

Disseny d'un sistema de log genèric per a la implementació de sistemes operacionals.

Lidia Grau Bolufer
E.T. Informàtica gestió

Consultor Alexandre Cornet
18 de Juny del 2004

Resum.

El present treball de fi de carrera, com descriu el seu títol, consisteix a desenvolupar un sistema genèric de seguiment de l'execució d'un sistema operacional. Caldrà, doncs, implementar un component que, associat a les accions d'usuari, enregistri les instruccions que executa i els valors de determinades variables d'interès. Un cop finalitzat, passarà a formar part d'un entorn de programació i alliberarà, a futurs desenvolupadors, de les tasques de construir mecanismes de notificació d'errors, de realimentació d'errors al servei d'atenció a l'usuari, d'auditoria d'utilització o, fins i tot, de depuració de codi.

Com tot projecte, complirà amb el requisit de disposar d'un pla que permeti fer un seguiment dels terminis d'execució, de les fites establertes i un control dels entregables identificats. Pel fet de tractar-se del desenvolupament d'un sistema informàtic, contemplarà les etapes d'especificació de requisits, anàlisi, disseny, codificació, proves unitàries i proves funcionals, generant els informes pertinents que serveixin de documentació i de referència a les etapes posteriors.

Des d'un punt de vista tecnològic, permetrà aprofundir en el coneixement de la estructura de funcionament del PL/SQL d'Oracle (crides a procediments i, especialment, el tractament d'excepcions), en la utilització de JDBC com a mecanisme de comunicació entre JAVA i Oracle, en l'ús de les classes de generació d'interfícies gràfiques d'usuari (swing) i, finalment, permetrà posar en pràctica funcionalitats d'Oracle que no havia tingut la oportunitat d'emprar, com ara, tipus genèrics de dades, objectes persistents o transaccions autònomes.

Índex de contingut

Resum.....	2
1. Introducció.....	5
1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC....	5
1.2 Objectius del TFC.....	5
1.3 Enfocament i mètode seguit.....	6
1.4 Planificació del projecte.....	6
1.5 Productes obtinguts.....	8
1.6 Altres capítols de la memòria.....	9
2. Especificació de requeriments.....	10
2.1 Funcions del sistema de log.....	10
2.1.1 <i>Funcions de seguiment</i>	10
2.1.2 <i>Funcions estadístiques</i>	11
2.1.3 <i>Funcions de configuració</i>	11
2.2 Altres entregables del projecte.....	11
3. Anàlisi.....	12
3.1 Model de dades.....	12
3.1.1 <i>Diagrama d'E/R</i>	12
3.1.2 <i>Definició d'entitats</i>	12
3.1.3 <i>Definició d'atributs</i>	13
3.2 Diagrames de flux de dades.....	14
3.3 Disseny extern : disseny d'interfícies.....	16
3.4 Definició de proves funcionals.....	20
4. Disseny.....	21
4.1 Model de dades. Ús d'ANYDATA Type i NESTED TABLES.....	21
4.2 Procediments necessaris.....	23
4.2.1 <i>El package UTL_REG_LOG. Ús de transaccions autònomes i sobrecàrrega de funcions</i>	23
4.2.2 <i>El package UTL_VIS_LOG</i>	27
4.2.3 <i>El procediment UTL_DEL_LOG</i>	28
4.3 Aplicatiu d'explotació de les dades de LOG.....	28
4.3.1 <i>Classes d'entitat</i>	28
4.3.2 <i>Classes frontera</i>	32
4.3.3 <i>Classes de control</i>	33
4.3.4 <i>Paquets d'agrupació de classes</i>	35
4.4 Altres possibilitats a explorar. Ús de ALL_ARGUMENTS i DBMS_DESCRIBE.....	36
4.5 Definició de proves unitàries.....	36
4.5.1 <i>UTL_REG_LOG</i>	36
4.5.2 <i>UTL_VIS_LOG</i>	38
4.5.3 <i>UTL_DEL_LOG</i>	39
4.5.4 <i>VisorLog</i>	39
5. Manual d'usuari del sistema de log genèric.....	41
5.1 Instal·lació i desinstal·lació.....	41
5.2 Normativa de codificació del sistema de seguiment.....	41
5.3 Manual d'usuari del visor.....	47
5.3.1 <i>Opció seguiment</i>	48
5.3.2 <i>Opció estadístiques</i>	53
5.3.3 <i>Opció configuració</i>	55
5.3.4 <i>Opció eliminar</i>	57
5.3.5 <i>Opció ajuda</i>	57
5.3.6 <i>Opció sortir</i>	57
6. Conclusions.....	58
Glossari.....	59
Bibliografia.....	60
Annexos.....	61
Annex 1. Joc de proves disponibles.....	61
Annex 2. Joc de dades disponibles.....	61

Índex de figures

1. Diagrama de Gantt	7
2. Diagrama de PERT	7
3. Taula de productes obtinguts	9
4. Diagrama d'entitats i relacions	12
5. Taula de descripció d'entitats	12
6. Taula de descripció dels atributs de l'entitat CONFIGURACIÓ	13
7. Taula de descripció dels atributs de l'entitat TRANSACCIO	13
8. Taula de descripció dels atributs de l'entitat TRAÇA	13
9. Taula de descripció dels atributs de l'entitat BÀSICA	13
10. Taula de descripció dels atributs de l'entitat BLOC	14
11. Taula de descripció dels atributs de l'entitat OBSERVACIO	14
12. Diagrama de context	14
13. DFD a nivell 1	15
14. DFD a nivell 2 del procés 2	15
15. Diagrama de col·laboració de la configuració de seguiment	16
16. Diagrama de col·laboració de la consulta de seguiment	16
17. Diagrama de col·laboració d'estadístiques de seguiment	16
18. Disseny del menú del visor de seguiment	16
19. Disseny de la pantalla de filtre de transaccions	17
20. Disseny de la pantalla amb les dades de l'arbre d'execució	17
21. Disseny de la pantalla d'observacions	18
22. Disseny de la pantalla d'estadístiques	18
23. Disseny de la pantalla de configuració	19
24. Disseny dels missatges d'avís o error	19
25. Taula de definició de proves funcionals	20
26. Taula de definició dels atributs del objecte TP_OBSERVACIO	22
27. Taula de definició dels atributs de la taula TRAÇA	22
28. Taula de definició dels atributs de la taula CONFIGURACIO	23
29. Estructura de paquets	35
30. Taula de definició de proves unitàries del UTL_REG_LOG	37
31. Taula de definició de proves unitàries del UTL_VIS_LOG	39
32. Taula de definició de proves unitàries del UTL_DEL_LOG	39
33. Taula de definició de proves unitàries del VisorLog	40
34. Taula d'atributs emmagatzemats per tipus de registre	45
35. Pantalla d'identificació	47
36. Menú del VisorLog	47
37. Pantalla de seguiment (filtre transaccions)	48
38. Pantalla de seguiment (exemple filtre transaccions)	49
39. Missatge d'avís en l'obtenció de transaccions	49
40. Missatge d'error en l'obtenció de transaccions	50
41. Pantalla de seguiment (arbre execució)	50
42. Pantalla de seguiment (detall d'un bloc)	51
43. Pantalla de seguiment (només arbre)	51
44. Pantalla de seguiment (només detall bloc)	52
45. Pantalla de seguiment (observacions)	52
46. Pantalla d'estadístiques	53
47. Pantalla d'estadístiques (exemple filtre)	54
48. Taula amb les franges horàries definides en el sistema	54
49. Pantalla d'estadístiques (exemple estadística)	55
50. Missatge d'avís obtenció estadística	55
51. Pantalla de configuració	56
52. Missatge d'avís del procés d'esborrat	57
53. Pantalla d'ajuda	57

1. Introducció.

1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.

Si, de sempre, el desenvolupament de programari ha tractat de reaprofitar codi, actualment és una necessitat. Disposar d'un marc de programació, d'un conjunt d'eines que implementen les funcionalitats més freqüents incrementa la productivitat de qualsevol equip i redunda en benefici de l'objectiu principal, proporcionar solucions informàtiques en un termini de temps cada cop més breu.

En aquest sentit, una de les primeres problemàtiques que es planteja quan es vol desenvolupar un sistema informàtic és la de notificar els errors a l'usuari i fer-ho de forma exhaustiva i entenedora per tal de cerciorar-nos de que, arribat el cas, l'operador del sistema serà capaç de descriure'l correctament. No hi ha res més descoratjador que un usuari dient 'No funciona' i que a la pregunta 'però, Què és el que no funciona?' la resposta sigui 'No ho sé'. Així doncs, el sistema genèric de seguiment ve a solucionar aquest problema, proporcionant les eines per, no només identificar el problema, sinó esbrinar per què ha arribat a produir-se.

Però el sistema genèric de seguiment pot ser utilitzat fora de l'entorn d'explotació. Nosaltres, com a codificadors, sabem molt bé que no sempre, per no dir mai, s'encerta a la primera, que cal depurar el codi i que cal contemplar totes les possibles casuístiques. Aquesta, en ocasions, no és una feina simple i una eina de seguiment de l'arbre de execució d'un procés i de visualització dels valors de les variables en punts determinats del codi pot ser de gran ajuda.

Així doncs, no s'ha de parlar de punt de partida del projecte, més aviat a l'inrevés, aquest treball és el punt de partida per la construcció d'un entorn de programació que ha de permetre el desenvolupament de altres sistemes.

1.2 Objectius del TFC.

Objectius genèrics

L'objectiu principal d'aquest projecte és mostrar l'assoliment de l'aprenentatge que s'ha dut a terme al llarg dels estudis d' Enginyeria Tècnica en Informàtica.

Aquest objectiu general es concreta en els següents:

- Analitzar un problema complex de tipus pràctic transformant-lo en un projecte informàtic.
- Planificar i estructurar el desenvolupament del projecte mitjançant l'elaboració d'un pla de treball aplicant una metodologia adient.
- Treballar a fons els aspectes formals del desenvolupament de projectes.
- Sintetitzar una solució viable i realista al problema proposat.
- Elaborar una memòria del projecte segons una estructura prefixada.
- Elaborar una presentació del desenvolupament i resultats finals del projecte.

Objectius específics.

Els objectius específics d'aquest treball són :

- Posar en pràctica els coneixements de base de dades adquirits en les assignatures de Bases de Dades I i Bases de Dades II.
- Conèixer el sistema d'excepcions i com es tracten en la base de dades Oracle.
- Conèixer i posar en pràctica el mecanisme de tractament de transaccions autònomes.
- Conèixer i posar en pràctica el nou tipus de dada ANYDATA TYPE.
- Adquirir coneixements pràctics de les classes de presentació JAVA (Swing).

1.3 Enfocament i mètode seguit.

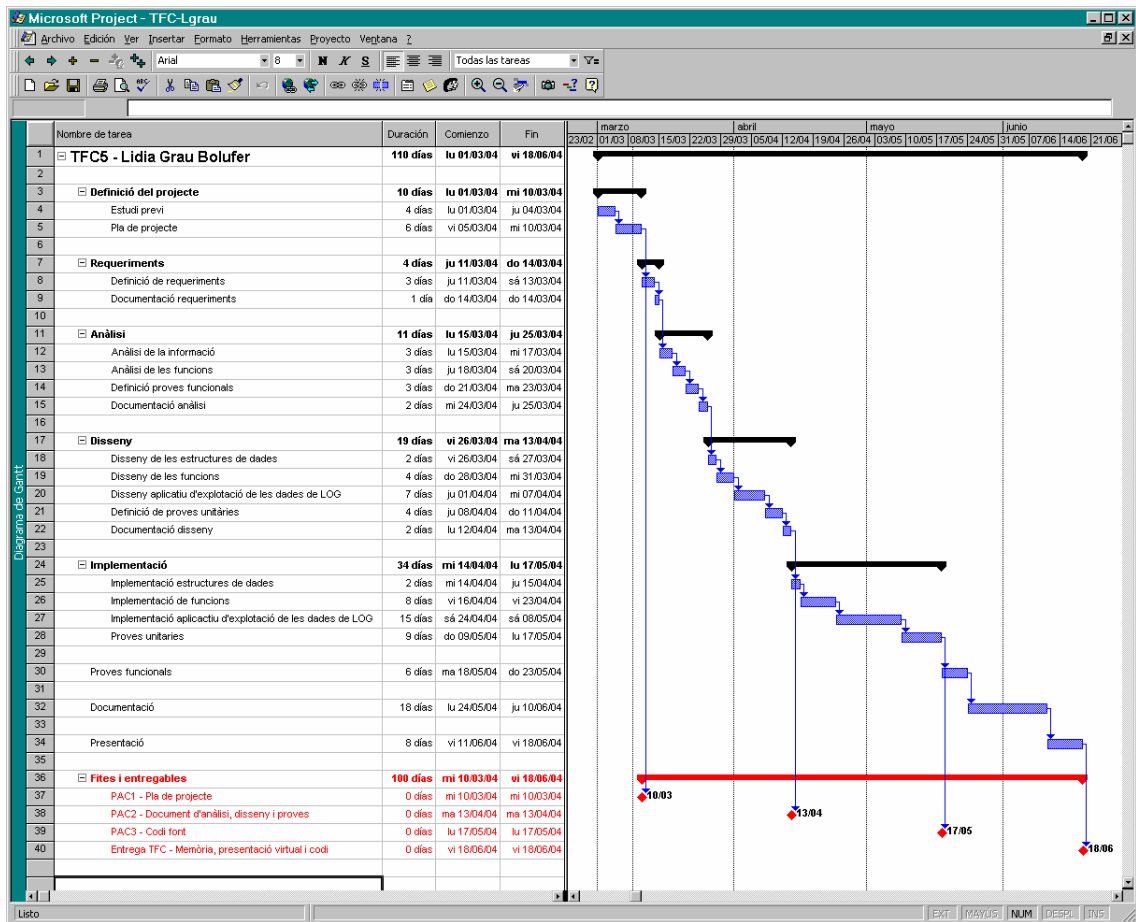
L'enfocament del projecte, pel que fa a la metodologia de desenvolupament triada, ve marcat per les peculiaritats pròpies d'un treball fi de carrera. D'una banda i com a qualsevol altre projecte, els recursos són escassos però, en aquest cas, la limitació és porta a l'extrem ja que l'únic recurs sóc jo mateixa. D'altra, s'imposen unes fites, marcades per les PACS, que impliquen l'elaboració de determinats entregables (pla de treball, documents d'anàlisi, disseny, fonts i executables del sistema).

Additional i òbviament, el projecte es veu afectat per les característiques del treball a dur a terme. Aquest és un TFC eminentment pràctic, es tracta d'un projecte de desenvolupament i no d'un estudi teòric, més concretament s'ha d'implementar un subsistema que ha de simplificar el control dels procediments emmagatzemats. Això implica que part dels requisits s'hagin d'obtenir de les característiques tècniques de la BB.DD. que s'utilitza, en aquest cas Oracle, i no directament del client.

Tots aquests condicionants i la meua experiència en el desenvolupament d'aplicacions amb PL/SQL, fan que em decanti per una metodologia clàssica o en cascada ja que el principal problema d'aquesta, les tornades enrere, es minimitza notablement pel fet de tenir una idea força clara des del començament. Un cop finalitzat el projecte, he de reconèixer, com no podia ser d'una altra manera, que han existit replantejaments derivats de problemes que es comentaran més endavant però, en cap cas, prou greus per invalidar la feina feta a cadascuna de les etapes.

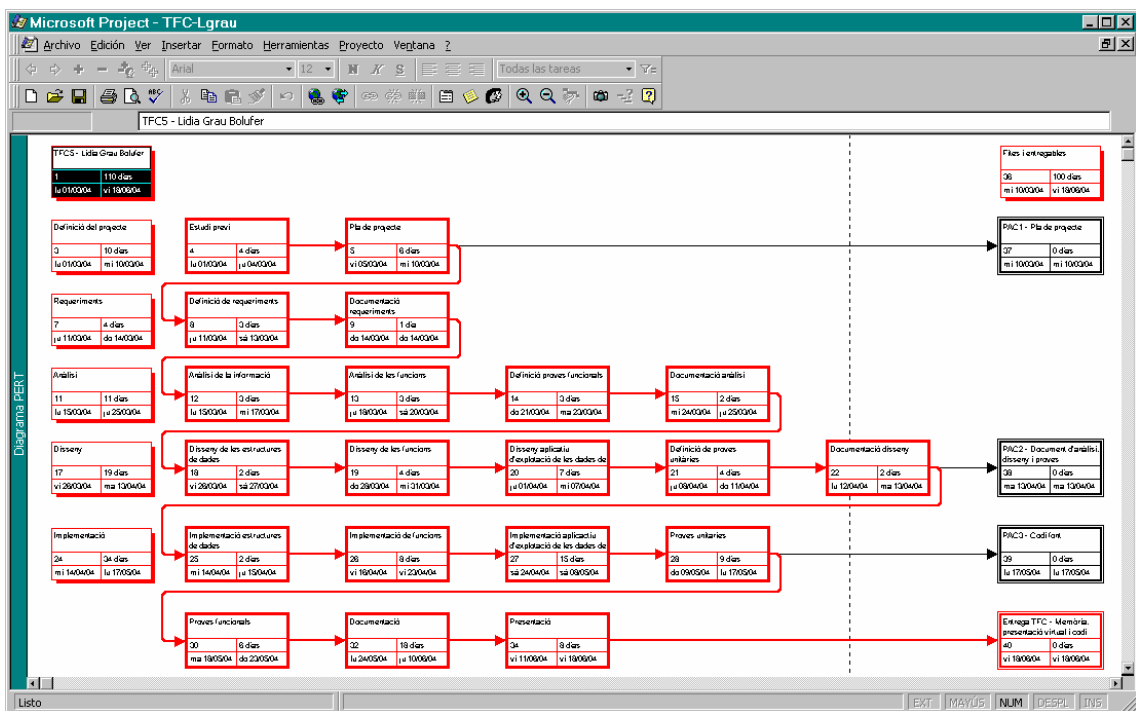
1.4 Planificació del projecte.

A continuació es presenta el diagrama de Gantt del projecte on es pot observar la planificació inicial, és a dir, la descomposició d'activitats, les relacions de dependència entre elles, i una estimació de l'esforç. No es mostra, per raons òbvies, una estimació de costos i tampoc un Gantt de seguiment ja que les derives no han estat significatives.



1. Diagrama de Gantt

Aquesta mateixa informació també es recull en el diagrama de PERT que es presenta a continuació i on, addicionalment es pot observar el camí crític, tot i que en un projecte d'aquestes característiques no té gaire interès.



2. Diagrama de PERT

1.5 Productes obtinguts.

Com a resultat del projecte de disseny d'un sistema de log genèric per a la implementació de sistemes operacionals he obtingut un conjunt d'objectes que formen l'estructura necessària per enregistrar les dades de log i l'aplicatiu de visualització que he anomenat VisorLog.

El contingut de l'entrega és el següent :

Fitxer	Descripció
Directorí : \estructura	
Creacio.sql	Crea la estructura de dades necessàries per el sistema de log
Borrat.sql	Elimina de la base de dades tots els objectes generats per el sistema de log
Utl_reg_log.pkb	Procediment amb les funcions necessàries per enregistrar les dades de log
Utl_vis_log.pkb	Cos del procediment amb les funcions per el visor
Utl_reg_log.pks	Especificació de les funcions necessàries per enregistrar les dades de log
Utl_vis_log.pks	Especificació de les funcions del visor
Utl_del_log.sql	Funció d'eliminació de les dades de log caducades
Directorí : \proves	
Test_variables.sql	Joc de proves de tots els tipus de dades PL/SQL (proves ANYDATA)
Test_config.sql	Joc de proves de la configurabilitat del sistema
Test_delete.sql	Joc de proves de la funció d'eliminació
Joc_de_proves	Creació de dades per a provar el VisorLog
Directorí : \aplicacio\fonts	
VisorLog.java	Classe principal de l'aplicatiu
VisorLogGUIAjuda.java	Interfície des de la que l'usuari té accés a la presentació de l'aplicatiu VisorLog i al manual d'usuari de l'aplicatiu.
VisorLogGUIBorrat.java	Interfície del procés d'eliminació de transaccions caducades
VisorLogGUIConfiguracio.java	Interfície que permet presentar i actualitzar els valors de parametrització de l'aplicatiu
VisorLogGUIEstadistiques.java	Interfície que permet presentar valors estadístics de la execució de transaccions.
VisorLogGUIFinestra.java	Gestiona la finestra de l'aplicatiu
VisorLogGUIIdentificacio.java	Interfície que permet obtenir les dades d'identificació de l'usuari, es a dir, el seu codi d'identificació i la seva clau d'accés a l'aplicatiu.
VisorLogGUIMenu.java	Interfície que permet presentar les opcions de menú de l'aplicatiu.
VisorLogGUINavegacio.java	Gestiona la interacció de l'usuari amb l'aplicatiu
VisorLogGUIObservacions.java	Interfície que permet presentar valors de les observacions, es a dir, els valors de les variables enregistrades a cada bloc d'execució de les transaccions
VisorLogGUISeguiment.java	Interfície que permet presentar els valors dels atributs de les transaccions enregistrades i seleccionar aquella de la que es vol fer un seguiment, visualitzant el seu arbre de execució, les traces emmagatzemades i les observacions enregistrades.
VisorLogGUITraces.java	Interfície que permet presentar l'arbre d'execució d'una transacció i els valors dels atributs de les traces enregistrades a cadascun del blocs d'execució, així com seleccionar una traça per fer un seguiment de les observacions de variables associades.
VisorLogRSCArbreExecucio.java	Conté un arbre que representa la execució del conjunts de blocs que formen una transacció. La relació jeràrquica representa la crida entre blocs i

Fitxer	Descripció
	la relació d'ordre representa l'ordre d'execució.
VisorLogRSCBorrat.java	Proporciona els mètodes necessaris per obtenir la data de caducitat de transaccions segons la configuració de l'aplicatiu i per activar el procés d'eliminació de transaccions caducades.
VisorLogRSCConfiguracio.java	Conté els atributs de parametrització de l'aplicatiu
VisorLogRSCConnexio.java	Gestiona els objectes persistents de l'aplicatiu
VisorLogRSCEstadistiques.java	Conté una taula amb els valors estadístics de les transaccions seleccionades
VisorLogRSCException.java	Declara una excepció per gestionar els error, avisos o missatges de l'aplicatiu de seguiment
VisorLogRSCObservacions.java	Conté una taula amb valors dels atributs de les observacions d'una traça.
VisorLogRSCTraces.java	Conté una taula amb els valors dels atributs de les traces d'un bloc d'execució.
VisorLogRSCTransaccions.java	Conté una taula amb els valors dels atributs de les transaccions seleccionades.
Directorio : \aplicacio\documentacio	
*.html	Documentació de l'aplicació VisorLog. S'accedeix amb el fitxer <i>índex</i> .
Directorio : \aplicacio\executables\VisorLog	
VisorLog.class	Classes del package VisorLog
Directorio : \aplicacio\executables\VisorLogGUI	
*.class	Classes del package VisorLogGUI
*.gif	Imatges de l'aplicatiu
VisorLogManualUsuari.*	Fitxers amb el manual d'usuari de l'aplicació VisorLog
Directorio : \aplicacio\executables\VisorLogRSC	
*.class	Classes del package VisorLogRSC

3. Taula de productes obtinguts

1.6 Altres capítols de la memòria.

La resta de capítols de la memòria recullen la documentació obtinguda a cada una de les fases del projecte i descriuen la tasca que he realitzat en cada una d'elles.

El capítol 2, està dedicada a la especificació de requeriments, és l'etapa de definició i estudi del problema i és a la que defineixo l'abast del projecte.

En el capítol 3, anàlisi, especifico el que ha de fer el sistema. Per això, en primer lloc s'estudien les dades d'interès i com estan relacionades i també quins processos s'han de dur a terme mitjançant el diagrames de flux de dades. Finalment, realitzo el disseny de les interfícies del sistema i defineixo el joc de proves funcional.

En el capítol 4, disseny, s'identifiquen els components que formen el sistema, definint per cadascun d'ells les proves unitàries i es descriu el procés de traducció de l'estructura lògica de dades, obtinguda a l'etapa d'anàlisi, en una estructura física.

En el capítol 5, Manual d'usuari del sistema de log, es detalla la forma d'utilitzar el sistema de log genèric. Defineix la forma d'instal·lar i desinstal·lar el producte, la normativa de codificació per als sistemes operacionals i el manual per utilitzar l'aplicatiu que explota les dades de log.

El capítol 6, recull les conclusions del TFC.

2. Especificació de requeriments.

El sistema de log pretén ser una eina genèrica de suport a qualsevol sistema que requereixi un control exhaustiu de la seva execució. Per assolir-ho, caldrà fer una generalització del que suposa l'execució d'una funció d'un sistema operacional, és a dir, cal descriure el que succeeix des de que l'operador realitza una petició fins que el sistema li dona una resposta.

La petició de l'usuari inicia l'execució d'una transacció que és una unitat lògica de treball que ha de ser confirmada o desestimada indivisiblement. En general, no es pot assegurar que això s'acompleixi a tots els sistemes operacionals, és possible que una petició d'usuari desencadeni l'execució de més d'una transacció i, fins i tot, que varies peticions s'incloguin dins de la mateixa transacció. Aquesta és una decisió que haurà de prendre qui dissenyi el sistema però, per facilitar la interacció amb l'operador, és més que recomanable que una petició d'usuari desencadeni una única transacció i que aquesta només inclogui les accions derivades de la petició que l'ha iniciat. En qualsevol cas i des del punt de vista del sistema de log, la unitat de treball haurà de ser la transacció i, per tant, s'haurà d'identificar unívocament amb un nom que haurà de ser conegut per l'operador per tal de fer el seguiment.

L'inici d'una transacció implica, simplificant-ho força, l'execució del que anomenarem una instrucció que no és més que una acció que gestiona informació. En principi, una instrucció coincideix amb l'execució de les sentències (assignació, control de flux, interacció amb el gestor de BB.DD., etc...) que proporcionen directament els llenguatges de programació. Però els llenguatges, i particularment PL/SQL, permeten la construcció de blocs de sentències ordenades seqüencialment i que, finalment, poden ser considerats com una ampliació de les sentències bàsiques del llenguatge. Per tant i continuant amb la descripció del que succeeix quan l'operador realitza una petició, l'inici d'una transacció implica l'execució d'una instrucció que pot ser bàsica però que, habitualment, serà un bloc que estarà format per altres instruccions que s'executen si són simples o que provoquen l'entrada en un altre bloc si són compostes. Aquesta estructura reflexiva es repeteix tantes vegades com calgui per implementar l'algorisme pertinent. Finalment, quan s'assoleix un punt de sortida d'un bloc de sentències, es retorna el control al bloc que l'ha cridat que, al seu torn, segueix la seva seqüència d'execució fins a arribar a un dels seus punts de sortida. I així, successivament, fins a arribar a un punt de sortida del bloc iniciat per la transacció, la qual cosa implica la finalització d'aquesta.

2.1 Funcions del sistema de log.

Un cop descrita l'execució d'una funció del sistema operacional, és el moment de detallar el que s'espera del sistema de log. En principi, es poden distingir tres tipus de funcions, les de seguiment, les estadístiques i les de configuració.

2.1.1 Funcions de seguiment.

Les funcions de seguiment han de permetre que l'operador o, si arribés el cas, el desenvolupador del sistema operacional segueixin el flux d'execució de les diferents instruccions per detectar possibles incorreccions en la utilització o malfuncionaments del sistema. El primer que cal assenyalar és que no s'ha d'enregistrar el pas per totes les instruccions sinó només per aquelles que es considerin necessàries per rastrejar el camí que ha seguit el procés. En aquest sentit, es considera imprescindible anotar el pas pel punt d'entrada i pel punt de sortida de cada bloc de instruccions, identificant clarament l'estat de finalització d'aquest (correcte, avís, incorrecte). Per contra, l'enregistrament del pas per altres punts es deixa a criteri del desenvolupador que podrà identificar els que consideri

significatius. L'execució de les instruccions escollides haurà d'emmagatzemar informació que permeti reconèixer el punt de traça, identificant el bloc i la posició dins d'aquest, informació relativa a la data i hora en que es va executar així com el valor de les variables d'interès (paràmetres, valor de retorn, altres variables locals) per dur a terme la tasca de seguiment.

Pel que fa a l'aplicatiu que ha de permetre el seguiment del flux d'execució, haurà de proporcionar les eines per seleccionar la transacció que es vol revisar, per visualitzar el seu arbre d'execució i per mostrar els valors de les variables observades a cada punt de traça. Les branques de l'arbre d'execució representaran els blocs d'instruccions pels que ha transcorregut el procés i se li haurà de dotar de dues funcionalitats bàsiques, d'una banda la possibilitat de diferenciar fàcilment els blocs que arriben al punt de sortida correcte dels que han finalitzat amb la generació d'un avís o un error, de l'altre i degut a la complexitat que pot arribar a tenir, la capacitat de seleccionar l'àmbit de visibilitat mitjançant la expansió o contracció de les branques adients en cada moment.

2.1.2 Funcions estadístiques.

El sistema haurà de proporcionar les funcions que permetin identificar :

- Procediments més utilitzats, globalment, per categories, per franges horàries i per usuari.
- Procediments agrupats per resultat.
- Franges horàries d'utilització, globalment i per usuari.

2.1.3 Funcions de configuració.

El sistema de log pot interferir en el funcionament normal del sistema operacional, especialment en el seu rendiment depenent de la grandària de les variables que es volen observar i de la profunditat de l'arbre d'execució. Per tal de no limitar les funcionalitats permeses, impeding l'emmagatzemament de determinats tipus PL/SQL, sinó proporcionant eines per fer-ho de forma controlada, caldrà que el sistema permeti configurar els següents aspectes :

- Tipus de dades que s'enregistraran (escalars, objectes).
- Tipus de variables que s'enregistraran (paràmetres, retorn, locals).
- Grandària màxima de la variables a enregistrar.
- Profunditat màxima de l'arbre d'execució.

D'una altra banda, també cal preveure el procés d'esborrat de les traces que eliminarà les que superin un interval de permanència en el sistema que haurà de ser configurable.

2.2 Altres entregables del projecte.

Com hem vist anteriorment, les funcions del sistema de log es basen, principalment, en una selecció acurada del punts de traça i de les variables a observar en cadascun d'ells. Aquesta tria la realitza el desenvolupador del sistema operacional pel que es fa imprescindible la confecció d'una normativa o metodologia de codificació que garanteixi el correcte funcionament del sistema.

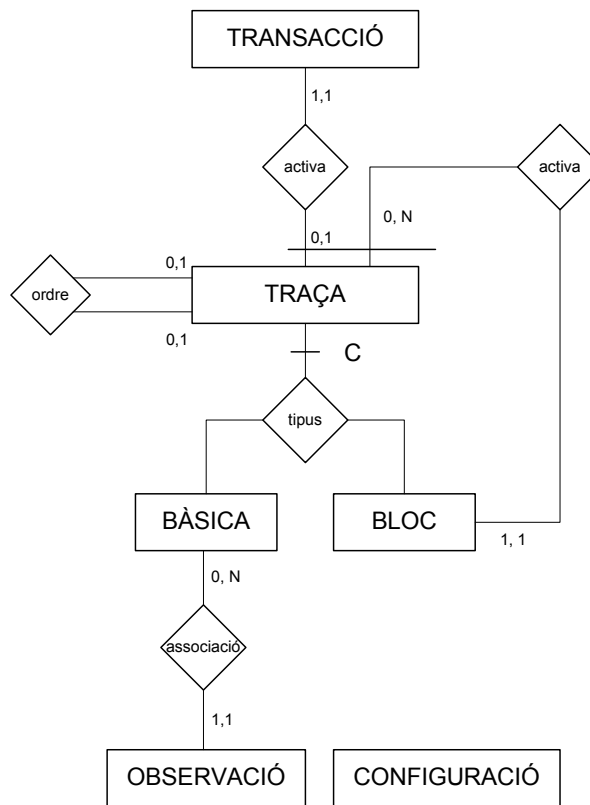
3. Anàlisi.

En aquesta fase es defineix conceptualment la solució al problema plantejat a la fase d'especificació de requeriments.

3.1 Model de dades.

A continuació es defineixen les dades d'interès per el sistema de log genèric i com es relacionen entre si a partir dels conceptes introduïts a la fase d'especificació de requeriments.

3.1.1 Diagrama d'E/R.



4. Diagrama d'entitats i relacions

3.1.2 Definició d'entitats.

Entitat	Descripció
CONFIGURACIÓ	Variables de definició del comportament del sistema de log
TRANSACCIÓ	Des de el punt de vista del sistema de log genèric, és una unitat lògica de treball que ha de ser totalment confirmada o desestimada indivisiblement.
TRAÇA	Instrucció rellevant des de el punt de vista del seguiment del flux d'execució d'una transacció.
TRAÇA BÀSICA	Traça formada per a una única instrucció.
TRAÇA BLOC	Traça formada per a un conjunt d'instruccions ordenades seqüencialment.
OBSERVACIÓ	Valors de les variables dels blocs d'instruccions en el instant de generació d'una traça.

5. Taula de descripció d'entitats.

3.1.3 Definició d'atributs.

CONFIGURACIÓ

Atribut	Descripció	Domini
Informatives	Indicador que defineix la restricció d'emmagatzemar les traces de tipus informatives.	S/N
Profunditat	Profunditat del arbre d'execució que el sistema de log permet emmagatzemar	
Paràmetres	Indicador que defineix la restricció d'emmagatzemar els paràmetres d'E/S	S/N
Valors retorn	Indicador que defineix la restricció d'emmagatzemar els valors de retorn	S/N
Variables locals	Indicador que defineix la restricció d'emmagatzemar les variables locals	S/N
Escalars	Indicador que defineix la restricció d'emmagatzemar les observacions de tipus de dades escalars	
LOBs	Indicador que defineix la restricció d'emmagatzemar les observacions de tipus de dades LOBs	
Grandària	Màxim valor en bytes que por ocupar un valor a emmagatzemar	
Caducitat	Nombre de dies que es manté una transacció en el sistema	

6. Taula de descripció dels atributs de l'entitat CONFIGURACIÓ

No li cal clau única ja que té una única ocurrència.

TRANSACCIÓ

Atribut	Descripció	Domini
Nom	Identificador extern de la transacció, serà l'atribut que permetrà que l'operador del sistema l'identifiqui.	
Usuari	Usuari que ha executat la transacció	

7. Taula de descripció dels atributs de l'entitat TRANSACCIO

El nom és la clau única.

TRAÇA

Atribut	Descripció	Domini
Tipus	Tipus de traça	Bàsica Bloc

8. Taula de descripció dels atributs de l'entitat TRAÇA

La clau única és la relació activa i la relació d'ordre. Com la relació activa és amb una transacció o amb un bloc la clau es compona d'aquesta or més la relació d'ordre.

BÀSICA

Atribut	Descripció	Domini
Tipus	Tipus de traça	Inici procés Fi correcte Fi erroni Fi amb avís Informativa
Data	Data d'execució	
Comentari	Text lliure a guardar amb la traça	
Codi error	Codi d'error intern del programa en execució	
Error oracle	Error oracle produït	

9. Taula de descripció dels atributs de l'entitat BÀSICA

BLOC

Atribut	Descripció	Domini
Nom	Nom del procediment o funció	
Categoria	Categorització del procediment. Útil des del punt de vista estadístic. Per exemple si el sistema operacional està dissenyat amb una arquitectura de capes es podrien definir les següents categories : Interfície client, Regla de negoci, Interfície objecte-relacional, Altres	
Data inici	Data d'inici de l'execució del bloc	
Data fi	Data de finalització de l'execució del bloc	
Resultat	Estat de finalització del bloc	Correcte Error Avis

10. Taula de descripció dels atributs de l'entitat BLOC

OBSERVACIÓ

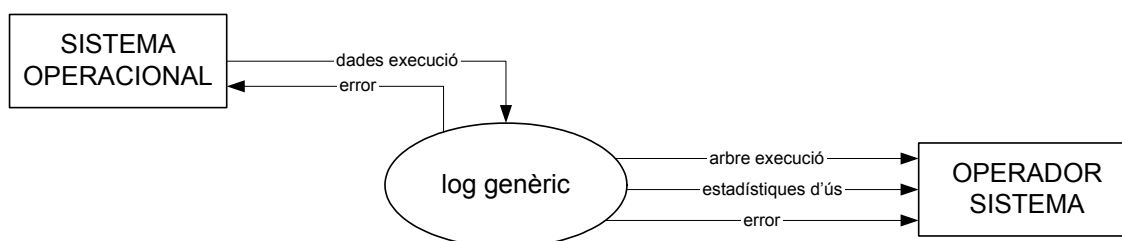
Atribut	Descripció	Domini
Nom	Nom de la variable	
Tipus	Tipus de variable	Entrada Sortida Retorn Local
Tipus de dades	Tipus de dades de la variable	Numèrica Alfanum.,...
Longitud	Longitud de la variable	
Precisió	Precisió de la variable	
Escala	Factor d'escala de la variable	
Valor	Valor que pren la variable	

11. Taula de descripció dels atributs de l'entitat OBSERVACIO

La clau es la relació d'associació més el nom.

3.2 Diagrames de flux de dades.

Veiem a continuació la comunicació entre el sistema i el seu entorn mitjançant el diagrama de context :



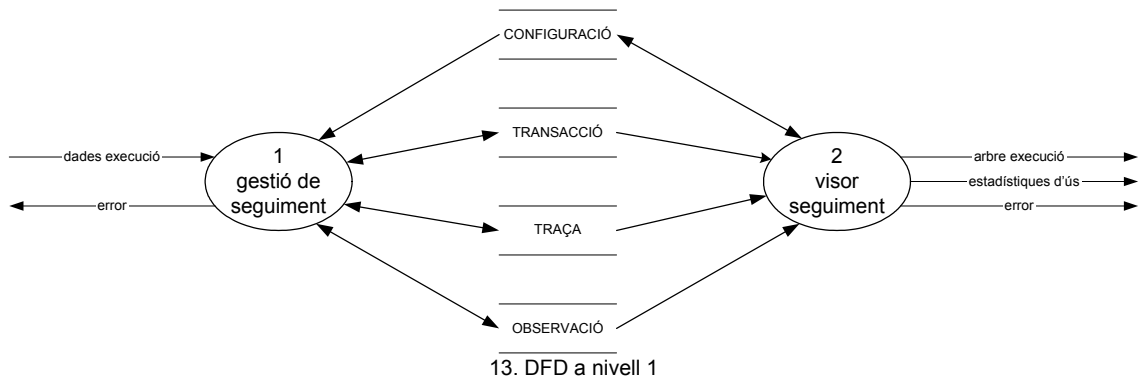
12. Diagrama de context

En aquests cas els emissors de la informació que rep el sistema són altres sistemes que proporcionen les dades d'execució dels mateixos. Estan representats en el diagrama de context amb l'entitat externa SISTEMA OPERACIONAL. El receptor de la informació que genera el sistema és l'operador, representat amb l'entitat externa OPERADOR SISTEMA.

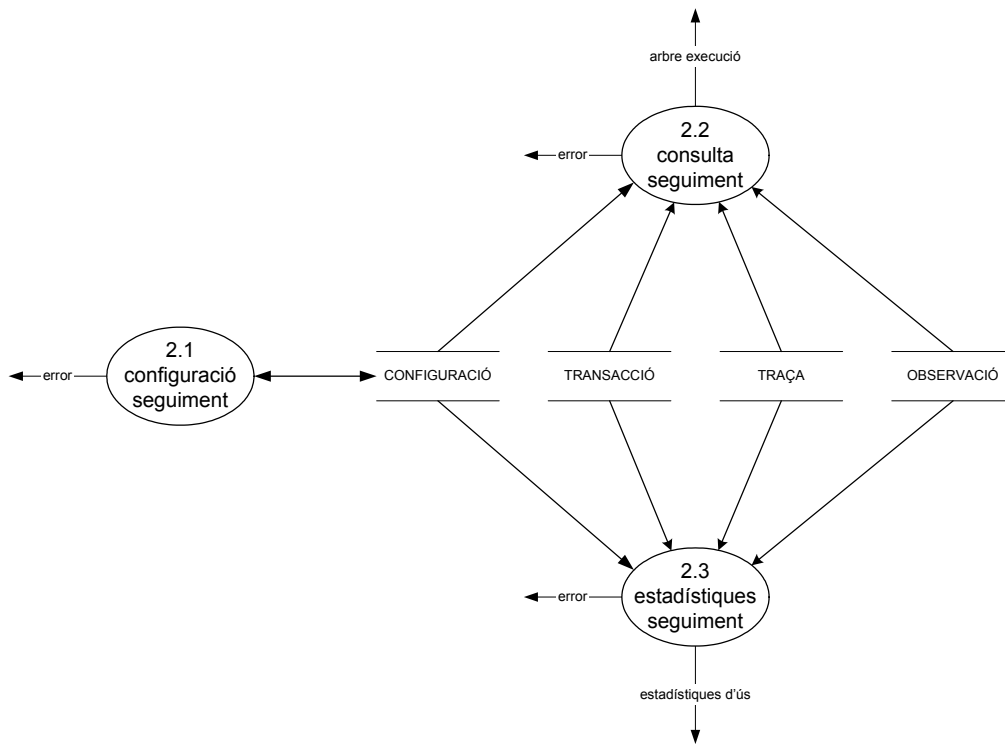
Hi ha un únic flux d'informació d'entrada al sistema, que és dades d'execució.

Ara ens cal definir les tasques que el sistema ha de dur a terme. En el nostre sistema identifiquem dos processos:

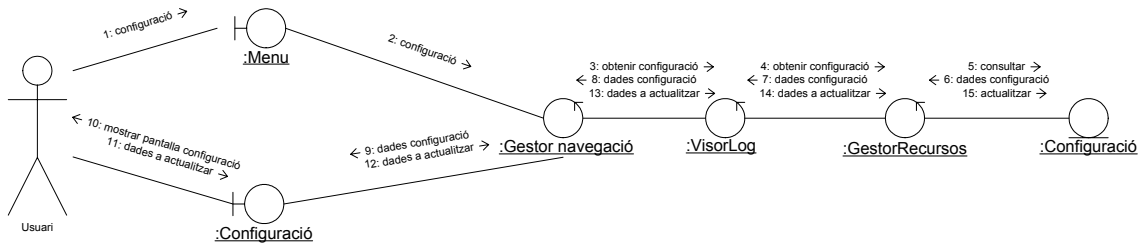
- El procés de gestió de seguiment que rep les dades d'execució dels sistemes operacionals i les emmagatzema a les entitats que hem definit en el diagrama d'E/R.
- El visor de seguiment que és el procés que ens permetrà visualitzar l'arbre d'execució d'un procés i obtenir estadístiques d'ús relatives a la freqüència d'execució mitjançant les dades emmagatzemades.



A continuació descriuré les tasques que ha de portar a terme el procés 2-visor seguiment. Per fer-ho el detallaré amb un DFD de nivell 2 i descriuré els processos identificats de forma textual i amb un diagrama de col·laboració simplificat que permeti observar les classes frontera, les de control, les d'entitat i les operacions associades.

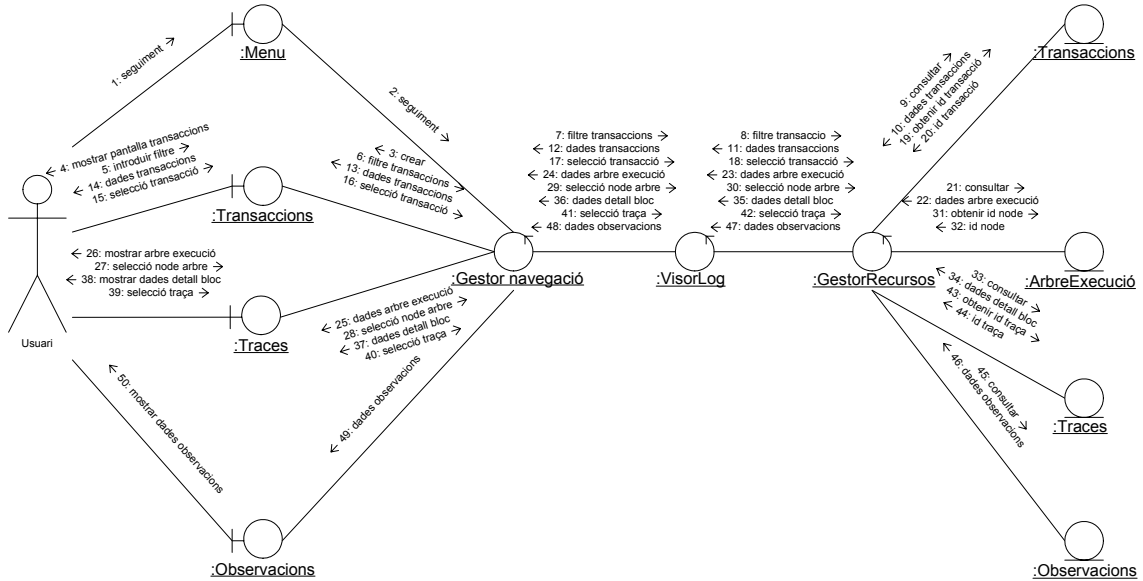


- El procés de configuració de seguiment que permet a l'administrador del sistema definir el comportament del sistema de log, independentment del flux de dades que rebí del sistema operacional.



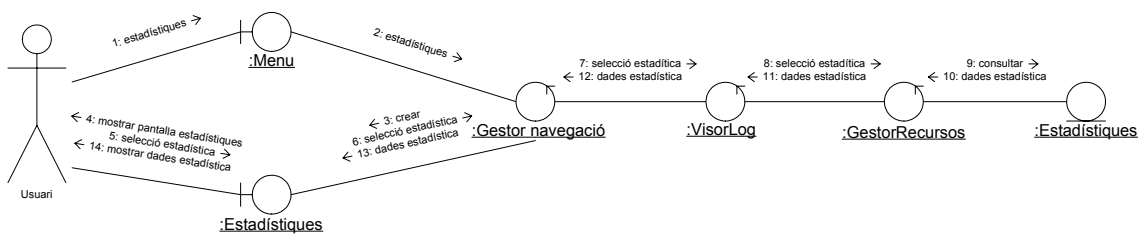
15. Diagrama de col·laboració de la configuració de seguiment

- La consulta de seguiment permet obtenir a l'operador del sistema l'arbre d'execució per facilitar la depuració del codi i identificar i seguir els errors que s'hagin produït.



16. Diagrama de col·laboració de la consulta de seguiment

- Per últim, el procés d'estadístiques de seguiment, permet a l'operador del sistema obtenir informació estadística dels diferents arbres d'execució emmagatzemats.



17. Diagrama de col·laboració d'estadístiques de seguiment

3.3 Disseny extern : disseny d'interfícies.

A continuació es descriuen les interfícies associades al procés 2 (visor seguiment), el procés de gestió de seguiment no té un tractament interactiu i per tant no té cap interfície d'usuari associada.

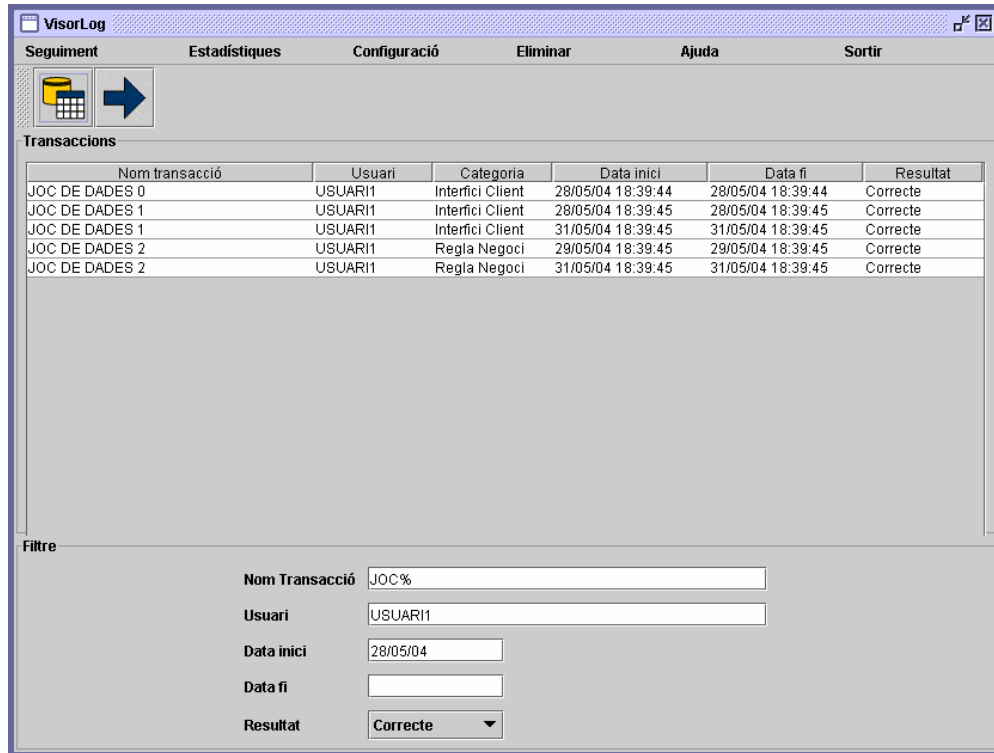
Menú del visor de seguiment :



18. Disseny del menú del visor de seguiment

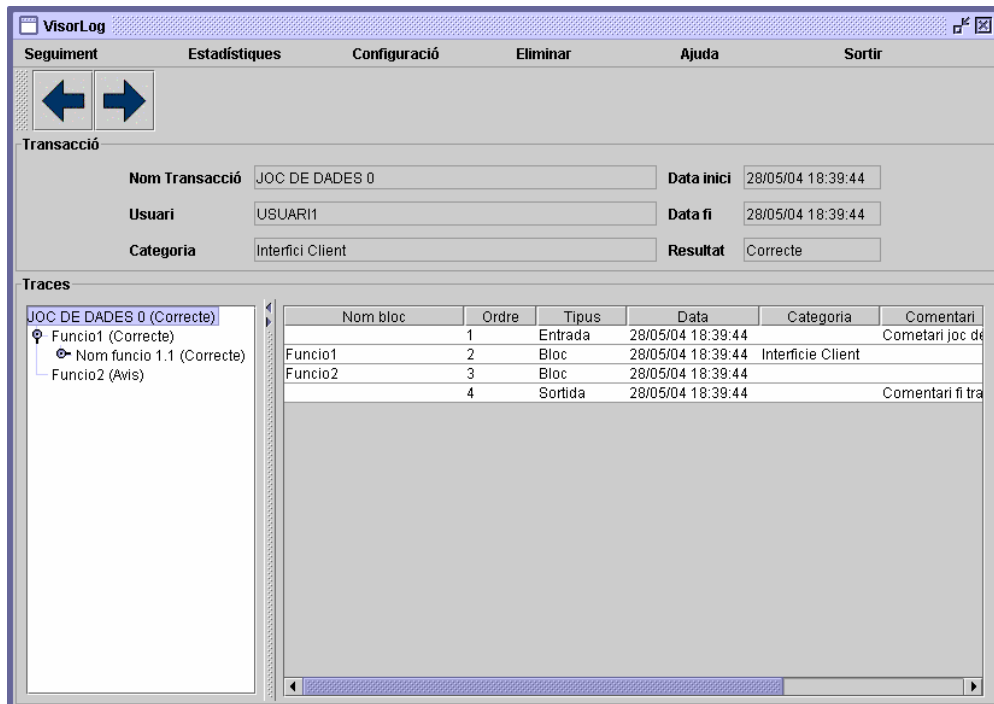
Consulta següent :

Aquesta pantalla permetrà realitzar un filtre per localitzar l'arbre d'execució que es vol consultar.



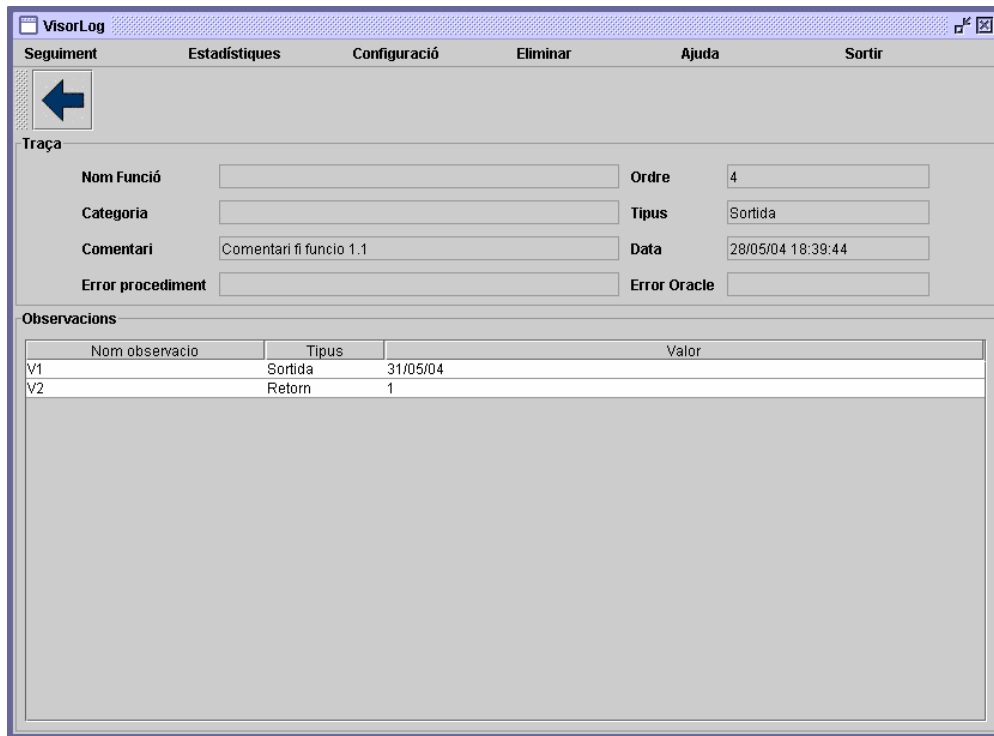
19. Disseny de la pantalla de filtre de transaccions

Un cop triat un arbre d'execució, accedirem a la pantalla que ens mostrarà la informació relativa a l'arbre d'execució :



20. Disseny de la pantalla amb les dades de l'arbre d'execució

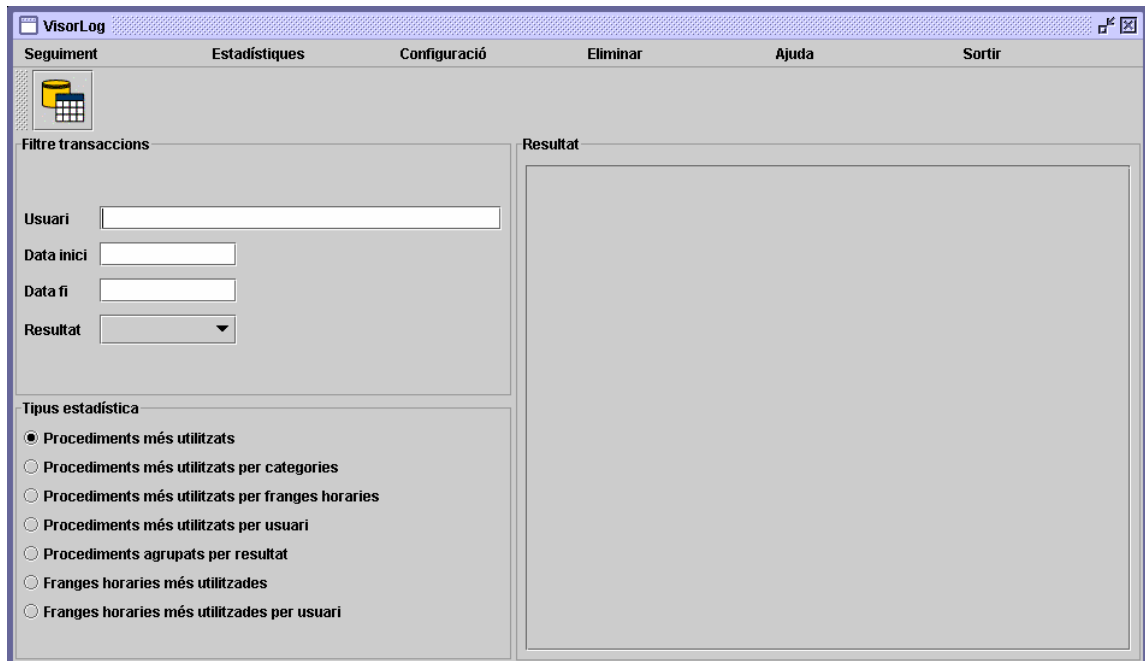
Situats sobre una traça podem accedir a la pantalla d'informació d'una traça, des de on podrem consultar tota la informació relativa a la traça i totes les observacions associades.



21. Disseny de la pantalla d'observacions

Estadístiques seguiment :

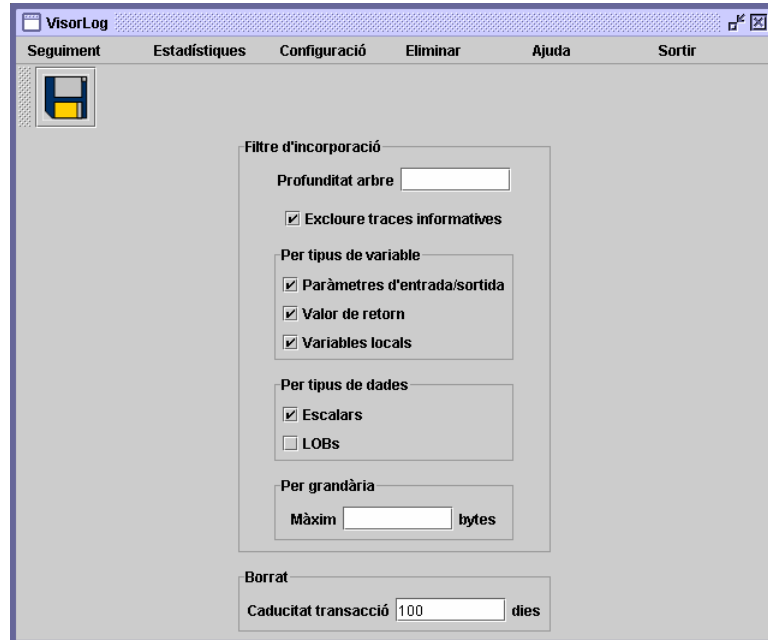
Aquesta pantalla ens permetrà triar una estadística a mostrar i permetrà acotar les traces que es consideraran per a generar l'estadística.



22. Disseny de la pantalla d'estadístiques

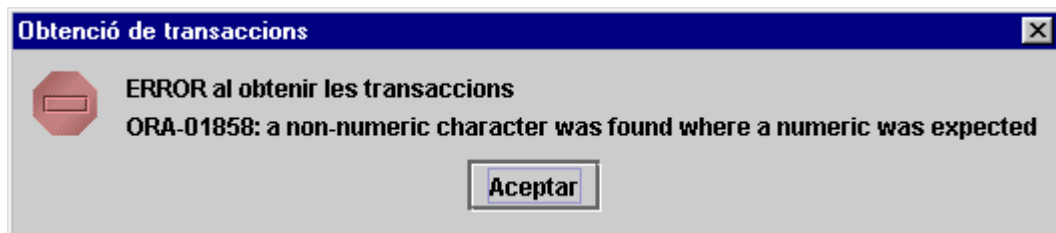
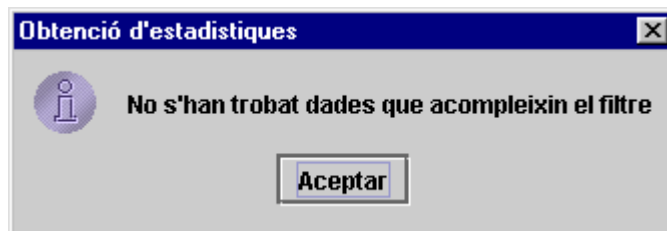
Configuració seguiment :

Aquesta pantalla ens permetrà configurar el comportament del sistema de log, és una eina per tal de que l'administrador del sistema pugui limitar per diferents conceptes el que el sistema genèric anirà emmagatzemant i quan s'esborrarà automàticament de la base de dades.



23. Disseny de la pantalla de configuració

Altres pantalles :



24. Disseny dels missatges d'avís o error

3.4 Definició de proves funcionals.

Codi	Descripció
LOGF_M001	El menú del sistema de log permet accedir a les tres funcionalitats identificades, seguiment, estadístiques i configuració.
LOGF_S001	El sistema permet visualitzar les transaccions executades, mostrant el seu nom, el de l'usuari que la ha iniciat, les dates d'inici i finalització i el resultat obtingut.
LOGF_S002	Permet acotar les transaccions a visualitzar, filtrant per qualsevol dels seus atributs.
LOGF_S003	Permet seleccionar una transacció per fer el seguiment del seu flux d'execució.
LOGF_S004	Permet visualitzar el flux d'execució d'una transacció en forma d'arbre on les branques representaran els blocs d'instruccions pels que ha transcorregut el procés
LOGF_S005	El sistema permet expandir les branques de l'arbre d'execució per visualitzar els blocs més interns.
LOGF_S006	El sistema permet contreure les branques de l'arbre d'execució per simplificar la visió que l'operador té del mateix.
LOGF_S007	El sistema permet seleccionar una branca de l'arbre d'execució per visualitzar la informació associada al bloc de traces que representa.
LOGF_S008	El sistema permet visualitzar les traces associades al bloc seleccionat, informant del ordre d'execució, del nom de la funció, la seva categoria, el seu tipus i de la data i hora d'execució.
LOGF_S009	El sistema permet seleccionar una traça per visualitzar la informació de les observacions associades.
LOGF_S010	El sistema permet visualitzar les observacions enregistrades en el moment de generació de la traça, informant del nom de la variable, del seu tipus, opcionalment de les seves característiques i, especialment del seu valor.
LOGF_E001	El sistema permet l'obtenció d'estadístiques de procediments més utilitzats, globalment, agrupant per categories, franges horàries o per usuari.
LOGF_E002	Permet l'obtenció d'estadístiques de procediments agrupats pel resultat obtingut.
LOGF_E003	El sistema permet l'obtenció d'estadístiques de franges horàries d'utilització del sistema, globalment o agrupant per usuari.
LOGF_E004	El sistema permet restringir l'àmbit de l'estudi estadístic, filtrant les traces a processar per resultat, data d'execució i nom de l'usuari que va iniciar la transacció.
LOGF_E005	El sistema permet visualitzar el resultat de l'informe estadístic en forma tabular.
LOGF_C001	El sistema permet configurar els tipus de dades que s'enregistraran.
LOGF_C002	El sistema permet configurar els tipus de variables que s'enregistraran.
LOGF_C003	El sistema permet configurar la grandària màxima de la variables a registrar.
LOGF_C004	El sistema permet configurar la profunditat màxima enregistrable de l'arbre d'execució.
LOGF_C005	El sistema permet configurar l'interval de permanència de les transaccions en el sistema.
LOGF_C006	El sistema permet la visualització dels valors establerts per la configuració del funcionament del sistema de log.
LOGF_C007	El sistema permet la modificació dels valors establerts per la configuració del funcionament del sistema de log.
LOGF_C008	Els sistema permet eliminar les traces obsoletes.

25. Taula de definició de proves funcionals

4. Disseny.

4.1 Model de dades. Ús d'ANYDATA TYPE i NESTED TABLES.

En aquest punt cal construir el model físic a partir del model conceptual definit a l'etapa d'anàlisi.

Ens fixem en primer lloc a la generalització completa que hi ha a l'entitat TRAÇA. Encara que podria semblar que les entitats específiques no tenen gaires atributs en comú opto per implementar una única taula aprofitant la característica que proporciona la base de dades d'oracle que els atributs buits no ocupen espai a la base de dades. També cal considerar possibles guanys en eficiència, però sobre tot la possibilitat d'utilitzar la clàusula **CONNECT BY** per a obtenir l'arbre d'execució amb una sola sentència SQL. Per aquesta mateixa raó també puc agrupar a la mateixa taula l'entitat transacció que serà la traça de tipus bloc que no tingui l'apuntador al procediment que l'activa, i a més serà a l'única traça on s'emmagatzemarà l'usuari.

Ara només queda revisar l'entitat observacions. Per enregistrar les observacions cal poder emmagatzemar qualsevol valor. Per això ens cal poder emmagatzemar atributs de forma genèrica, és a dir, atributs del que no sabem el tipus fins que s'executa el codi. El primer que penso és a dissenyar un objecte on emmagatzemar el valor en un VARCHAR2 i afegir els atributs necessaris per a identificar el tipus, és a dir, el tipus de dades, la longitud, la precisió i l'escala. Un cop d'ull a les noves funcionalitats d'Oracle9 m'ofereix l'oportunitat de simplificar el problema i alhora treballar amb un nou tipus que pot representar qualsevol tipus de dades i que ofereix els mètodes per a convertir qualsevol variable en un 'any type' i el valor emmagatzemat un altre cop en un escalar o objecte. Evidentment m'estic referint al '**SYS.ANYDATA TYPE**'.

Si genero un objecte amb els atributs nom, tipus de paràmetre (entrada / sortida / local o retorn) i el valor i defineixo una taula que emmagatzema aquests tipus d'objectes puc definir un atribut observació a la taula de TRAÇA d'aquests tipus com a una **NESTED TABLE**.

En aquest disseny descobreixo un problema i és que a la versió actual no es suporta emmagatzemar LOBs en columnes persistent de tipus ANYDATA. El fet de que Oracle recalqui que 'encara' no està permès, em fa confiar que a versions futures es podrà fer i mantinc el disseny realitzant conversions per a aquests tipus d'objectes que es podrien eliminar en pròximes versions en cas de que realment s'elimini la restricció.

Per tant, el sistema de log finalment només constarà de la taula CONFIGURACIÓ que inclou les variables que defineixen el comportament del sistema de log i la taula TRAÇA que emmagatzema tota la informació relativa als arbres d'execució.

Així el model de dades quedarà format per els següents objectes :

- La seqüència SEQ_ID.

Per tal d'obtenir els identificadors interns per emmagatzemar a la columna identificador es farà servir una seqüència.

- *La taula TRAÇA*

Per tal de definir el tipus de la columna que contendran els valors de totes les observacions (paràmetres, valor de retorn,...) associades a una traça, ens caldrà encara definir un parell d'objectes més a la base de dades :

- El tipus TP_OBSERVACIO com un registre format per :

Columna	Tipus	Domini
Nom	VARCHAR2(30)	
Tipus	VARCHAR2(1)	Paràmetre Entrada Paràmetre Sortida Valor de Retorn Variable Local
Valor	SYS.ANYDATA	

26. Taula de definició dels atributs del objecte TP_OBSERVACIO

- El tipus TP_TB_OBSERVACIONS con una taula de TP_OBSERVACIO.

Les columnes que formen la taula TRAÇA són les que veiem a continuació :

Columna	Tipus	Obligatori	Domini
Identificador	NUMBER	Si	
Nom	VARCHAR2(30)		
Usuari	VARCHAR2(30)		
Ordre	NUMBER	Si	
Tipus	VARCHAR2(1)	Si	Bloc Entrada Sortida Informativa
Data inici	DATE	Si	
Data fi	DATE		
Resultat	VARCHAR2(1)		Correcte Erroni Avis Desconegut
Categoria	VARCHAR2(30)		(1)
Comentari	VARCHAR2(2000)		
Error procediment	VARCHAR2(2000)		
Error oracle	NUMBER		
Observació	TP_TB_OBSERVACIONS		
Pare	NUMBER		

27. Taula de definició dels atributs de la taula TRAÇA

(1) Tot i que es podria establir un domini a la columna categoria es preferible donades la diversitat de dissenys que pot existir a cada organització, permetre que sigui cada una d'elles que es defineixi opcionalment les categories en que vol agrupar els procediments dels seus sistemes operacionals.

La columna pare és una clau forana (FK) a la traça que la precedeix o activa.

L'identificador únic es fa servir com a clau primària (PK).

La relació reflexiva amb traça (pare) i la relació d'ordre implementada amb l'atribut ordre formen un índex no únic.

- *La taula CONFIGURACIO*

Les columnes que formen la taula CONFIGURACIO són les que veiem a continuació :

Columna	Tipus	Obligatori	Domini
Informatives	VARCHAR2(1)	S	S/N
Profunditat	NUMBER		
Paràmetres	VARCHAR2(1)	S	S/N
Valors retorn	VARCHAR2(1)	S	S/N
Variables locals	VARCHAR2(1)	S	S/N
Escalars	VARCHAR2(1)	S	
LOBs	VARCHAR2(1)	S	
Grandària	NUMBER		
Caducitat	NUMBER	S	

28. Taula de definició dels atributs de la taula CONFIGURACIO

La taula CONFIGURACIO constarà d'una única fila, per aquest motiu no cal definir cap identificador únic, ni tant sols la clau primària (PK).

4.2 Procediments necessaris.

Alhora de dissenyar el package per emmagatzemar tots els procediment i funcions necessaris en el sistema de log genèric, cal adonar-se de que els dos processos principals obtinguts a la fase d'anàlisi : gestor de seguiment i visor de seguiment mai s'executaran plegats. Mentre el primer estarà executant-se de forma diferida cada cop que s'arranqui un sistema operacional que faci servir el sistema de log, el segon només s'executarà quan l'administrador del sistema o qualsevol operador executin el visor de seguiment.

Per aquesta raó, i per no tenir carregat a memòria tot el package en moments en que possiblement el visor no s'estarà executant, cal crear dos packages, un amb les funcions necessàries per al gestor de seguiment i un altre amb les funcions per al visor. Cal adonar-se també que una de les funcions definides a la part de configuració, la de l'esborrat d'arbres d'execució caducats es podria activar de forma diferida, és a dir, es podria activar diàriament de forma automàtica, i per aquesta raó no caldrà que s'inclogui en el mateix package. Es crearà un procediment per realitzar els esborrats automàtics.

El package UTL_REG_LOG serà el package que contindrà les funcions per enregistrar les traces. El package UTL_VIS_LOG serà el package que contindrà les funcions per el visor de seguiment.

4.2.1 El package UTL_REG_LOG. Ús de transaccions autònomes i sobrecàrrega de funcions.

El sistema de log ha d'enregistrar el pas per una instrucció independentment del resultat de la transacció del sistema operacional. És a dir, cal que s'anotin les traces tot i que es desestimi la transacció principal. Per aquesta raó les funcions del sistema de log fan ús de les **transaccions autònomes**.

Una transacció autònoma és una transacció independent iniciada per una altra que es considera la principal. Permet confirmar o desestimar un conjunt de accions, independentment de la confirmació o desestimació de la transacció principal. Quan es crida una transacció autònoma la transacció original (la que la crida) queda temporalment

suspesa. La transacció autònoma ha de finalitzar confirmant o desestimant explícitament la transacció, amb un COMMIT o un ROLLBACK, abans de retornar el control a la transacció principal.

L'utilització de transaccions autònomes, especialment si s'utilitzen de forma niuada, pot provocar l'assoliment del número màxim de transaccions obertes especificat al INIT.ORA. En el nostre cas, el sistema de log utilitza una única transacció, però caldrà tenir en compte que cada sistema operacional necessitarà una transacció més a les que utilitzi el propi sistema, per el sistema de log. Caldrà doncs, configurar correctament el INIT.ORA amb el número de transaccions concurrents que poden haver a la base de dades, però caldrà tenir en compte que incrementar el valor de la variable TRANSACTIONS comporta un increment de la grandària de la SGA i del nombre de segments de ROLLBACK ubicats a memòria (allocated) que caldrà considerar.

D'una altra banda, donat que he decidit que es podrà emmagatzemar qualsevol variable PL/SQL, cal dotar a l'API de funcions per poder subministrar-les des de el sistema transaccional. Evidentment, la solució més fàcil és de proporcionar funcions diferents per a cada tipus de dades, però això també pot ser problemàtic per el codificador. Una de les premisses que m'he proposat en el disseny és aconseguir una interfície clara, senzilla i ergonòmica, i haver de conèixer diferents noms de funcions per cada tipus de dades no és un bon començament. La solució és l'ús de la **sobrecàrrega de funcions**. Defineixo una única funció sobrecarregada per a cada tipus de variable PL/SQL que converteix a ANYDATA la dada proporcionada i l'emmagatzema. Haig de contemplar variables numèriques, caràcter, objectes, dates i booleans.

Les funcions de conversió a ANYDATA són les següents :

```
STATIC FUNCTION ConvertNumber(num IN NUMBER) RETURN AnyData,  
STATIC FUNCTION ConvertDate(dat IN DATE) RETURN AnyData,  
STATIC FUNCTION ConvertChar(c IN CHAR) RETURN AnyData,  
STATIC FUNCTION ConvertVarchar(c IN VARCHAR) RETURN AnyData,  
STATIC FUNCTION ConvertVarchar2(c IN VARCHAR2) RETURN AnyData,  
STATIC FUNCTION ConvertRaw(r IN RAW) RETURN AnyData,  
STATIC FUNCTION ConvertBlob(b IN BLOB) RETURN AnyData,  
STATIC FUNCTION ConvertClob(c IN CLOB) RETURN AnyData,  
STATIC FUNCTION ConvertBfile(b IN BFILE) RETURN AnyData,
```

La meua intenció és definir una funció a l'API per a cada una d'elles, i com no hi ha funció de conversió per a les variables boleanes, utilitzar la funció de conversió a numèric amb els valors 0 i 1.

Com que la sobrecàrrega de funcions te algunes restriccions entre elles que no es poden sobrecarregar funcions amb paràmetres de diferents tipus dins de la mateixa família i CHAR, VARCHAR i VARCHAR2 són de la mateixa família només defineixo l'últim. El problema sorgeix amb el RAW. Encara que en un primer moment em sorprèn que hi hagi conflicte d'interpretació amb el VARCHAR2, potser em semblaria més proper a un objecte, realment el RAW és semblant a un VARCHAR2 exceptuant en el fet que PL/SQL no pot interpretar-ho. Cal doncs, eliminar-ho ja que no és necessari.

Encara tindrè una altre sorpresa. Els TIMESTAMP i INTERVALs els tracta com DATE o caràcters i per tant hi ha un cop més conflictes d'interpretació. Per solucionar-ho el millor és eliminar també la funció de tipus DATE i utilitzar les conversions implícites per a guardar-ho en un VARCHAR2.

Les funcions que caldrà definir finalment són :

```
FUNCTION AfegirObservacio ( nom VARCHAR2, tipus VARCHAR2, valor NUMBER) RETURN NUMBER;  
FUNCTION AfegirObservacio ( nom VARCHAR2, tipus VARCHAR2, valor VARCHAR2) RETURN NUMBER;  
FUNCTION AfegirObservacio ( nom VARCHAR2, tipus VARCHAR2, valor BLOB) RETURN NUMBER;  
FUNCTION AfegirObservacio ( nom VARCHAR2, tipus VARCHAR2, valor CLOB) RETURN NUMBER;  
FUNCTION AfegirObservacio ( nom VARCHAR2, tipus VARCHAR2, valor BFILE) RETURN NUMBER;  
FUNCTION AfegirObservacio ( nom VARCHAR2, tipus VARCHAR2, valor BOOLEAN) RETURN NUMBER;
```

El funcionament del sistema serà el següent :

- El sistema transaccional enregistra l'inici i el fi de cada bloc o funció mitjançant les crides a les funcions *Inici* i *Fi*.
- El sistema transaccional també podrà enregistrar altres punts d'interès mitjançant la crida a la funció *AfegirTraça*.
- El sistema transaccional enregistrarà observacions a emmagatzemar amb la traça mitjançant la crida a la funció *AfegirObservacio*.

Pel que fa a les dades globals necessàries en el package, es disposarà d'una taula PL que implementarà una pila on es mantindran els identificadors dels procediments que s'estan executant en un moment determinat i un comptador del nombre d'instruccions que s'han executat dins de cada bloc i que ens proporcionarà l'atribut que implementa la relació d'ordre.

També caldrà una variable del tipus TP_TB_OBSERVACIONS on anar emmagatzemant les observacions que caldrà a associar a una traça.

Les funcions principals de que es compona el packages són :

PROCEDURE *AfegirObservacio* (*funció sobrecarregada*)

PARÀMETRES ENTRADA

Nom : Nom del paràmetre a emmagatzemar
Tipus : Tipus de paràmetre a emmagatzemar (entrada, sortida, valor de retorn, variable local).
Valor : valor que pren el paràmetre a emmagatzemar

DESCRIPCIÓ

Emmagatzema el paràmetre proporcionat a una taula local de tipus TP_TB_OBSERVACIONS. Utilitzarà els packages DBMS_ANYDATA i DBMS_TYPE per tal de codificar els tipus de variable.

PROCEDURE *Inici* (*transacció autònoma*)

PARÀMETRES ENTRADA

Nom : Nom del bloc a emmagatzemar.
Categoria DEFAULT NULL : Categoria del bloc
Comentari DEFAULT NULL : Comentari associat al inici de bloc
Usuari DEFAULT NULL : Usuari que ha activat la transacció.
(només cal proporcionar-lo a la crida que correspongui al inici de transacció).

DESCRIPCIÓ

Obté de la seqüència SEQ_ID un nou identificador i l'emmagatzema a la taula PL que implementa la jerarquia e incrementa l'apuntador. Inserta dues tuples a la taula TRAÇA, una de tipus **Bloc** i un altre d' **Inici** procés. Associa a la TRAÇA d'Inici procés les variables que es troben emmagatzemades fins el moment a la variable local de tipus TP_TB_OBSERVACIONS i la inicialitza.

PROCEDURE *Fi* (*transacció autònoma*)

PARÀMETRES ENTRADA

Resultat : Indicador de finalització del bloc
(Correcte, Incorrecta o amb Avís).
ErrorOracle DEFAULT NULL : Error oracle produït en el bloc.
Error DEFAULT NULL : Codi d'error intern del programa en execució.
Comentari DEFAULT NULL : Comentari associat al fi de bloc

DESCRIPCIO

Decrementa l'apuntador de la taula PL i inserta una tupla a la taula TRAÇA de tipus **Fi** procés i actualitza la fila de tipus **Bloc** per actualitzar la data de fi i el resultat del bloc. Associa a la TRAÇA de fi les variables que es troben emmagatzemades fins el moment a la variable local de tipus TP_TB_OBSERVACIONS i la inicialitza.

PROCEDURE AfegirTraça (transacció autònoma)

PARÀMETRES ENTRADA

Comentari : Text lliure a guardar amb la traça
ErrorOracle DEFAULT NULL : Error oracle produït en el bloc.
Error DEFAULT NULL : Codi d'error intern del programa en execució.

DESCRIPCIÓ

Inserta una tupla a la taula de TRAÇA de tipus Informativa i li associa les variables que es troben emmagatzemades fins el moment a la variable local de tipus TB_TB_OBSERVACIONS i la inicialitza.

4.2.2 El package UTL_VIS_LOG.

Les funcions principals de que es compona el packages són :

FUNCTION VeureTransaccions

PARÀMETRES ENTRADA

Nom DEFAULT NULL : Nom de la transacció a localitzar
Usuari DEFAULT NULL : Usuari de la transacció a localitzar
DataInici DEFAULT NULL : Data d'inici de l'execució de la transacció
DataFi DEFAULT NULL : Data de finalització de l'execució de la transacció
Resultat DEFAULT NULL : Resultat

VALOR DE RETORN

Retorna una taula amb les transaccions localitzades.

DESCRIPCIÓ

Localitza les transaccions que compleixen les condicions expressades per els paràmetres proporcionats.

FUNCTION ArbreExecucio

PARÀMETRES ENTRADA

IdTransaccio : Identificador intern de la transacció de la que es vol obtenir l'arbre d'execució.

VALOR DE RETORN

Retorna una taula amb l'arbre d'execució de la transacció proporcionada.

DESCRIPCIÓ

Obté la informació per construir l'arbre d'execució a partir de l'identificador de la transacció. Obté les dades de la taula TRAÇA utilitzant la clàusula CONNECT BY.

FUNCTION DetallBloc

PARÀMETRES ENTRADA

IdBloc : Identificador intern del bloc del que es vol obtenir informació.

VALOR DE RETORN

Retorna una taula amb les dades del bloc.

DESCRIPCIÓ

Obté de la taula TRAÇA tota la informació relativa al bloc sol·licitat.

FUNCTION VeureObservacions

PARÀMETRES ENTRADA

IdTraça : Identificador intern de la traça de la que es volen obtenir les observacions emmagatzemades.

VALOR DE RETORN

Retorna una taula amb les observacions de la traça.

DESCRIPCIÓ

Obté de la taula TRAÇA totes les observacions de la traça.

FUNCTION ObtenirConfiguracio

VALOR DE RETORN

Retorna una taula amb les dades de configuració.

DESCRIPCIÓ

Obté les dades actuals de la taula de CONFIGURACIÓ.

PROCEDURE ActualitzarConfiguracio

PARÀMETRES ENTRADA

Informatives : Indicador que defineix la restricció d'emmagatzemar les traces d'aquest tipus.
Profunditat : Profunditat del arbre d'execució que el sistema de log permet emmagatzemar.
Paràmetres : Indicador que defineix la restricció d'emmagatzemar els paràmetres d'E/S.
Valor retorn : Indicador que defineix la restricció d'emmagatzemar els valors de retorn.
Locals : Indicador que defineix la restricció d'emmagatzemar les variables locals.
Escalars : Indicador que defineix la restricció d'emmagatzemar les observacions escalars.
Lobs : Indicador que defineix la restricció d'emmagatzemar les observacions LOBs.
Grandària : Màxim valor en bytes que pot ocupar un valor a emmagatzemar.
Caducitat : Nombre de dies que es manté una transacció en el sistema.

DESCRIPCIÓ

Emmagatzema les dades proporcionades a la taula de CONFIGURACIO.

FUNCTION *ObtenirEstadística*

PARÀMETRES ENTRADA

Estadística	: Estadística sol·licitada
	1. Procediments més utilitzats
	2. Procediments més utilitzats per categories
	3. Procediments més utilitzats per franges horàries
	4. Procediments més utilitzats per usuari
	5. Procediments agrupats per resultat
	6. Franges horàries més utilitzades
	7. Franges horàries més utilitzades per usuari
Usuari	DEFAULT NULL : Usuari de la transacció que intervindran a l'estadística.
Resultat	DEFAULT NULL : Resultat obtingut per les transaccions a intervenir a l'estadística
DataInici	DEFAULT NULL : Data d'inici de l'execució de la transacció
DataFi	DEFAULT NULL : Data de finalització de l'execució de la transacció

VALOR DE RETORN

Retorna una taula amb les dades obtingudes.

DESCRIPCIÓ

Obté l'estadística considerant únicament les transaccions que compleixen les condicions expressades per els paràmetres proporcionats.

4.2.3 El procediment UTL_DEL_LOG.

Serà un procediment sense cap paràmetre d'entrada que consultarà la columna caducitat de la taula CONFIGURACIO i esborrarà totes les traces jerarquitzaes a les transaccions caducades, és a dir, totes aquelles on la data resultant d'afegir el nombre de dies definit a la taula de configuració a la data de finalització de la transacció, sigui anterior a la data d'execució del procés d'esborrat.

4.3 Aplicatiu d'explotació de les dades de LOG

La primera decisió que cal prendre, tot i que encara no és estrictament necessari, és l'elecció del llenguatge de programació. Inicialment i donat que es demanava un aplicatiu client-servidor, la selecció va quedar limitada a escollir entre l'entorn de programació de Microsoft (Vbasic, VC++) o l'entorn Java, però, arribats a aquest punt, la tria no ha estat simple ja que no existien requisits funcionals ni restriccions de l'entorn d'explotació que prioritzessin la selecció d'una opció en vers l'altre. No obstant, vaig triar JAVA perquè és un llenguatge més clarament orientat a objectes, perquè em donava la oportunitat de practicar la utilització de les classes de presentació (swing) i perquè la U.O.C. sempre ha potenciat el seu ús.

Un cop escollit el llenguatge de programació, passo a la tasca de disseny pròpiament dita, identificant les classes a implementar, descrivint els seus atributs i els seus mètodes. Per fer-ho, parteixo dels diagrames de col·laboració simplificats del casos d'ús elaborats a la etapa d'anàlisi on es poden observar les classes frontera que són el punt de partida del disseny de la interfície d'usuari, les classes d'entitat que són la base per dissenyar la gestió de la persistència i les classes de control.

4.3.1 Classes d'entitat.

Tal com es pot veure als diagrames de casos d'ús, es poden identificar cinc classes principals (Transaccions, Arbre d'execució, Traces, Observacions i Estadístiques) i dues classes d'utilitat (Configuració i Eliminació).

Pel que fa a les principals, tenen dues característiques comunes derivades del fet que aquest és un aplicatiu de consulta. La primera és que no es defineixen operacions d'incorporació o actualització i que les operacions d'eliminació es concentren en un

procediment extern a la interacció amb l'usuari. La segona, encara més important de cara al disseny, és que les operacions de consulta no són individuals, és a dir, que es presenten en forma de col·leccions. Per exemple, el formulari de transaccions mostra les que acompleixen les condicions indicades per l'usuari i el d'observacions mostra les associades a una traça.

Aquesta estratègia condiona la definició de les classes principals de forma que, en realitat, implementen una taula que conté la col·lecció que es presentarà. De fet, reaprofitant les classes definides a JAVA i per facilitar la utilització de les classes de presentació (swing), les classes principals, exceptuant la que implementa l'arbre d'execució, estenen i implementen la classe *'javax.swing.table.AbstractTableModel'*

TRANSACCIONS.

Aquesta entitat s'implementa a la classe *VisorLogRSCTransacions* i, com he indicat anteriorment, implementa una taula amb les transaccions seleccionades per l'usuari que conté el seu nom, l'usuari que l'activa, la data d'inici, la data de finalització, el resultat i la categoria. Addicionalment, també manté un vector amb els identificadors informàtics de les transaccions de la taula per tal de proporcionar-ho quan l'usuari la selecciona per obtenir el seu arbre d'execució.

Pel que fa als mètodes, es poden dividir en dos tipus :

- Derivats de les operacions indicades al diagrama de casos d'us.

La consulta de transaccions implementada en el constructor de la classe i que accedeix a la BB.DD. per obtenir la informació de les transaccions que compleixen els criteris de selecció establerts per l'usuari.

L'obtenció de l'identificador de la transacció, implementada a la funció *getIidint()*, que obté l'identificador informàtic de la transacció que ocupa una fila determinada.

- Derivats de la reutilització de la classe *AbstractTableModel*.

El mètode *getColumnCount()* que retorna el nombre de columnes de la taula.

El mètode *getRowCount()* que retorna el nombre de files de la taula.

El mètode *getValueAt()* que retorna el valor de la cel·la ubicada a una fila i columna determinades.

El mètode *isCellEditable()* que indica si l'usuari ha de poder modificar el valor de la cel·la ubicada a una fila i columna determinades.

ARBRE D'EXECUCIÓ.

Aquesta entitat s'implementa a la classe *VisorLogRSCArbreExecució* i, en aquest cas, la col·lecció no està estructurada en forma de taula sinó en forma d'arbre, utilitzant la classe *'javax.swing.tree.DefaultMutableTreeNode'*. Com en el cas anterior, també manté un vector amb els identificadors informàtics dels blocs que formen l'arbre d'execució per tal de proporcionar-ho quan l'usuari selecciona un node per obtenir les traces associades.

Pel que fa als mètodes s'identifiquen els següents :

- Derivats de les operacions indicades al diagrama de casos d'us.

La consulta de l'arbre d'execució implementada amb dos mètodes, d'una banda, el constructor de la classe i que accedeix a la BB.DD. per obtenir la informació dels blocs associats a la transacció seleccionada per l'usuari i, de l'altre, el mètode *getArrel()* que retorna l'arbre a les classes de control.

La obtenció de l'identificador de la transacció, implementada a la funció *getIidint()*, que obté l'identificador informàtic de la transacció que ocupa un node determinat.

TRACES.

Aquesta entitat s'implementa a la classe `VisorLogRSCTraces` i, com en el cas de transaccions, implementa una taula amb les traces associades a un node de l'arbre d'execució que conté el seu nom, el tipus i la categoria, l'ordre i la data d'execució, els identificadors dels errors detectats i, per últim, els comentaris que el programador vulgui incloure com aclaridors. De la mateixa manera, també es manté un vector amb els identificadors informàtics de les traces de la taula per tal de proporcionar-ho quan l'usuari la selecciona per obtenir les observacions associades.

Pel que fa als mètodes, es poden dividir en dos tipus :

- Derivats de les operacions indicades al diagrama de casos d'us.
La consulta de traces, implementada en el constructor de la classe, que accedeix a la BB.DD. per obtenir la informació de les traces associades al bloc seleccionat a l'arbre d'execució.
La obtenció de l'identificador de la traça, implementada a la funció `getIdint()`, que obté l'identificador informàtic de la traça que ocupa una fila determinada.
- Derivats de la reutilització de la classe *AbstractTableModel*.
El mètode `getColumnCount()` que retorna el nombre de columnes de la taula.
El mètode `getRowCount()` que retorna el nombre de files de la taula.
El mètode `getValueAt()` que retorna el valor de la cel·la ubicada a una fila i columna determinades.
El mètode `isCellEditable()` que indica si l'usuari ha de poder modificar el valor de la cel·la ubicada a una fila i columna determinades.

OBSERVACIONS.

Aquesta entitat s'implementa a la classe `VisorLogRSCObservacions` i, com en casos anteriors, implementa una taula amb les observacions associades a una traça que conté el seu nom, el tipus i el valor de la variable.

Pel que fa als mètodes, es poden dividir en dos tipus :

- Derivats de les operacions indicades al diagrama de casos d'us.
La consulta d'observacions implementada en el constructor de la classe i que accedeix a la BB.DD. per obtenir la informació de les observacions associades a una traça.
- Derivats de la reutilització de la classe *AbstractTableModel*.
El mètode `getColumnCount()` que retorna el nombre de columnes de la taula.
El mètode `getRowCount()` que retorna el nombre de files de la taula.
El mètode `getValueAt()` que retorna el valor de la cel·la ubicada a una fila i columna determinades.
El mètode `isCellEditable()` que indica si l'usuari ha de poder modificar el valor de la cel·la ubicada a una fila i columna determinades.

ESTADÍSTIQUES.

Aquesta entitat s'implementa a la classe `VisorLogRSEstadistiques` i, com en casos anteriors, implementa una taula amb els valors estadístics d'utilització de transaccions que ha seleccionat i configurat l'usuari.

Pel que fa als mètodes, es poden dividir en dos tipus :

- Derivats de les operacions indicades al diagrama de casos d'us.
La consulta d'estadístiques implementada en el constructor de la classe i que accedeix a la BB.DD. per obtenir la informació de les estadístiques d'utilització de transaccions.
- Derivats de la reutilització de la classe *AbstractTableModel*.
El mètode getColumnCount() que retorna el nombre de columnes de la taula.
El mètode getRowCount() que retorna el nombre de files de la taula.
El mètode getValueAt() que retorna el valor de la cel·la ubicada a una fila i columna determinades.
El mètode isCellEditable() que indica si l'usuari ha de poder modificar el valor de la cel·la ubicada a una fila i columna determinades.

La resta de classes d'entitat són d'utilitat i implementen les funcionalitats de parametrització i eliminació de transaccions.

CONFIGURACIÓ.

Aquesta entitat s'implementa a la classe VisorLogRSCConfiguracio i manté la informació dels valors parametritzables de l'aplicatiu :

- Indicadors d'enregistrabilitat per tipus de traça (traces informatives).
- Indicadors d'enregistrabilitat per tipus de variable (paràmetres, valors de retorn, variables locals).
- Indicadors d'enregistrabilitat per tipus de dada (escalars, objectes).
- Límit d'enregistrabilitat de la profunditat de l'arbre d'execució.
- Límit d'enregistrabilitat de la grandària de la variable.
- Límit de permanència en el sistema d'una transacció i les seves dades associades.

Pel que fa als mètodes s'identifiquen dos:

- La consulta dels valors de parametrització, implementada amb dos mètodes, d'una banda el constructor de la classe que accedeix a la BB.DD. per obtenir els valors de parametrització i, de l'altre el mètode obtenirConfiguració() que retorna el paràmetres a les classes de control.
- L'actualització dels valors de parametrització, implementada amb el mètode actualitzarConfiguracio() que fa persistents les modificacions realitzades per l'usuari.

ELIMINACIÓ.

- Aquesta entitat s'implementa a la classe VisorLogRSCBorrat i només manté la data límit de persistència en el sistema de les transaccions.

Pel que fa als mètodes s'identifiquen dos:

- La consulta de la data límit de persistència, implementada amb dos mètodes, d'una banda el constructor de la classe que accedeix a la BB.DD. per obtenir la data i, de l'altre el mètode obtenirData() que retorna la retorna a les classes de control.
- L'eliminació de transaccions, implementada en el mètode borrarTransaccions(), que esborra les caducades i, en cascada, les traces i observacions associades.

4.3.2 Classes frontera.

Al capítol d'anàlisi ja es descriu detalladament l'aparença de la interfície gràfica amb l'usuari i per tant, es pot extreure, fàcilment, un relatori de les dades que cal mantenir a les classes frontera. Es per això, que en aquest apartat es passa per sobre la definició d'atributs i la descripció dels mètodes de creació dels formularis i es centra en l'identificació i descripció dels mètodes que gestionen la interacció.

MENÚ.

Aquesta interfície s'implementa amb la classe `VisorLogGUIMenu`, s'estén de `'javax.swing.JMenuBar'` i no es defineixen més mètodes que el constructor que mostra les opcions de menú.

TRANSACCIONS.

Aquesta interfície s'implementa amb la classe `VisorLogGUISeguiment`, s'estén de `'javax.swing.JPanel'` i les seves operacions específiques són :

- L'obtenció del criteris de selecció de transaccions, implementada al mètode `obtenirFiltre()` que retorna els valors d'aquest criteris a les classes de control.
- La presentació de la col·lecció de transaccions, implementada als mètodes `presentarTaula()` i `netejarTaula()` que mostren, respectivament, les transaccions seleccionades i una taula buida quan es detecta un error o no existeixen instàncies que compleixin els requisits.
- L'obtenció de la transacció seleccionada, implementada al mètode `obtenirSeleccio()` i que obté la posició de la transacció seleccionada per l'usuari.
- La notificació d'errors, implementada al mètode `error()`.

TRACES.

Aquesta interfície s'implementa amb la classe `VisorLogGUITraces`, s'estén de `'javax.swing.JPanel'` i les seves operacions específiques són :

- La presentació de la transacció seleccionada, implementada al mètode `presentarTransaccio()` i que permet veure la informació de transacció un cop es canvia la interfície de transaccions per la de traces.
- La presentació de l'arbre d'execució, implementada al mètode `presentarArbre()` i que permet visualitzar els blocs associats a la transacció seleccionada.
- La presentació de la col·lecció de traces, implementada als mètodes `presentarTaula()` i `netejarTaula()` que mostren, respectivament, les traces associades al bloc seleccionat i una taula buida quan es detecta un error o no existeixen instàncies que compleixin els requisits.
- La obtenció de la traça seleccionada, implementada al mètode `obtenirSeleccio()` i que obté la posició de la traça seleccionada per l'usuari.
- La notificació d'errors, implementada al mètode `error()`.

OBSERVACIONS.

Aquesta interfície s'implementa amb la classe `VisorLogGUIObservacions`, s'estén de `'javax.swing.JPanel'` i les seves operacions específiques són :

- La presentació de la traça seleccionada, implementada al mètode `presentarTraca()` i que permet veure la informació de traça un cop es canvia la interfície de traces per la d'observacions.

- La presentació de la col·lecció d'observacions, implementada al mètode `presentarTaula()` que mostra, les observacions associades a la traça seleccionada.
- La notificació d'errors, implementada al mètode `error()`.

ESTADÍSTIQUES.

Aquesta interfície s'implementa amb la classe `VisorLogGUIEstadistiques`, s'estén de `'javax.swing.JPanel'` i les seves operacions específiques són :

- La obtenció del criteris de selecció de transaccions i configuració de l'informe, implementada al mètode `obtenirFiltre()` que retorna els valors d'aquest criteris a les classes de control.
- La presentació de la col·lecció de valors estadístics, implementada als mètodes `presentarTaula()` i `netejarTaula()` que mostren, respectivament, els informes obtinguts o una taula buida quan es detecta un error o no existeixen instàncies que compleixin els requisits.
- La notificació d'errors, implementada al mètode `error()`.

CONFIGURACIÓ.

Aquesta interfície s'implementa amb la classe `VisorLogGUIConfiguracio`, s'estén de `'javax.swing.JPanel'` i les seves operacions específiques són :

- La presentació de les dades de parametrització, implementada al mètode `presentarDades()` mostra els valors de la configuració del sistema.
- La obtenció de les dades de parametrització, implementada al mètode `obtenirDades` que retorna els valors introduïts per l'usuari() a les classes de control.
- La notificació d'errors, implementada al mètode `error()`.

ELIMINACIÓ.

Aquesta interfície s'implementa amb la classe `VisorLogGUIBorrat`, s'estén de `'javax.swing.JPanel'` i les seves operacions específiques són :

- La sol·licitud d'autorització per eliminar les transaccions fins a una data, `demanarAutoritzacio()` que ha de mostrar la data límit i demanar la confirmació per continuar amb el procés.
- La notificació de procés finalitzat correctament, implementada al mètode `procesFinalitzat` que adverteix al usuari de la finalització correcta del procés d'eliminació.
- La notificació d'errors, implementada al mètode `error()`.

4.3.3 Classes de control.

Dins d'aquesta categoria s'inclouen tres classes, la primera és la principal del sistema i les altres dues estan lligades, respectivament, al control de la navegació per la interfície d'usuari i al control de la connexió amb la BB.DD. Donada la 'simplicitat' del sistema pot semblar una divisió excessiva però s'adopta aquesta estratègia ni que sigui, només, per tenir un codi més estructurat.

CONTROL de NAVEGACIÓ.

Aquesta és la classe que gestiona la interacció de l'usuari amb els formulari del sistema i controla la navegació entre ells. Conseqüentment, també gestiona la informació que es presenta i la que introdueix l'usuari. Està implementada per la classe `VisorLogGUINavegació`

i conté una instància de cadascuna de les classes frontera de las que rep i a les que envia informació.

Pel que fa als mètodes, en aquesta classe s'implementen tres interfícies :

- WindowListener que permet el control de la finestra de l'aplicatiu i implica la implementació de diversos mètodes, dels qual només és rellevant, windowClosing() que controla el tancament de la finestra i, consegüentment de l'aplicatiu.
- ActionListener que permet, mitjançant el mètode actionPerformed(), el control de les opcions de la interfície de connexió (Acceptar, Cancel·lar), de les opcions de menú (Seguiment, Estadístiques, Configuració, Borrat, Ajuda i Sortir), així com de les diferents barres d'eines, la de la interfície de transaccions (Aplicar filtre i Veure les traces associades), de la interfície de traces (Tornar a transaccions, Veure observacions), de la interfície d'observacions (Tornar a traces), de la d'estadístiques (Aplicar filtre) i, per últim, de la de configuració (Actualitzar configuració).
- TreeSelectionListener que permet, mitjançant el mètode valueChanged(), identificar el node seleccionat de l'arbre d'execució.

CONTROL de CONNEXIÓ.

Aquesta és la classe que gestiona la interacció del sistema amb la BB.DD. Està implementada a la classe VisorLogRSCConnexió i conté un identificador de la connexió amb la BB.DD. i una instància de cadascuna de les classes d'entitat.

Pel que fa als mètodes, es poden dividir en tres categories :

- Gestió de la connexió amb la BB.DD. implementada amb dos mètodes, d'una banda el constructor de la classe que estableix la connexió i de l'altra el mètode tancar() que la dona per finalitzada.
- Gestió de la informació de les classes principals, implementada amb els mètodes obtenirTransaccions() que retorna una col·lecció de transaccions que compleixen uns criteris, obtenirDadesTransaccio() que retorna la informació relativa a una transacció, obtenirArbreExecució() que retorna l'arbre de blocs d'execució associats a una transacció, obtenirTraces() que retorna una col·lecció de les traces associades a un node de l'arbre d'execució, obtenirDadesTraca() que retorna la informació relativa a una traça determinada, obtenirObservacions() que retorna una col·lecció de les observacions associades a una traça i, per últim, obtenirEstadistiques() que retorna una col·lecció del valors estadístics d'utilització de les transaccions que compleixen uns criteris.
- Gestió de la informació de les classes d'utilitat, implementada amb els mètodes obtenirConfiguracio() i actualitzarConfiguració() que obté i actualitza, respectivament, els valors de parametrizació del sistema. També inclou els mètodes obtenirBorrat() i borrarTransaccions() que permeten activar el procés d'eliminació de transaccions caducades.

CONTROL del SISTEMA.

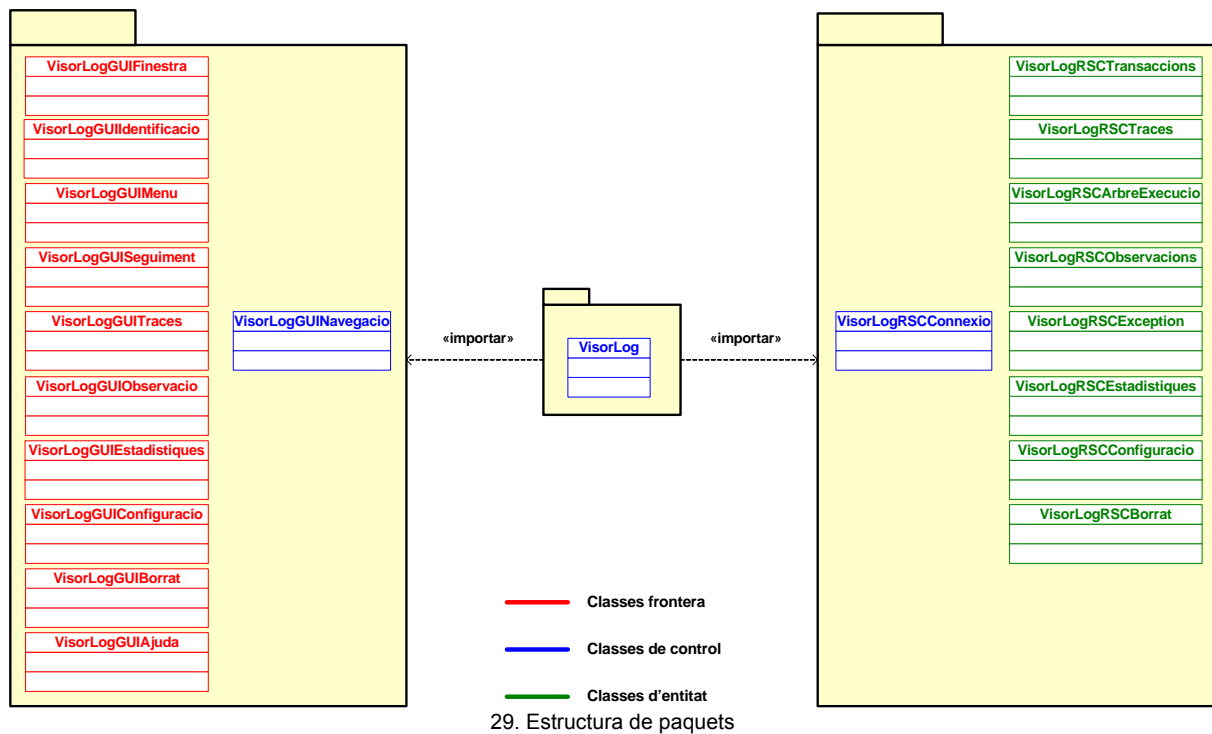
Aquesta és la classe que permet activar l'aplicatiu i que en sistemes més complexos implementaria les regles del negoci. En aquest cas només fa d'interfície entre les classes de presentació i les d'interacció amb la BB.DD. A tal efecte, manté una instància de les classes de control esmentades anteriorment i pel que fa als mètodes implementa els necessaris per gestionar la informació que circula entre la classe de control de navegació i de connexió :

- connectar que obre una connexió per tal d'obtenir la informació emmagatzemada a BB.DD.

- sortir que Tanca la connexió amb la BB.DD. i surt de l'aplicatiu.
- obtenirTransaccions que obté les transaccions que compleixen les condicions indicades per l'usuari.
- obtenirDadesTransaccio que obté els valors dels atributs d'una transacció.
- obtenirArbreExecucio que obté l'arbre d'execució d'una transacció. Cadascun dels nodes representa una funció o, millor dit, un bloc.
- obtenirTraces que obté les traces enregistrades per un node de l'arbre d'execució.
- obtenirDadesTraca que obté els valors dels atributs d'una traça.
- obtenirObservacions que obté les observacions enregistrades per una traça.
- obtenirEstadistiques que obté l'informe estadístic seleccionat.
- obtenirConfiguracio que obté els valors actuals de parametrització de l'aplicatiu.
- actualitzarConfiguracio que canvia els valors actuals de parametrització de l'aplicatiu.
- abtenirBorrat que obté la data fins la que s'eliminaran les transaccions enregistrades.
- borrarTransaccions que elimina les transaccions executades abans d'una data que es pot parametritzar dins de l'aplicatiu.

4.3.4 Paquets d'agrupació de classes.

Tal com es por veure al diagrama següent, les classes han estat agrupades en tres paquets, un per les classes d'interacció amb l'usuari, un altre per la interacció amb la BB.DD. i, per últim, una altre per la classe d'activació de l'aplicatiu.



4.4 Altres possibilitats a explorar. Ús de ALL_ARGUMENTS i DBMS_DESCRIBE.

Pel que fa a la definició de la metodologia de programació, en un primer moment el meu propòsit era fer totalment transparent l'ús del seguiment al programador del sistema transaccional, de forma que no li calgués indicar quina és la funció que s'està executant o els paràmetres amb que s'està executant.

En concret he mirat la vista ALL_ARGUMENTS i el package DBMS_DESCRIBE per tal de poder obtenir la informació sobre el procediment i cada un dels paràmetres.

Per tal de poder obtenir informació sobre la funció que s'està executant caldria explorar el package DBMS_DEBUG.

Donat a que aquesta part podria ser per si mateix un projecte he deixat de banda aquesta possibilitat, encara que crec que podria resultar força interessant i es podria plantejar com ampliació del projecte. Caldria però anar amb especial cura amb el rendiment.

4.5 Definició de proves unitàries.

4.5.1 UTL_REG_LOG

Codi	Descripció
LOGU_R001	<p>Emmagatzemar una transacció. Cal comprovar que es guarden correctament els paràmetres.</p> <pre> DECLARE return NUMBER := 0; BEGIN -- Inici transacció return := UTL_REG_LOG.Inici ('LOGU_R001', 'Regla Negoci', NULL, 'USU1'); IF return != 0 THEN dbms_output.put_line (return '-' UTL_REG_LOG.ObtenirMissatge); END IF; -- Fi transacció return := UTL_REG_LOG.Fi ('C'); IF return != 0 THEN dbms_output.put_line (return '-' UTL_REG_LOG.ObtenirMissatge); END IF; COMMIT; END; / </pre> <p>A la taula traça s'han d'haver emmagatzemat tres tuples una de tipus bloc que correspon a la transacció, i una entrada i una sortida depenen d'ella.</p>
LOGU_R002	<p>Emmagatzemar una traça niuada.</p> <pre> DECLARE return NUMBER := 0; BEGIN return := UTL_REG_LOG.Inici ('LOGU_R002', 'Regla Negoci', NULL, 'USU1'); IF return != 0 THEN dbms_output.put_line (return '-' UTL_REG_LOG.ObtenirMissatge); END IF; return := UTL_REG_LOG.Inici ('Funcio1'); IF return != 0 THEN dbms_output.put_line (return '-' UTL_REG_LOG.ObtenirMissatge); END IF; return := UTL_REG_LOG.Fi ('C'); IF return != 0 THEN dbms_output.put_line (return '-' UTL_REG_LOG.ObtenirMissatge); END IF; return := UTL_REG_LOG.Fi ('C'); IF return != 0 THEN dbms_output.put_line (return '-' UTL_REG_LOG.ObtenirMissatge); END IF; </pre>

Codi	Descripció
	<pre> COMMIT; END; / </pre> <p>A la taula traça s'han d'haver emmagatzemat sis tuples una de tipus bloc que correspon a la transacció amb una entrada, un bloc més i una sortida. El bloc niat tindrà a més una entrada i una sortida.</p>
LOGU_R003	<p>Emmagatzemar traces informatives.</p> <pre> DECLARE retorn NUMBER := 0; BEGIN retorn := UTL_REG_LOG.Inici ('LOGU_R003', 'Regla Negoci', NULL, 'USU1'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; retorn := UTL_REG_LOG.AfegirTraça ('Dades a afegir'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; retorn := UTL_REG_LOG.Fi ('C'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; COMMIT; END; / </pre> <p>A la taula traça s'han d'haver emmagatzemat quatre tuples una de tipus bloc que correspon a la transacció amb una entrada, una traça informativa i una sortida.</p>
LOGU_R004	<p>Emmagatzemar observacions a una traça. S'afegeixen paràmetres d'entrada a l'inici funció. Una variable local a la traça informativa. Valors de retorn a la traça de sortida.</p> <pre> DECLARE retorn NUMBER := 0; BEGIN retorn := UTL_REG_LOG.Inici ('LOGU_R004', 'Regla Negoci', NULL, 'USU1'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; retorn := UTL_REG_LOG.AfegirObservacio('Var1',UTL_REG_LOG.Entrada, 1); retorn := UTL_REG_LOG.Inici ('Funcio1'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; retorn := UTL_REG_LOG.AfegirObservacio('Var2',UTL_REG_LOG.local, 'U'); retorn := UTL_REG_LOG.AfegirTraça ('Dades a afegir'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; retorn :=UTL_REG_LOG.AfegirObservacio('Var3',UTL_REG_LOG.retorn, true); retorn := UTL_REG_LOG.Fi ('C'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; retorn := UTL_REG_LOG.Fi ('C'); IF retorn != 0 THEN dbms_output.put_line (retorn '-' UTL_REG_LOG.ObtenirMissatge); END IF; COMMIT; END; / </pre> <p>S'han d'haver emmagatzemat les observacions corresponents a cada una de les traces.</p>
LOGU_R005	<p>Emmagatzemar qualsevol tipus d'observació a una traça. Cal provar d'emmagatzemar qualsevol tipus de variable PL/SQL. Fitxer : Test_variables.sql</p>
LOGU_R006	<p>Configurabilitat. Provar que el sistema de log emmagatzema la informació segons indica la configuració actual. Cal provar cada variable de configuració. Fitxer test_config.sql</p>

30. Taula de definició de proves unitàries del UTL_REG_LOG

4.5.2 UTL_VIS_LOG

Codi	Descripció
LOGU_V001	veureTransaccions : Executant la funció sense paràmetres cal obtenir totes les transaccions emmagatzemades exec utl_vis_log.veureTransaccions
LOGU_V002	veureTransaccions : Obtenir les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.veureTransaccions (<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V003	arbreExecucio : Obtenir l'arbre d'execució d'una transacció (cal conèixer l'identificador). exec utl_vis_log.arbreExecucio (<idTransaccio>)
LOGU_V004	detallBloc : Obtenir les traces associades a un bloc (cal passar-li l'identificador del bloc (node de l'arbre)). exec utl_vis_log.detallBloc (<idBloc>)
LOGU_V005	veureObservacions : Obtenir les observacions associades a una traça (cal conèixer l'identificador de la traça). exec utl_vis_log.veureObservacions (<idTraça>)
LOGU_V006	obtenirEstadistica : Obtenir estadística procediments més utilitzats exec utl_vis_log.obtenirEstadistica (1)
LOGU_V007	obtenirEstadistica : Obtenir estadística procediments més utilitzats utilitzant només les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.obtenirEstadistica (1 ,<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V008	obtenirEstadistica : Obtenir estadística procediments més utilitzats per categories exec utl_vis_log.obtenirEstadistica (2)
LOGU_V009	obtenirEstadistica : Obtenir estadística procediments més utilitzats per categories utilitzant només les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.obtenirEstadistica (2 ,<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V010	obtenirEstadistica : Obtenir estadística procediments més utilitzats per franges horàries exec utl_vis_log.obtenirEstadistica (3)
LOGU_V011	obtenirEstadistica : Obtenir estadística procediments més utilitzats per franges horàries utilitzant només les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.obtenirEstadistica (3 ,<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V012	obtenirEstadistica : Obtenir estadística procediments més utilitzats per usuari exec utl_vis_log.obtenirEstadistica (4)
LOGU_V013	obtenirEstadistica : Obtenir estadística procediments més utilitzats per usuari utilitzant només les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.obtenirEstadistica (4 ,<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V014	obtenirEstadistica : Obtenir estadística procediments agrupats per resultat exec utl_vis_log.obtenirEstadistica (5)
LOGU_V015	obtenirEstadistica : Obtenir estadística procediments agrupats per resultat utilitzant només les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.obtenirEstadistica (5 ,<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V016	obtenirEstadistica : Obtenir estadística de franges horàries més utilitzades

Codi	Descripció
	exec utl_vis_log.obtenirEstadistica (6)
LOGU_V017	obtenirEstadistica : Obtenir estadística franges horàries més utilitzades utilitzant només les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.obtenirEstadistica (6 ,<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V018	obtenirEstadistica : Obtenir estadística franges horàries més utilitzades per usuari exec utl_vis_log.obtenirEstadistica (7)
LOGU_V019	obtenirEstadistica : Obtenir estadística franges horàries més utilitzades per usuari utilitzant només les transaccions que compleixin el filtre (provar els filtres per nom, usuari, data (inici - fi) i resultat). exec utl_vis_log.obtenirEstadistica (7 ,<nom>, <usuari>, <inici>, <fi>, <resultat>)
LOGU_V020	VeureUnaTransaccio : obtenir totes les dades d'una transacció exec utl_vis_log.veureUnaTransaccio (<idTransaccio>)
LOGU_V021	VeureUnaTransaccio : obtenir totes les dades d'una transacció exec utl_vis_log.veureUnaTransaccio (<nomTransaccio>)

31. Taula de definició de proves unitàries del UTL_VIS_LOG

4.5.3 UTL_DEL_LOG

Codi	Descripció
LOGU_D001	Cal validar el procés de neteja de transaccions caducades, és a dir, totes aquelles on la data resultant d'afegir el nombre de dies definit a la taula de configuració a la data de finalització de la transacció, sigui anterior a la data d'execució del procés d'esborrat. Fitxer : test_delete.sql

32. Taula de definició de proves unitàries del UTL_DEL_LOG

4.5.4 VisorLog.

Codi	Descripció
LOGU_M001	El menú del sistema de log permet accedir a les sis funcionalitats identificades, seguiment, estadístiques, configuració, eliminar, ajuda i sortir.
LOGU_S001	Des de l'opció de seguiment es veuen totes les transaccions executades, mostrant el seu nom, el de l'usuari que la ha iniciat, la categoria, les dates d'inici i finalització i el resultat obtingut. Accedir a l'opció seguiment i executar la consulta sense aplicar cap filtre. Han d'aparèixer totes les transaccions de la taula TRAÇA.
LOGU_S002	Es poden acotar les transaccions a visualitzar, filtrant per nom de transacció, usuari, data inici, data fi i resultat. Accedir a l'opció seguiment i executar la consulta omplint els atributs del filtre. Han d'aparèixer només les transaccions que el compleixen.
LOGU_S003	Es permet seleccionar una transacció per fer el seguiment del seu flux d'execució. Si ens situem sobre una transacció i prenem el botó endavant obtenim les dades de la transacció. S'ha de situar a la pantalla amb l'arbre d'execució.
LOGU_S004	Permet visualitzar el flux d'execució d'una transacció en forma d'arbre on els nodes representaran els blocs d'instruccions pels que ha transcorregut el procés. S'ha de veure l'arbre d'execució de la transacció que s'ha seleccionat. Cada branca és un bloc i mostra el resultat de finalització del bloc.
LOGU_S005	El sistema permet expandir les branques de l'arbre d'execució per visualitzar els blocs més interns. Situats sobre un node de l'arbre i amb un doble clic o premen el icona d'expansió es veuen els bloc interns.
LOGU_S006	Permet contreure les branques de l'arbre d'execució, simplificant la visió de l'operador. Situats sobre un node de l'arbre i amb un doble clic o premen l'icona de contracció es simplifica l'arbre.
LOGU_S007	Es permet seleccionar una branca de l'arbre d'execució per visualitzar la informació associada

Codi	Descripció
	al bloc de traces que representa. Cada cop que naveguem sobre un branca de l'arbre la zona de la pantalla de detall del bloc s'actualitza amb el detall de traces del bloc.
LOGU_S008	Es permet visualitzar les traces associades al bloc seleccionat, informant del ordre d'execució, del nom de la funció, la seva categoria, els comentaris, errors, el tipus i de la data i hora d'execució. S'obtenen les dades de les traces del bloc emmagatzemades a la taula TRAÇA.
LOGU_S009	Es permet seleccionar una traça per visualitzar la informació de les observacions associades. Si ens situem sobre una traça de la zona de detall del bloc i prenem el botó endavant obtenim les observacions associades a la traça. S'ha de situar a la pantalla amb les observacions.
LOGU_S010	El sistema permet visualitzar les observacions enregistrades en el moment de generació de la traça, informant del nom de la variable, del seu tipus i especialment del seu valor.
LOGU_E001	El sistema permet la obtenció d'estadístiques de procediments més utilitzats, globalment, agrupant per categories, franges horàries o per usuari. Des de l'opció d'estadístiques es poden obtenir les estadístiques de procediments més utilitzats. A la zona de la pantalla destinada a visualitzar el resultat es presenta aquests en forma tabular.
LOGU_E002	Permet la obtenció d'estadístiques de procediments agrupats pel resultat obtingut. Des de l'opció d'estadístiques es poden obtenir la estadística de procediments agrupats per resultat. A la zona de la pantalla destinada a visualitzar el resultat es presenta aquests en forma tabular.
LOGU_E003	El sistema permet la obtenció d'estadístiques de franges horàries d'utilització del sistema, globalment o agrupant per usuari. Des de l'opció d'estadístiques es poden obtenir les estadístiques de franges horàries d'utilització del sistema. A la zona de la pantalla destinada a visualitzar el resultat es presenta aquests en forma tabular.
LOGU_E004	Es permet restringir l'àmbit de l'estudi estadístic, filtrant les traces a processar per resultat, data d'execució i nom de l'usuari que va iniciar la transacció. A la zona de la pantalla destinada a visualitzar el resultat es presenta aquests en forma tabular i no han d'haver intervingut les transaccions que no compleixen el filtre.
LOGU_C001	Permet configurar els tipus de dades que s'enregistraran. Des de l'opció de menú Configuració es permet controlar l'enregistrament dels escalars i de LOBs. Si es desmarca el check box aquests tipus de dades no s'enregistren i a la taula CONFIGURACIO s'ha actualitzat el valor.
LOGU_C002	El sistema permet configurar els tipus de variables que s'enregistraran. Des de l'opció de menú configuració es permet controlar l'enregistrament de variables d'E/S, variables locals i valors de retorn. Si es desmarca el check box no s'enregistren aquests tipus de variables i a la taula CONFIGURACIO s'ha actualitzat el valor.
LOGU_C003	El sistema permet configurar la grandària màxima de les variables de tipus LOBs a enregistrar. Es permet indicar el grandària màxima de les variables des de l'opció de menú de configuració i a la taula CONFIGURACIO s'ha actualitzat el valor.
LOGU_C004	El sistema permet configurar la profunditat màxima enregistrable de l'arbre d'execució. Des de l'opció de menú configuració es pot indicar la profunditat de l'arbre i a la taula CONFIGURACIO s'ha actualitzat el valor.
LOGU_C005	El sistema permet configurar l'interval de permanència de les transaccions en el sistema. Des de l'opció de menú configuració es permet actualitzar els dies de caducitat d'una transacció per el procés de neteja i a la taula CONFIGURACIO s'ha actualitzat el valor.
LOGU_C006	El sistema permet la visualització dels valors establerts per la configuració del funcionament del sistema de log. Des de l'opció de menú configuració es visualitza el valor actual dels paràmetres de configuració emmagatzemats a la taula CONFIGURACIO.
LOGU_E008	Els sistema permet eliminar les traces obsoletes. Des de l'opció Eliminar el sistema elimina de la taula TRAÇA les transaccions anteriors a la data màxima de permanència.
LOGU_E009	El sistema mostra un missatge d'advertència mostrant la data fins a la que s'esborraran les transaccions. Des de l'opció Eliminar el mostra el missatge i es permet cancel·lar per a no continuar amb el procés d'esborrat.
LOGU_A001	Es mostra la ajuda del sistema. Des de l'opció ajuda es pot consultar el manual d'usuari de l'aplicació i es visualitza la informació sobre el VisorLog.
LOGU_F001	Des de l'opció sortir s'abandona l'aplicació.

33. Taula de definició de proves unitàries del VisorLog

5. Manual d'usuari del sistema de log genèric.

5.1 Instal·lació i desinstal·lació.

Per la instal·lació del sistema de log genèric és necessari utilitzar la *versió 9i d'Oracle* i per tal de disposar també del visor de logs, és necessari la *versió 1.4 del JDSK*.

En primer lloc cal crear l'estructura necessària per al funcionament del seguiment genèric. Per això, cal executar el script `creacio.sql` des de el SQL*Plus per a crear tota l'estructura emmagatzemada :

```
SQL> @creacio
```

NOTA : És necessari disposar de privilegis per crear seqüències, taules i packages.

Per tal d'executar el Visor de log cal extreure del zip del producte l'aplicació en un directori i assegurar-se que en el classpath hi ha les rutes necessàries, és a dir, les pròpies del java i les d'oracle apuntant al directori on es trobin els fitxers `classes*.zip` i `nls_charset*.zip`. L'aplicació es troba `\aplicacio\executables*.*`.

Cal assegurar-se que es pot compilar i executar codi JAVA.

Per activar el VisorLog cal situar-se en el directori `\aplicacio\executables` i executar la següent instrucció :

```
java VisorLog/VisorLog
```

Per desinstal·lar l'aplicació només cal eliminar els fitxers del VisorLog i per eliminar tots els objectes creats per a l'aplicació executar el script `borrat.sql` des del sql*Plus :

```
SQL> @borrat
```

5.2 Normativa de codificació del sistema de seguiment.

Les funcions del sistema de log es basen, principalment, en una selecció acurada del punts de traça i de les variables a observar en cadascun d'ells. Aquesta tria la realitza el desenvolupador del sistema operacional pel què es fa imprescindible la confecció d'una normativa o metodologia de codificació que garanteixi el correcte funcionament del sistema.

El funcionament del sistema serà el següent :

- El sistema transaccional enregistra l'inici i el fi de cada bloc o funció mitjançant les crides a les funcions *Inici* i *Fi*.
- El sistema transaccional també podrà enregistrar altres punts d'interès mitjançant la crida a la funció *AfegirTraça*.
- El sistema transaccional enregistrarà observacions a emmagatzemar amb la traça mitjançant la crida a la funció *AfegirObservacio*.

La funció *INICI* crea una tupla de tipus Bloc i una tupla de tipus Inici procés. La primera crida a la funció *INICI* es crea com a transacció, no té pare i s'emmagatzema l'usuari que l'ha iniciat. Si no es passa el paràmetre usuari s'emmagatzema l'usuari connectat, aquesta estratègia es correcta en architectures client-servidor on cada usuari pot tenir diferents comptes a la base de dades, per contra, no es gens adient en architectures web on el servidor d'aplicacions sempre es connecta amb el mateix usuari.

A partir d'aquest moment cada crida a la funció *INICI* es jerarquitzava al bloc actual. Totes les traces s'emmagatzemaven a aquests bloc i cada crida a la funció *INICI* nia un nivell més. La funció *FI* crea una tupla de fi procés i retrocedeix l'arbre un nivell.

Cada crida a la funció *AfegirObservacio* emmagatzema a una taula PL/SQL la variable proporcionada. Les funcions *INICI*, *FI* o *AfegirInformació* associen a la traça que incorporen (d'inici procés, fi procés o traça informativa) totes les observacions emmagatzemades fins el moment.

El sistema és totalment flexible per emmagatzemar informació associada o no a blocs reals del sistema transaccional. És responsabilitat del programador de triar el lloc i la informació representativa a emmagatzemar.

És necessari incloure un FI per a cada bloc iniciat amb el resultat obtingut. Si no es produeix el FI d'un bloc i s'inicia un altre el sistema de log genèric els niarà. Si finalitza la transacció del sistema transaccional i no s'han tancat amb el corresponent FI tots els blocs iniciats l'arbre d'execució serà incorrecte i no es podrà visualitzar l'estat de finalització de la transacció.

L'estructura d'un bloc PL/SQL que utilitzi el sistema de log genèric és la que segueix :

```
DECLARE
...
BEGIN
-- registrar les observacions necessàries d'inici
UTL_REG_LOG.AfegirObservacio (nom, tipus, valor);
-- indicació d'entrada a procediment
UTL_REG_LOG.Inici (nom, categoria, comentari, usuari)
...
...
-- registre d'alguna variable local d'interès
UTL_REG_LOG.AfegirObservacio (nom, tipus, valor);
-- introduir traça informativa
UTL_REG_LOG.AfegirTraça (comentari)
...
...
CRIDA FUNCIO;
...
...
-- registre del valor de retorn o variables de sortida
UTL_REG_LOG.AfegirObservacio (nom, tipus, valor);
-- indicació de sortida del procediment
UTL_REG_LOG.Fi (resultat, error_oracle, error, comentari)

EXCEPTION
WHEN ...
THEN ...
    -- registre del valor de retorn o variables de sortida
    UTL_REG_LOG.AfegirObservacio (nom, tipus, valor);
    -- indicació de sortida del procediment
    UTL_REG_LOG.Fi (resultat, error_oracle, error, comentari)
    RAISE (si cal)

END;
```

Els errors que es puguin produir en el sistema de log genèric no han d'interferir a l'execució normal del sistema transaccional, per això no es fa cap propagació mitjançant excepcions del problemes o errors que es puguin produir. Cada funció retorna un 0 si la traça s'ha emmagatzemat correctament o un codi entre -20000 i -20006 si s'ha produït una incidència, i el programador pot opcionalment veure el que ha passat mitjançant l'ús de la funció *obtenirMissatge'* Si la configuració del sistema no permet emmagatzemar les traces que s'envien, el sistema informarà d'aquests fet a cada crida.

```

/*****\
* UTL_REG_LOG      Mòdul de seguiment de log genèric.      *
/*****\
* Constants de tipus d'observacions                        *
* -----*
* Entrada : paràmetre d'entrada                            *
* Sortida : paràmetre de sortida                          *
* Return  : valor de retorn                               *
* Local   : variable local                                *
* Constants de possibles resultats                        *
* -----*
* Correcte : finalització correcte                        *
* Erroni   : finalització incorrecte                     *
* Avis     : finalització amb avís                       *
\*****/
* MODUL          : AfegirObservacio                       *
* DESCRIPCIO     : Emmagatzema una observació             *
* ENTRADES       :                                       *
*   Nom          : Nom de l'observació a emmagatzemar    *
*   Tipus        : Tipus de variable a emmagatzemar (Entrada/Sortida/Retorn/Local) *
*   Valor        : Valor que pren la variable a emmagatzemar *
* RETORN         :                                       *
*   0            : Execució correcta.                    *
*   -20000       : S'ha produït un error oracle.        *
*   -20002       : La configuració actual no permet emmagatzemar escalars. *
*   -20003       : La configuració actual no permet emmagatzemar aquest tipus d'observació. *
*   -20004       : La configuració actual no permet emmagatzemar LOBs. *
*   -20005       : La grandària de l'observació és superior a la permesa a la configuració actual*
/*****\
* MODUL          : Inici                                   *
* DESCRIPCIO     : Inicialitza un bloc d'execució. Si és el primer inicialitza la transacció *
* ENTRADES       :                                       *
*   Nom          : Nom del bloc                           *
*   categoria    : Categoria del bloc                     *
*   Comentari    : Text lliure a guardar amb la traça    *
*   Usuari       : Usuari que ha activat la transacció  *
* RETORN         :                                       *
*   0            : Execució correcta.                    *
*   -20000       : S'ha produït un error oracle.        *
*   -20006       : S'ha superat la profunditat de l'arbre permesa. *
\*****/

```

```

/*****\
* MODUL      : Fi
* DESCRIPCIO : Finalitza un bloc d'execució. Si és el primer inicialitza la transacció
* ENTRADES   :
* Resultat   : Indicador de finalització del bloc (Correcte, Incorrecta o amb Avís).
* ErrorOracle : Error oracle produït en el bloc
* Error      : Codi d'error intern del programa en execució.
* Comentari  : Text lliure a guardar amb la traça.
* RETORN     :
* 0          : Execució correcta.
* -20000     : S'ha produït un error oracle.
* -20006     : S'ha superat la profunditat de l'arbre permesa.
\*****/
* MODUL      : AfegirTraça
* DESCRIPCIO : Afegeix una traça informativa
* ENTRADES   :
* Comentari  : Text lliure a guardar amb la traça.
* ErrorOracle : Error oracle produït en el bloc
* Error      : Codi d'error intern del programa en execució.
* RETORN     :
* 0          : Execució correcta.
* -20000     : S'ha produït un error oracle.
* -20006     : S'ha superat la profunditat de l'arbre permesa.
* -20007     : La configuració actual no permet emmagatzemar traces informatives.
/*****/
* MODUL      : ObtenirMissatge
* DESCRIPCIO : retorna els missatges d'error acumulats en el procés
* SORTIDA    :
* Missatge   : Missatge d'error
\*****/

```

Atributs emmagatzemats per a cada tipus de registre.

Columna	Bloc	Entrada	Sortida	Informativa
Identificador	√	√	√	√
Nom	√			
Usuari	Només al bloc corresponent a la transacció			
Ordre	√	√	√	√
Tipus	√ (Bloc)	√ (inici Procés)	√ (Fi Procés)	√ (Informativa)
Data inici	√	√	√	√
Data fi	√			
Resultat	√ (C/E/A)			
Categoria	Opcional			
Comentari		Opcional	Opcional	Opcional
Error procediment			Opcional	Opcional
Error oracle			Opcional	Opcional
Observació		Opcional	Opcional	Opcional
Pare	Si no és la transacció	√	√	√

34. Taula d'atributs emmagatzemats per tipus de registre

El sistema de log genèric proporciona a més del visor que s'explicarà a continuació, d'un conjunt d'eines per visualitzar el contingut de les dades emmagatzemades.

El package UTL_VIS_LOG inclou per a cada funció disponible per el visor una versió que volca per pantalla les dades emmagatzemades i una funció que encadena totes les crides.

```

/*****\
* UTL_VIS_LOG Funcions de visualització de les traces emmagatzemades. *
\*****/
* MODUL      : veureUnaTransaccio *
* DESCRIPCIO : Mostra per pantalla tota la informació relativa a una transacció *
* ENTRADES   : *
* transacció : identificador de la transacció *
\*****/
    
```

```

/*****\
* MODUL      : veureUnaTransaccio
* DESCRIPCIO : Mostra per pantalla tota la informació relativa a una transacció
* ENTRADES   :
* nom        : nom de la transacció
/*****\
* MODUL      : VeureTransaccions
* DESCRIPCIO : Mostra per pantalla les transaccions que compleixen les condicions de filtre
*             introduïdes als paràmetres d'entrada.
* ENTRADES   :
* Nom        : Nom de la transacció a localitzar
* Usuari     : Usuari de la transacció a localitzar
* Inici      : Data d'inici de l'execució de la transacció
* Fi         : Data de finalització de l'execució de la transacció
* Resultat   : Resultat obtingut per les transaccions a visualitzar
\*****/
* MODUL      : Arbre execució
* DESCRIPCIO : Mostra per pantalla l'arbre d'execució d'una transacció
* ENTRADES   :
* transacció : identificador de la transacció
\*****/
* MODUL      : detallBloc
* DESCRIPCIO : Mostra per pantalla les traces associades a un bloc
* ENTRADES   :
* bloc       : identificador de la traça que correspon al bloc del que cal obtenir el detall
\*****/
* MODUL      : veureObservacions
* DESCRIPCIO : Mostra per pantalla les observacions associades a una traça
* ENTRADES   :
* traça      : identificador de la traça de la que cal obtenir les observacions
/*****\
* MODUL      : obtenirEstadistica
* DESCRIPCIO : mostra per pantalla informació estadística de les dades emmagatzemades
* ENTRADES   :
* estadística : identificador de l'estadística a obtenir
* Usuari     : Usuari de la transacció que intervindran a l'estadística
* Resultat   : Resultat obtingut per les transaccions a intervenir a l'estadística
* Inici      : Data d'inici de l'execució de la transacció
* Fi         : Data de finalització de l'execució de la transacció
\*****/

```

5.3 Manual d'usuari del visor.

El VisorLog és l'aplicatiu que permet el seguiment del flux d'execució, i que proporciona les eines necessàries per seleccionar la transacció que es vol revisar, per visualitzar el seu arbre d'execució i per mostrar els valors de les variables observades a cada punt de traça.

També proporciona les funcions que permetin identificar :

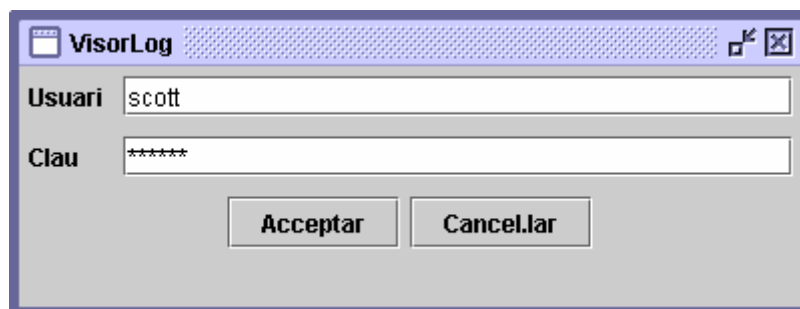
- Procediments més utilitzats, globalment, per categories, per franges horàries i per usuari.
- Procediments agrupats per resultat.
- Franges horàries d'utilització, globalment i per usuari.

El sistema de log pot interferir en el funcionament normal del sistema operacional, especialment en el seu rendiment depenent de la grandària de les variables que es volen observar i de la profunditat de l'arbre d'execució. És per això que mitjançant el VisorLog també es permet configurar els següents aspectes :

- Tipus de dades que s'enregistraran (escalars, objectes).
- Tipus de variables que s'enregistraran (paràmetres, retorn, locals).
- Grandària màxima de la variables a enregistrar.
- Profunditat màxima de l'arbre d'execució.

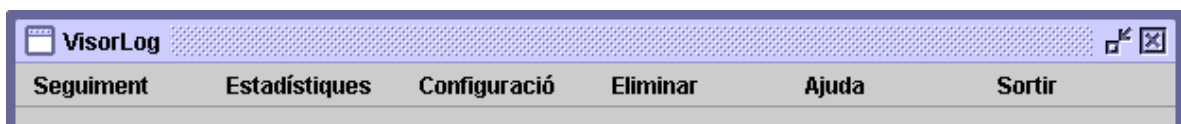
D'una altra banda, també cal preveure el procés d'esborrat de les traces que s'activarà diàriament i que eliminarà les que superin un interval de permanència en el sistema que és també configurable.

L'accés al visor es realitza a través de l'execució de l'aplicació VisorLog. Un cop iniciada l'aplicació és necessari introduir l'usuari i la clau d'accés.



35. Pantalla d'identificació.

Un cop introduïdes aquestes dades apareix el menú de l'aplicació.



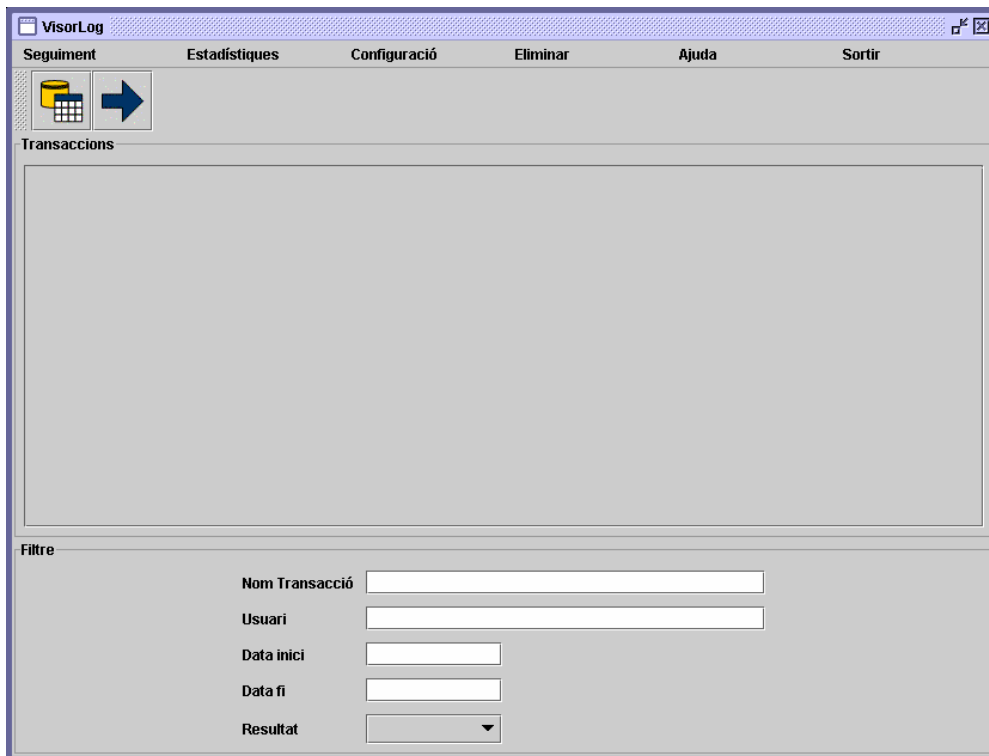
36. Menú del VisorLog

Les opcions disponibles són :

- Opció seguiment Accedeix a les pantalles des de on es poden visualitzar les dades de les transaccions emmagatzemades a la base de dades.
- Opció estadístiques Accedeix a les pantalles per a obtenir diferents estadístiques amb les dades emmagatzemades a la base de dades.
- Opció configuració Accedeix a la pantalla que visualitza i permet actualitzar la configuració actual des sistema de log genèric.
- Opció eliminar Esborra de la base de dades les transaccions caducades.
- Opció ajuda Mostra informació sobre el VisorLog i l'ajuda del sistema de log genèric.
- Opció sortir Tanca l'aplicació.

5.3.1 Opció seguiment.

Quan es prem l'opció de seguiment apareix la pantalla de filtres de transaccions que permet localitzar la transacció a revisar.



37. Pantalla de seguiment (filtre transaccions)

La pantalla es divideix en tres zones :

1. La barra d'eines,



botó d'execució del filtre.



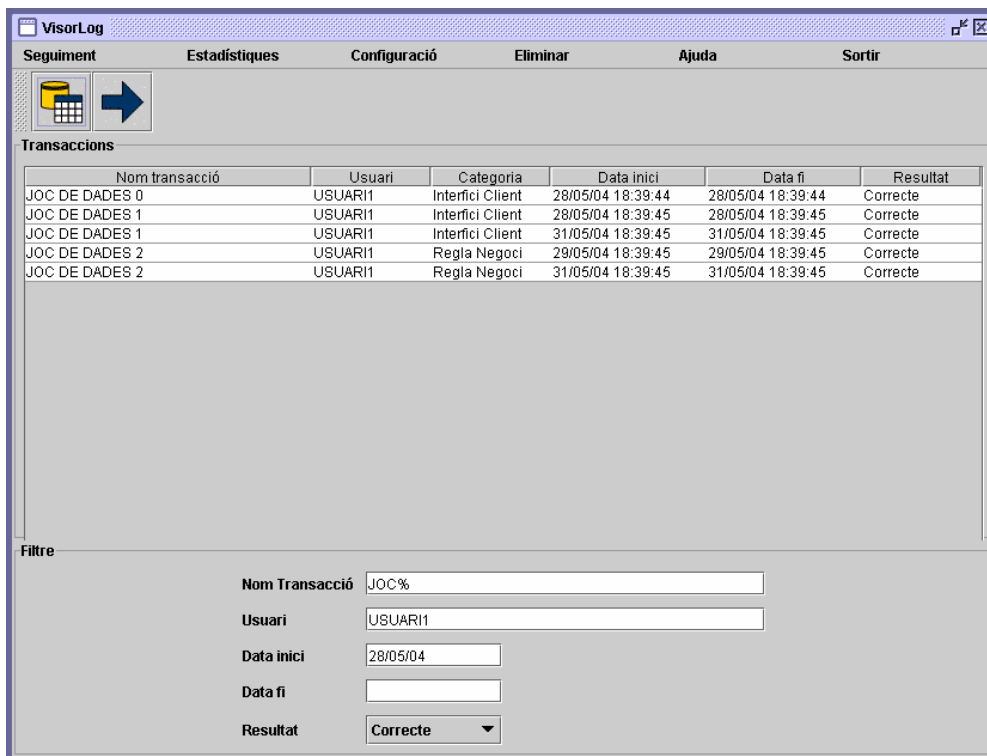
botó endavant

2. La zona amb les transaccions seleccionades.
3. I per últim, la zona de filtres.

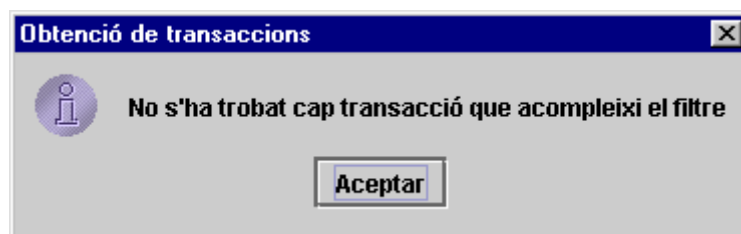
Mitjançant el filtre serà possible acotar la informació en funció dels criteris introduïts, que són :

- Nom : Nom de la transacció. És pot posar el nom complet o utilitzar els comodins per realitzar el filtre.
- Usuari : Nom de l'usuari que ha executat la transacció. És pot posar el nom complet o utilitzar els comodins per realitzar el filtre.
- Temporal (entre les dates d'inici i fi de la transacció) : inclou només aquelles transaccions que s'han produït entre les dates introduïdes.
- Resultat (correcte/incorrecte/avís/Desconegut) : inclou només aquelles transaccions que han finalitzat amb el resultat escollit.

Un cop introduïdes les dades s'executa el filtre mitjançant el botó execució i la zona de transaccions es refresca amb les transaccions que compleixen el filtre. Si no hi ha transaccions que compleixen el filtre proposat es mostrarà un avís i es permetrà introduir un nou filtre.

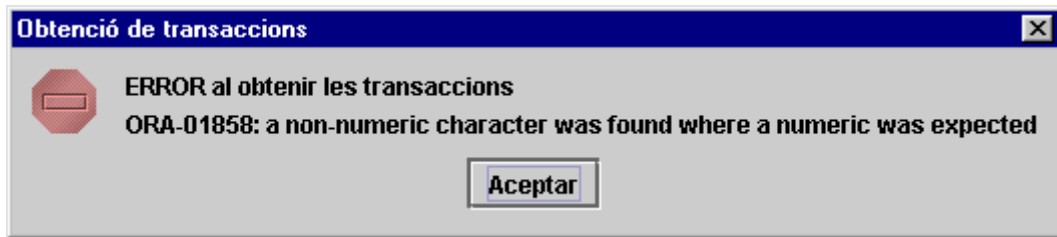


38. Pantalla de seguiment (exemple filtre transaccions)



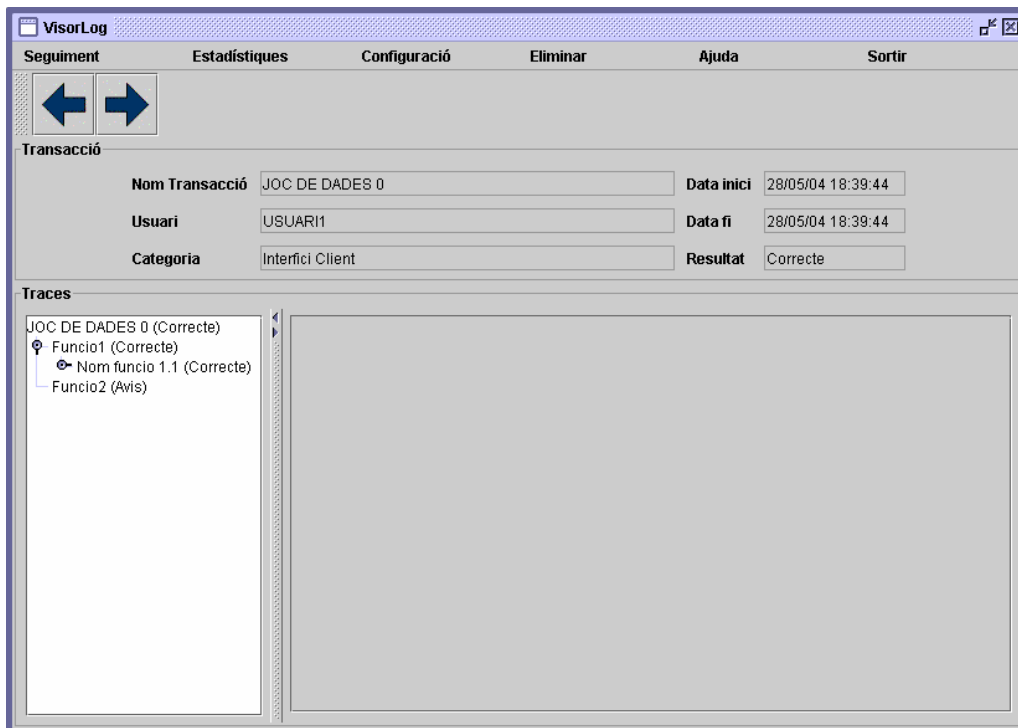
39. Missatge d'avís en l'obtenció de transaccions

Si es produeix un error a l'obtenció de les transaccions es mostrarà un missatge amb la descripció de la incidència :



40. Missatge d'error en l'obtenció de transaccions

Un cop s'escull una transacció i és selecciona amb el botó endavant s'accedeix a la pantalla que mostra l'arbre d'execució.



41. Pantalla de seguiment (arbre execució)

La pantalla es divideix en quatre zones :

1. La barra d'eines,



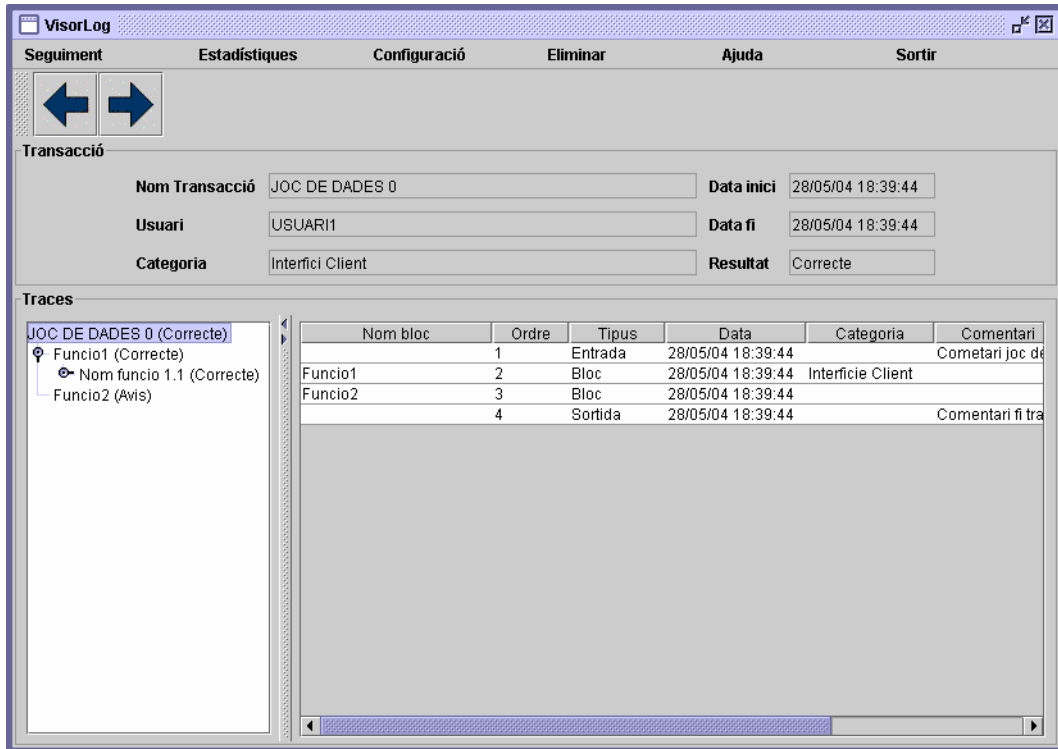
botó enrera (retorna a la pantalla de filtres de transacció).



botó endavant

2. La zona amb les dades de la transacció activa.
3. La zona amb l'arbre d'execució.
4. I per últim, la zona de detall del bloc.

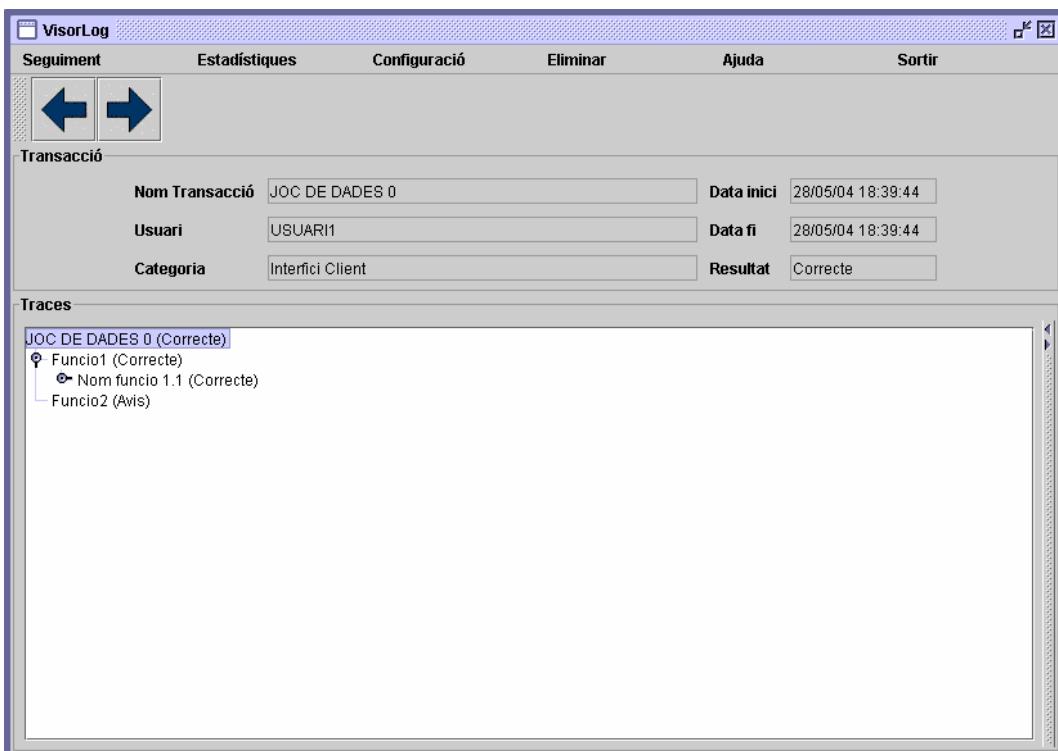
La de detall del bloc només apareix un cop s'ha seleccionat un bloc, és a dir, un node de l'arbre.



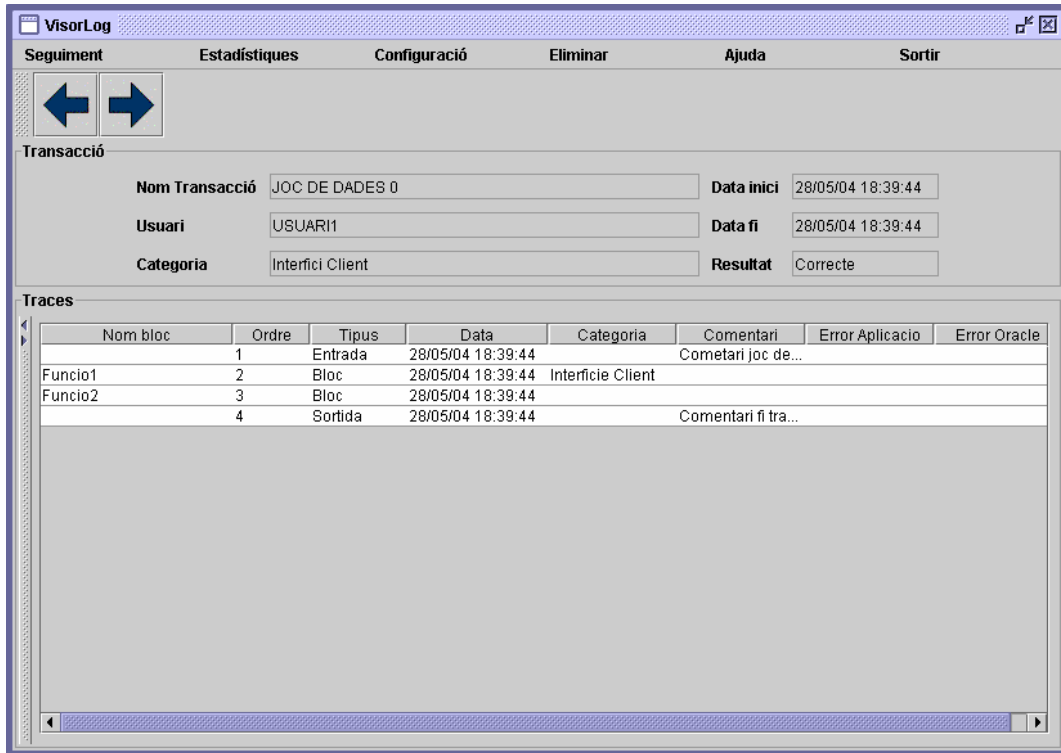
42. Pantalla de seguiment (detall d'un bloc)

A l'arbre d'execució es poden visualitzar totes les crides a funcions de la transacció i l'estat de finalització de cada una d'elles i en el detall de cada bloc es pot veure la relació de totes les traces del bloc.

Mitjançant la barra de desplaçament podem visualitzar únicament la zona de l'arbre d'execució o la zona de detall del bloc :

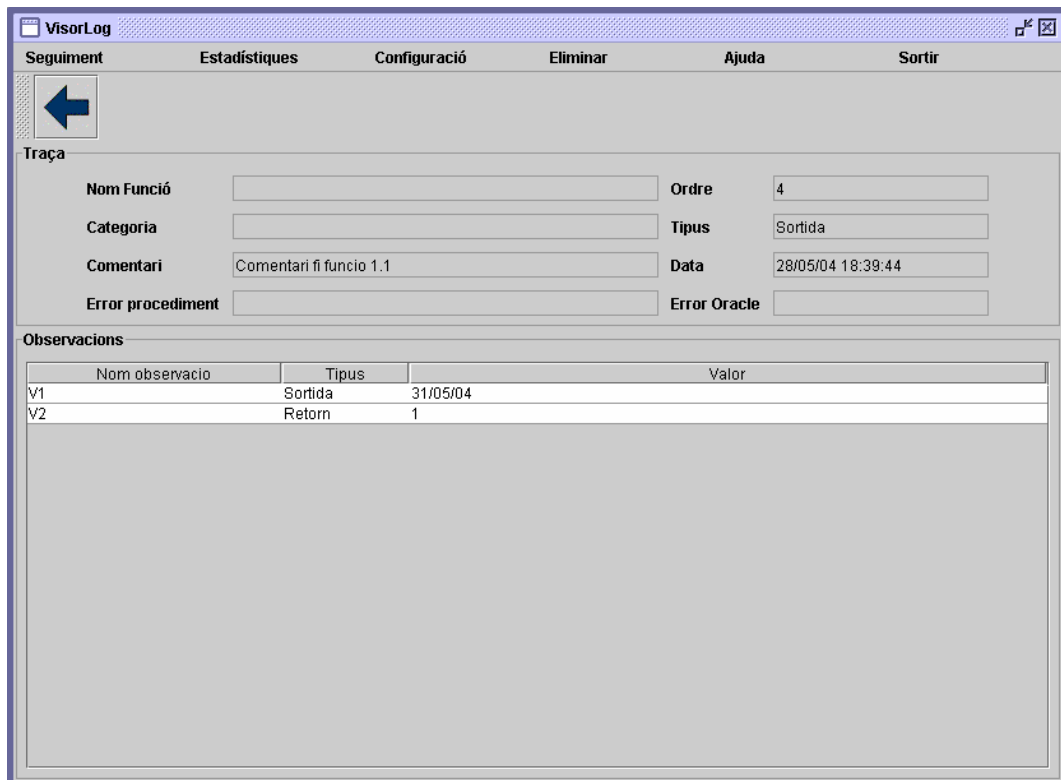


43. Pantalla de seguiment (només arbre)



44. Pantalla de seguiment (només detall bloc)

Si seleccionem una traça i premem el botó endavant s'accedeix a la pantalla que mostra les observacions emmagatzemades a la traça, és a dir, els paràmetres d'entrada, sortida, variables locals o de retorn que s'han emmagatzemat a la traça. Si no hi ha observacions emmagatzemades a la traça seleccionada, l'aplicació ens mostrarà un avís i permetrà seleccionar un altre traça.



45. Pantalla de seguiment (observacions)

La pantalla es divideix en tres zones :

1. La barra d'eines,



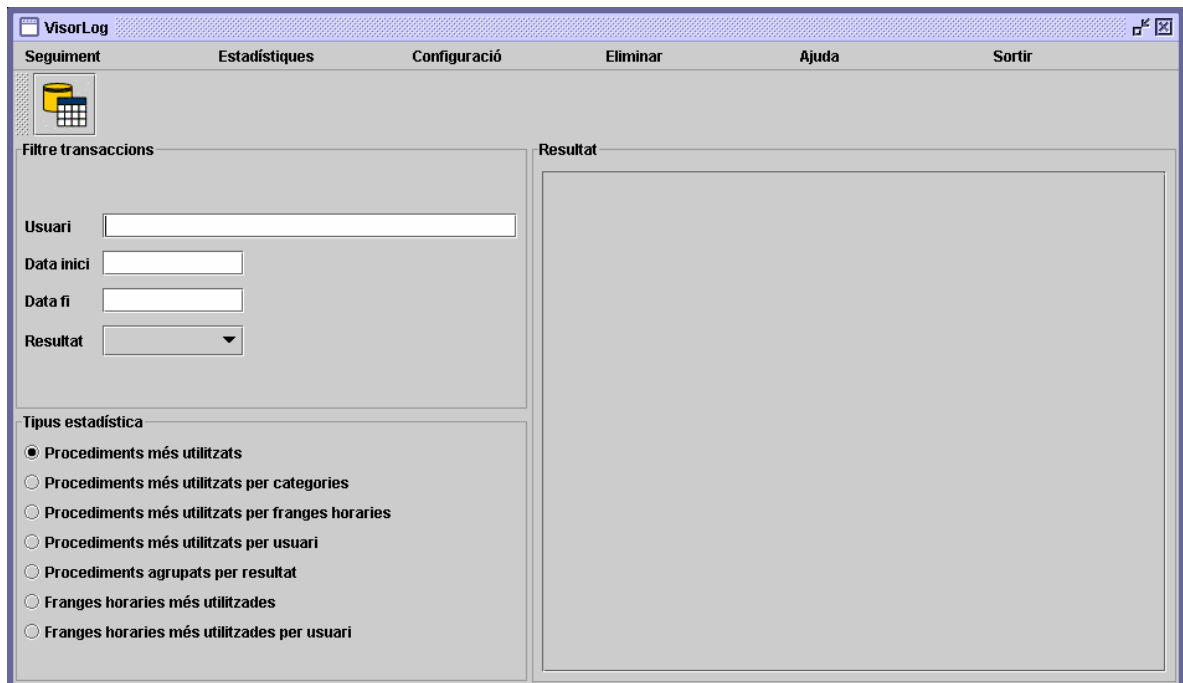
botó enrera (retorna a la pantalla amb l'arbre d'execució).

2. Una zona amb la Informació relativa a la traça activa.
3. I per últim la zona amb les observacions de la traça.

5.3.2 Opció estadístiques.

La funcionalitat d'estadístiques permet obtenir informació estadística de les dades emmagatzemades per el sistema de log genèric.

Al entrar a l'opció apareix la següent pantalla :



46. Pantalla d'estadístiques

La pantalla es divideix en quatre zones :

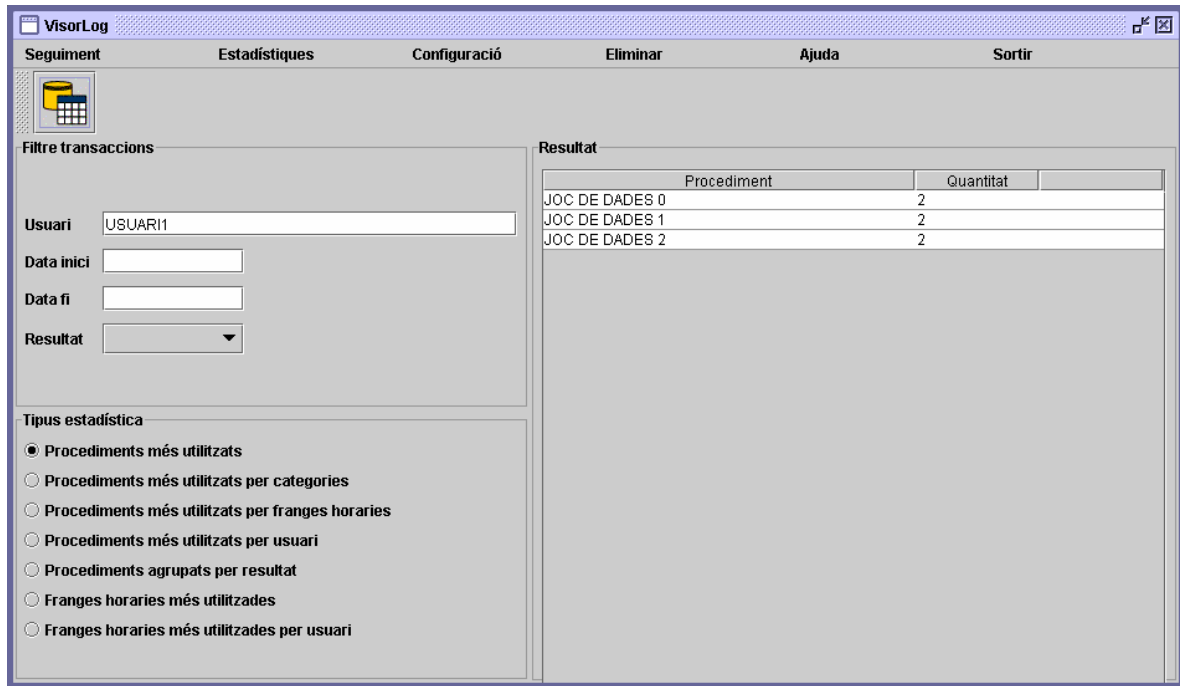
1. La barra d'eines,



botó d'execució del filtre.

2. La zona de filtres.
3. La zona de selecció d'estadística.
4. I la zona de visualització de resultats.

Utilitzant els filtres es poden acotar les transaccions a intervenir a l'estadística.



47. Pantalla d'estadístiques (exemple filtre)

Els filtres disponibles són :

- Usuari : Nom de l'usuari que ha executat la transacció. És pot posar el nom complet o utilitzar els comodins per realitzar el filtre.
- Temporal (entre les dates d'inici i fi de la transacció) : inclou només aquelles transaccions que s'han produït entre les dates introduïdes.
- Resultat (correcte/incorrecte/avis) : inclou només aquelles transaccions que han finalitzat amb el resultat escollit.

Les estadístiques disponibles són :

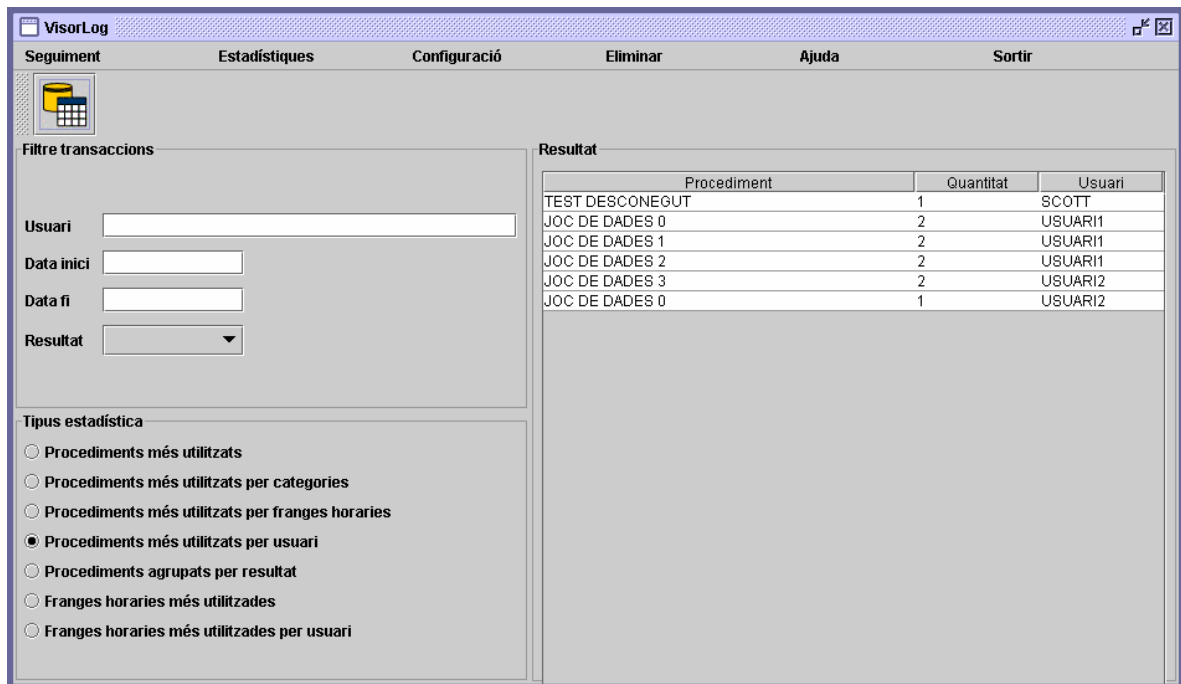
1. Procediments més utilitzats
2. Procediments més utilitzats per categories
3. Procediments més utilitzats per franges horàries
4. Procediments més utilitzats per usuari
5. Procediments agrupats per resultat
6. Franges horàries més utilitzades
7. Franges horàries més utilitzades per usuari

Les franges horàries definides en el sistema són :

Franja	Des de	Fins a
MATINADA	1 h	8 h
MATI	9 h	13 h
DINAR	14 h	15 h
TARDA	16 h	19 h
VESPRE	20 h	22 h
NIT	23 h	24 h

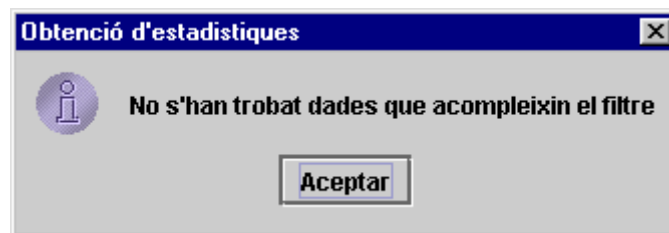
48. Taula amb les franges horàries definides en el sistema

Al prémer el botó executar a la zona de resultats es visualitza les dades obtingudes de l'estadística seleccionada.



49. Pantalla d'estadístiques (exemple estadística)

Si no hi ha dades es mostrarà un avís :



50. Missatge d'avís obtenció estadística

5.3.3 Opció configuració.

Des de aquesta pantalla es pot consultar i configurar el comportament del sistema de log, és una eina per tal de que l'administrador del sistema pugui limitar per diferents conceptes el que el sistema genèric anirà emmagatzemant i quan s'esborrarà automàticament de la base de dades.

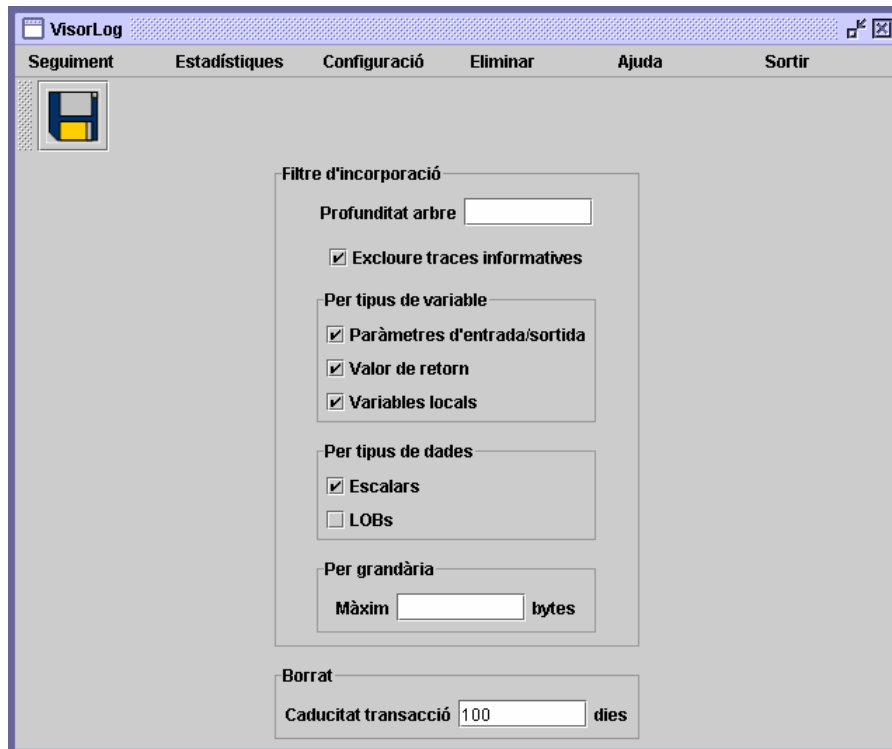
La pantalla es divideix en dues zones :

1. La barra d'eines,



botó per actualitzar la configuració.


2. La zona amb les dades de configuració.



51. Pantalla de configuració

Les dades que es poden actualitzar són :

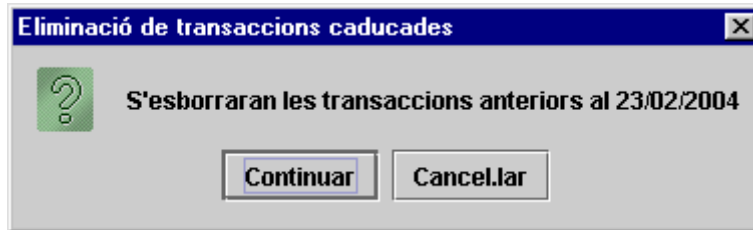
- Profunditat arbre : Profunditat de l'arbre d'execució que el sistema de log permet emmagatzemar.
- Informatives : Indicador que defineix la restricció d'emmagatzemar les traces de tipus informatives.
- Paràmetres : Indicador que defineix la restricció d'emmagatzemar els paràmetres d'E/S.
- Valors retorn : Indicador que defineix la restricció d'emmagatzemar els valors de retorn.
- Variables locals : Indicador que defineix la restricció d'emmagatzemar les variables locals.
- Escalars : Indicador que defineix la restricció d'emmagatzemar les observacions de tipus de dades escalars.
- LOBs : Indicador que defineix la restricció d'emmagatzemar les observacions de tipus de dades LOBs.
- Grandària : Màxim valor en bytes que pot ocupar un valor a emmagatzemar.
- Caducitat : Nombre de dies que es manté una transacció en el sistema.

Al entrar a l'opció configuració la pantalla mostra la configuració actual del sistema. Es poden realitzar canvis a la configuració i guardar-los a base de dades amb el botó d'actualitzar configuració . Si es fan canvis i no es confirmen premen el botó, la configuració quedarà tal i com estava abans dels canvis.

5.3.4 Opció eliminar.

Mitjançant aquesta opció s'eliminen de la base de dades les transaccions anteriors a la data de caducitat obtinguda amb el paràmetre configurat a la base de dades. El valor del paràmetre es pot consultar i modificar des de l'opció de configuració.

L'aplicació mostrarà un avís per demanar confirmació abans de realitzar l'esborrat. Si es confirma, s'eliminaran de la base de dades totes les traces de les transaccions anteriors a la data indicada.



52. Missatge d'avís del procés d'esborrat

5.3.5 Opció ajuda.

Mostra informació sobre el VisorLog i l'ajuda del sistema.



53. Pantalla d'ajuda

5.3.6 Opció sortir.

Tanca el visor.

6. Conclusions.

Pel que fa a les conclusions, es poden classificar en dos grans grups, les referides a la evolució i el resultat del projecte i les associades a les tècniques emprades per desenvolupar la solució.

Des del punt de vista del projecte es pot dir que :

- El projecte s'ha desenvolupat dins dels terminis establerts, amb petites derives superades amb un cert esforç addicional.
- El resultat és l'esperat i s'han lliurat tots els entregables identificats al pla de projecte, incloent-hi els executables del sistema que han superat les proves unitàries i funcionals.

Des d'un punt de vista tecnològic :

- S'ha arribat a conèixer la estructura del PL/SQL d'Oracle, el funcionament dels blocs d'execució, la utilitat del procediments, funcions, paquets proporcionats per Oracle i paquets desenvolupats per mi mateixa i, especialment, el mecanisme de propagació d'excepcions.
- Considero que l'aparició dels tipus genèrics de dades (ANYDATA) pot proporcionar grans avantatges als dissenyadors de sistemes, tot i que cal solucionar el problema de la persistència de LOBs dins de columnes de tipus genèric.
- Les taules niades (NESTED TABLES) poden simplificar, utilitzades amb prudència, el model lògic de la BB.DD. i facilitar la seva comprensió, tant per part del client com del desenvolupador.
- Cal conèixer les restriccions que Oracle imposa a la sobrecàrrega de funcions, especialment pel que fa al tipus RAW que es considera de la família dels CHAR i als tipus TIMESTAMP i INTERVAL que es poden considerar com a CHAR o com a DATE. Crec que Oracle hauria de solucionar aquestes indefinicions.
- La tècnica de la transacció autònoma ha simplificat notablement el disseny dels procediments i la considero una eina d'especial utilitat en el desenvolupament de sistemes de seguiment i traçabilitat i, en general, sistemes que necessitin independitzar dues transaccions.

Per últim, voldria expressar la meua inquietud per continuar la investigació en el tema de l'automatització del procés d'identificació dels paràmetres d'un procediment mitjançant la utilització del package DBMS_DESCRIBE i la vista ALL_ARGUMENTS. Una troballa en aquest sentit permetria afegir funcionalitat al sistema de seguiment i lliurar al programador de la necessitat d'identificar-los manualment. Tot i això, el que realment espero és que Oracle doni una solució més eficient, potser dins del package DBMS_TRACE, ja que aquest és un problema força genèric.

Glossari.

ALL_ARGUMENTS

Vista que proporciona Oracle dels arguments d'un objecte accessibles per un usuari.

ANYDATA TYPE

És un nou tipus genèric de dada que pot contenir un valor de qualsevol altre tipus, és a dir, és possible desar qualsevol tipus de dada PL/SQL donat que emmagatzema, de forma auto-continguda, el valor i una descripció del tipus.

DBMS_DESCRIBE

Package que conté una funció (DESCRIBE_PROCEDURE) que descriu els arguments d'un procediment PL/SQL. Retorna un conjunt de taules PL/SQL amb informació relativa als paràmetres del procediment.

CONNECT BY

Mitjançant la clàusula CONNECT BY és possible establir relacions jeràrquiques entre pares i fills.

NESTED TABLES

És un tipus de col·lecció, que no té límit en el número d'ocurrències per fila. És, de fet, una taula dins d'una altra i es representa amb una columna dins de la taula.

Sobrecàrrega

S'entén per sobrecàrrega la capacitat de definir dos procediments amb el mateix nom però amb paràmetres diferents. Conceptualment implementen la mateixa operació però ho fan sobre dades de tipus diferents.

Swing

Biblioteca de controls per al desenvolupament d'interfícies gràfiques d'usuari.

TRANSACCIÓ AUTÒNOMA

Una transacció autònoma és una transacció independent iniciada per una altre transacció. Permet confirmar o desestimar, independentment de la confirmació o desestimació de la resta d'accions del mateix procés un conjunt de accions. Quan es crida una transacció autònoma la transacció original (la que la crida) queda temporalment suspesa. La transacció autònoma ha de finalitzar confirmant o desestimant explícitament la transacció, amb un COMMIT o un ROLLBACK) abans de retornar el control a la transacció principal.

VisorLog

És l'aplicatiu que permet el seguiment del flux d'execució, i que proporciona les eines necessàries per seleccionar la transacció que es vol revisar, per visualitzar el seu arbre d'execució i per mostrar els valors de les variables observades a cada punt.

Bibliografia.

- **Oracle9i Database Concepts, Release 1 (9.0.1).** *Part No. A88856-02*
Copyright © 2001, Oracle Corporation. All rights reserved.
- **Oracle9i SQL Reference, Release 1 (9.0.1).** *Part No. A90125-01*
Copyright © 1996, 2001, Oracle Corporation.
- **PL/SQL User's Guide and Reference, Release 9.0.1.** *Part No. A89856-01*
Copyright © 1996, 2001, Oracle Corporation.
- **Oracle9i Supplied PL/SQL Packages and Types Reference, Release 1 (9.0.1).** *Part No. A89852-02*
Copyright © 1996, 2001, Oracle Corporation.
- **Oracle9i JDBC Developer's Guide and Reference, Release 1 (9.0.1).** *Part No. A90211-01*
Copyright © 1996, 2001, Oracle Corporation.
- **Oracle9i Database Reference, Release 1 (9.0.1)** . *Part No. A90190-02*
Copyright © 1996, 2001, Oracle Corporation. All rights reserved.
- **Oracle PL/SQL Programming, Third Edition, and Oracle PL/SQL Best Practices**
(O'Reilly & Associates, <http://oracle.oreilly.com/>).
- **Java™ 2 Platform, Standard Edition, v 1.4.2. API Specification**
<http://java.sun.com/j2se/1.4.2/docs/api/index.html>
- **The JDBC Tutorial: Chapter 3 - Advanced Tutorial**
By Maydene Fisher
<http://java.sun.com/developer/Books/JDBCTutorial/>
- **Tutorials & Code Camps. The Java Tutorial**
<http://java.sun.com/docs/books/tutorial/index.html>

Articles :

- **Building a Code-Analysis Utility and Doing It Right the First Time**
By Steven Feuerstein
http://otn.oracle.com/oramag/webcolumns/2003/techarticles/codech_index.html

Annexos

Annex 1. Joc de proves disponibles.

Al directori `\proves` es troben els fitxers que contenen dades per validar el sistema.

Test_delete.

Test per validar el procediment `utl_del_log`. Per tal de validar el procediment de neteja de transaccions antigues el test en primer lloc actualitza la configuració per posar l'atribut de configuració a 1 (un dia).

El procediment crea tres transaccions (*TEST DELETE1*, *TEST DELETE2* i *TEST DELETE3*) i actualitza la data de la transacció *TEST DELETE2* per simular que es va produir fa dos dies. A continuació executa el procediment `UTL_VIS_LOG.VeureTransaccions` que visualitza per pantalla les transaccions i executa el procediment d'esborrat i torna a visualitzar les transaccions actuals per comprovar que la transacció *TEST DELETE2* ja no existeix a la base de dades.

Cal executar el script `test_delete.sql` des de el SQL*Plus. La prova genera el fitxer `test_delete.log` al directori `c:\.`:

```
SQL> @test_delete
```

Test_config.

Test per validar que el sistema de log emmagatzema la informació segons indica la configuració actual del sistema.

Cal executar el script `test_config.sql` des de el SQL*Plus. La prova genera el fitxer `test_config.log` al directori `c:\.`:

```
SQL> @test_config
```

Test_variables.

Test per a validar l'emmagatzemament de tots els tipus de variables PL/SQL. Introdueix una transacció on té definides tots els tipus de dades PL/SQL i després la visualitza.

Cal executar el script `test_variables.sql` des de el SQL*Plus. La prova genera el fitxer `test_variables.log` al directori `c:\.`:

```
SQL> @test_variables
```

Annex 2. Joc de dades disponibles.

Joc_de_dades.

Introdueix algunes transaccions per poder visualitzar-les des del visor o des de les funcions que mostren per pantalla del `UTL_VIS_LOG`.

Cal executar el script `joc_de_dades.sql` des de el SQL*Plus.

```
SQL> @joc_de_dades
```