATA-300 a 7200 rpm.
- RAID 0 ( 2 Discs en Mirror )
- Reproductor DVD x52



**IMPORT .....2.100,00 € (IVA inclòs)**

Ordinadors per les diferents estacions de treball:



### Ordinador Estació de Treball Windows XP

#### ASUS CORE 2 DUO (002)

- Potència (W):300 - Format: Torre
- Dimensions (Alt x Ample x Prof.): 38 Cm x 16 Cm x 35.5 Cm
- Processador (GHz): Pentium 2,8 GHz Socket :478 HT :Cache :512 MB
- Memòria:DDR II 1 GB,
- Disc Dur: 160 GB Interface SATA-300
- Targeta Video: Integrada Intel GMA 900
- Teclat i Mouse inalàmbric Logitech
- DVD Regrable doble cara Velocitat: x16
- Xarxa: Integrada 10/100
- 6 USB Frontal x 2 USB Posterior x 4
- Audio:Integrada 5.1
- Conectors: Sèriex1 Paralel·lelx1 Line-Inx1 Sortida Altaveusx1 Satax2 IEEE1394x0



**IMPORT .....807,00 € (IVA inclòs)**

Monitors:

- Monitor TFT 17" ACER AL1717 AS .....195,00 €



- Monitor TFT 17" ASUS MB17SE .....205,00 €



❖ Necessitats a nivell Software:

Per a la creació del software hem hagut d'utilitzar l'entorn de desenvolupament del llenguatge FDS que ja compta amb els *runtimes* d'execució per les estacions amb les quals es vol treballar.

Els diferents preus de les llicències de l'entorn FDS són els següents:

- Llicència de l'FDS ..... 456,00 € per usuari (IVA inclòs)

Llavors, com a sistemes operatius necessitem per a cada estació un Windows XP:

- WINDOWS XP PROFESSIONAL SP2 ..... 150,00 € (IVA inclòs)
- WINDOWS XP HOME SP2.....95,00 € (IVA inclòs)

També hauríem d'incloure com a cost el paquet ofimàtic de Microsoft:

- OFFICE STANDARD 2007 ..... 315,00 € (IVA inclòs)
- SOFTWARE DESENVOLUPAT <sup>1</sup>

En el cas del servidor, s'ha optat per l'ús de l'OpenSuse, de distribució lliure i gratuïta, per tal de minimitzar els costos a l'usuari final. Actualment es disposa de la versió 10.2.

- **Com repartim el software dins de l'arquitectura presentada del sistema informàtic?**

El software de l'aplicació es divideix en tres parts:

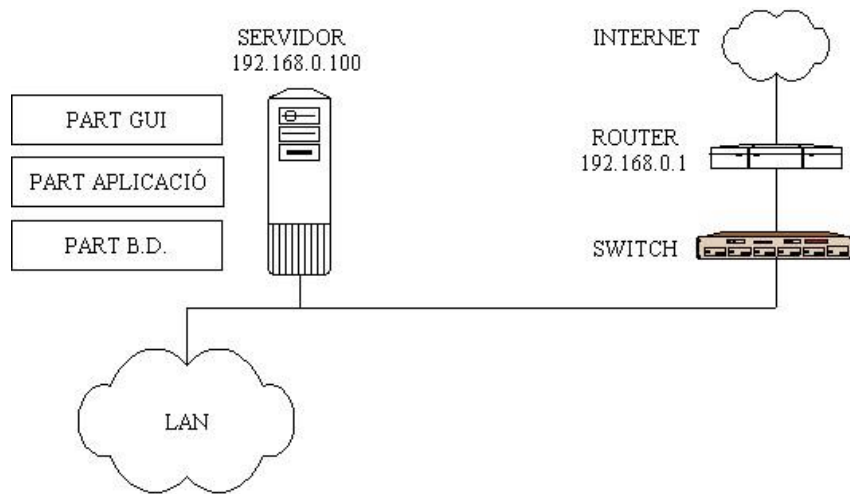
1. **Part GUI:** Software que desenvolupem per les interfícies gràfiques de comunicació amb l'usuari.
2. **Part aplicació:** Software que es desenvolupa per satisfer els requeriments de l'aplicació.
3. **Part Base de Dades:** En aquest cas es tracta del sistema gestor de fitxers associat a l'aplicació.

En el cas de la nostra aplicació tenim que les tres parts estan ubicades en el servidor i en el client només hi ha un intèrpret que és capaç d'entendre el format de les dades que li envia el servidor i presentar-ho. Aquesta arquitectura de distribució s'anomena *distribució de presentació*.

---

<sup>1</sup> Ara per ara és impossible calcular-ho ja que ens hem passat més temps aprenent i familiaritzant-nos amb l'entorn que desenvolupant.

Tot seguit mostrem un esquema de la d'aquesta arquitectura:



## 4. Millores i conclusions

### 4.1. Millores

Algunes de les millores que es poden que es poden realitzar sobre l'aplicació són les següents (que actualment ja demana l'empresa):

- Certificacions:

Incorporar un control de les certificacions realitzades per l'arquitecte, el constructor i el client sobre l'obra. Les certificacions consisteixen en l'acceptació de la realització d'una part del total de l'obra sobre el terreny on estan presents els membres abans esmentats i se certifica que s'han realitzat una sèrie de capítols. Aquest pas es realitza per tal d'anar liquidant econòmicament el total de l'obra per part del client.

- Albarans de control:

Afegir la possibilitat de mantenir un control de totes les hores i materials que s'han anat utilitzant en la realització d'una obra, per així mantenir un control sobre el què s'ha pressupostat i el què realment s'està realitzant. Es tractaria que al acabar el dia s'entres un albarà amb totes les despeses, hores realitzades i materials utilitzats per cada operari i cada obra i es poguessin contrastar amb les dades inicials del pressupost.

- Facturació:

Un cop acabada la realització d'una obra i d'haver quadrat tots els números i comprovar que tot és correcte, tenir la possibilitat de traspasar el pressupost a un albarà d'una facturació estàndard que ja té el client per així agilitzar la feina.

Un cop conegut l'entorn no és massa complicat de realitzar, simplement es tracta de dissenyar els fitxers necessaris i desenvolupar el codi per dur a terme aquestes funcions. Per tant, simplement necessitem temps.



## 4.2. Conclusions

Després de concloure la realització d'aquest projecte en podem fer una valoració realment positiva. En primer lloc destacaria que es tracta d'un projecte real, amb tot el què això implica. No es tracta d'una pràctica més com podria ser una de les realitzades durant la carrera, que es presenta al professorat per a la seva avaluació sinó un projecte que s'ha hagut d'aplicar en una empresa. Això implica que he hagut de posar-me en la pell de l'usuari final que utilitzarà l'aplicació fent que el seu ús sigui el màxim d'intuïtiu i senzill; tot adequant-ne la programació.

La implementació de l'aplicació ha estat força llarga. Tot aquest procés també inclou uns coneixements previs sobre l'argot i la manera de treballar dels constructors i dels diferents tècnics que intervenen en una obra: arquitectes, paletes, electricistes, llauers...

Després d'assolir aquests coneixements i de veure la manera de treballar *in situ* dels diferents professionals, s'ha realitzat l'aplicació i s'ha posat en marxa tenint en compte els requeriments inicials del client. Cal destacar que tot aquest procés ha tingut una durada aproximada d'uns 12 mesos.

Una de les coses que potser m'ha costat més ha estat l'aprenentatge i l'ús del llenguatge FDS, tot i que tampoc ha estat un gran *handicap*.

Com a pràctica real, aquest projecte m'ha aportat molts coneixements que fins ara desconeixia sobre el món de la construcció i m'he familiaritzat amb l'argot que utilitzen i amb la seva manera de treballar. D'altra banda, també m'ha servit per aprendre a posar-me en la situació de l'usuari que utilitzarà el programa, per tal de facilitar-li la feina i controlar els possibles errors que pot cometre i poder-los tractar.

He de dir que aquest programa actualment s'està emprant en una empresa real de la comarca d'Osona amb uns resultats molt satisfactoris segons els objectius previstos.

## 5. Bibliografia

Per a la realització d'aquest projecte s'ha utilitzat l'ajuda que proporciona l'entorn FDS, on hi ha la definició i explicació de les instruccions.

- PRESSMANN. *Enginyeria del Software*. McGraw Hill
- *Apunts Introducció als Fitxers i a les Bases de Dades (IFBD)*. Escola Politècnica Superior (EPS). Universitat de Vic.
- *Apunts Enginyeria del Software de Gestió (ESG)*. Escola Politècnica Superior (EPS). Universitat de Vic.
- *Apunts Metodologia de Projectes (MP)*. Escola Politècnica Superior (EPS). Universitat de Vic.