ESPECIFICACIÓ DEL FORMAT "DBF ESTESA" (INCLOU LA IMPLEMENTACIÓ "NOMS ESTESOS")

Autors del document: Xavier Pons i Abel Pau
Proposta inicial: 23-12-2011
Darrera modificació i versió del document: 16-02-2024. **1.9**

1. Antecedents i motivació.

El **format DBF**, inicialment desenvolupat en el context del programari dBASE, seguit en altres *softwares* com FoxBase i FoxPro, i descrit en diversos llocs a la literatura especialitzada i a Internet, és, exceptuant les formes tabulars en text pla, potser el més popular dels formats de taules alfanumèriques de dades. El **MiraMon** pot accedir a dades tabulars en format DBF III, III+ i IV (i admet algunes addicions posteriors, com camps F, etc), així com també a dades tabulars (físiques o resultat d'expressions de consulta o càlcul) contingudes en altres formats de fitxers (com ara XLS, MDB, etc) i en bases de dades (com ara Oracle, SQL-Server, etc); en aquests darrers casos caldrà que existeixin i estiguin correctament instal·lats els corresponents *drivers* ODBC i, amb l'excepció de l'MDB i l'ACCDB (que admeten accés directe des del MiraMon), que es creï un fitxer DSN per a accedir a les dades.

Malgrat el gran potencial que presenta l'accés a les diferents fonts tabulars esbossades més amunt, en el context MiraMon sovint DBF és el format de taules d'elecció, tant per la seva senzillesa i velocitat, com pel fet que no cal dependre de tercers, o simplement perquè, en difondre informació, es vol tenir la certesa que es podrà obrir la informació en un ordinador del qual desconeixem quins *drivers* té instal·lats (motiu pel qual és l'opció per defecte en la creació de fitxers MMZX i MMZ, destinats a una àmplia difusió per Internet). Cal recordar també que les taules DBF poden ser enllaçades entre si i amb altres taules (MDB, SQL-Server,

etc) tot establint enllaços (*join*) en què és possible indicar la cardinalitat (1 a 1, 1 a molts, etc) i el nivell d'obligatorietat de la resolució de l'enllaç (tipus diccionari o no) en un conjunt de relacions que no està limitat ni en nombre de relacions per camp, ni en nombre de nivells de relacions. Aquests enllaços s'emmagatzemen en els corresponents **fitxers REL**.

Tanmateix, el format DBF, per la seva antiguitat, presenta una sèrie de limitacions. L'evolució del format DBF després de la versió IV va ser força complicada (venda del producte entre diferents empreses, estranyes decisions com la inclusió de nous tipus numèrics que sovint no aportaven res de significatiu, etc [vegeu http://en.wikipedia.org/wiki/DBase]), per la qual cosa en general la comunitat d'usuaris i desenvolupadors continua fidel als formats III, III+ i IV tant per a lectura com per a exportació. Actualment hi ha encara un producte dBASE a la venda (http://www.dbase.com).

Algunes de les limitacions del DBF IV han estat superades gràcies a indicacions que també s'emmagatzemen en els fitxers REL i s'estableixen des del **Gestor Universal de Metadades Geoespacials del MiraMon**, **GeM+**. Algunes de les més destacades són:

- La longitud del nom del fitxer, que en l'especificació original era 8+3, i que en el MiraMon s'amplia a qualsevol longitud suportada pel sistema operatiu. A més, en el MiraMon els noms dels fitxers DBF i els directoris on són poden contenir espais, accents, etc.
- La longitud del nom de cada camp DBF està limitada a 10 caràcters que, a més, no poden contenir lletres accentuades, espais, caràcters especials, etc. En canvi, des del GeM+ es pot indicar un descriptor de text lliure, sense limitacions d'accents, caràcters especials, etc, per a cada camp; a més, el descriptor pot ser multiidiomàtic si es desitja i admet una longitud, en caràcters, de:
  max(_MAX_PATH+100,256)
- En les taules anteriors a dBASE IV el joc de caràcters utilitzat en els camps de tipus 'C' no estava especificat. El MiraMon permet una solució flexible i configurable per a aquests casos i, com en les taules dBASE IV, admet indicar el joc de caràcters (byte a l'*offset* 29), amb la qual cosa la interpretació d'accents i caràcters especials deixa de ser ambigua.
- En els camps amb contingut numèric no es pot indicar les seves unitats quan en tenen. En canvi, des del GeM+ aquestes poden ser especificades, així com si es desitja que siguin mostrades en les consultes.
- En les taules DBF no es pot indicar la qualitat del contingut de cada camp. En canvi, des del GeM+ aquesta pot ser especificada.
- En les taules DBF no es pot indicar el tractament (categòric, ordinal o quantitatiu continu) del contingut de cada camp. En canvi, des del GeM+ aquesta propietat també pot ser especificada.

A pesar d'aquestes importants extensions introduïdes a través dels fitxers REL (de les quals també es poden beneficiar altres formats tabulars llegibles des del

MiraMon), existeixen altres limitacions del format DBF que no poden ser resoltes excepte amb la introducció de petites modificacions en el propi format. Entre les més importants podem destacar:

- La limitació del nombre de camps a 128 en dBASE III+ (llibre de dBASE de Jordi Abadal, p. 137, i criteri d'exportació de Excel 2000) o 255 (dBASE IV). Aquesta limitació no només pot ser trobada en taules de tota mena, sinó que encara és més freqüent en la taula única creada en resoldre tots els enllaços de l'arbre de relacions especificat des del GeM+ quan les relacions són nombroses, i especialment quan la vinculació es fa amb taules amb molts camps. Finalment, noteu que el MiraMon no estableix aquesta distinció entre versions dBASE III+ o dBASE IV i assumeix, per a qualsevol DBF clàssica, independentment de la versió:
  - 255
- La limitació del nombre de caràcters, en camps de tipus 'C':
  - 254
- La limitació del format i longitud del nom dels camps de la DBF a 10 caràcters en majúscules (sense suport a lletres accentuades, ç, etc).
  - 11 (incloent el \0 de final de cadena)

Hi ha d'altres millores possibles a esmentar que podrien ser objecte de discussió sobre l'oportunitat del seu disseny i implementació en un futur:

- Els nombres s'escriuen en realitat com a text que cal interpretar. Seria desitjable definir tipus numèrics binaris seguint els estàndards habituals per a enters i reals.
- Suport a camps de text de longitud variable amb un sistema d'indexació del fitxer que diu on comença cada registre (tot i que això faria una mica més lent l'accés).
- Suport a camps binaris de longitud il·limitada.
- Suport a compressió individual per registre amb un sistema d'indexació del fitxer que diu on comença cada registre.
- Suport a una marca d'esborrat d'un camp (columna).
- Optar per una descripció en text de la capçalera.
- La primera "minicapçalera" de 32 bytes podria passar a ser extensible per encabir ampliacions futures. Això implica codificar la mida de la pròpia minicapçalera.
- La descripció de cada camp podria ser extensible. Això implica codificar a la minicapçalera la mida de la descripció d'un camp.
- Passar el nombre màxim de registres a un **unsigned __int64** (enter de 64 bits). El valor es pot escriure combinant els 4 bytes de la DBF clàssica més els 4 bytes 16-19 de la capçalera.
- La data interna del fitxer es podria eliminar. Sempre ha estat un maldecap i és redundant amb la del sistema de fitxers. A més, no conté l'hora i, doncs, no és útil ni per a restaurar la data-hora exacta d'un fitxer que s'ha enviat per correu electrònic i que ha perdut la data.
- Es podria introduir un camp de tipus data-hora.

L’equip del MiraMon proposa solucionar les més importants d’aquestes limitacions i adoptar alguna de les millores, tot establint una variació del format DBF que anomenarem “**DBF estesa**”.

2. Característiques i utilització del format “DBF estesa”.

2.1 Si una taula no necessita superar les limitacions de la DBF clàssica, és preferible escriure-la en aquest format a fi que sigui llegible per altres *softwares* que no suportin la DBF estesa. Noteu que el format no manté la compatibilitat descendent (un *software* que llegeix DBF clàssiques no llegirà una DBF estesa, ni parcialment, llevat que hagi implementat la proposta d’aquest document).

2.2 L'**extensió** del fitxer és **.dbf** com en la DBF clàssica.

2.3 El **primer byte** és **0x90**. Per comprovar si el software pot llegir això, només comprova el '9', el que permet que canviï el segon número per a futures modificacions compatibles descendents. No s’utilitzen valors menors per no entrar en conflicte amb altres numeracions.

2.4 El **nombre de camps** possible passa a ser de 13.4 milions (el valor màxim exacte es justifica en el següent apartat).

2.5 La **longitud d'un registre** és un **unsigned __int32**, comptant el byte d’esborrat del registre.

2.6 La **mida d'un camp** C pot arribar a **unsigned __int32 - 1** (es renuncia a un byte per a encabir el byte d’esborrat).

2.7 El fitxer continuarà contenint la marca de final de definició de camps (*Header Record Terminator* - 0x0D) com en les DBF clàssiques ja que en el codi caldria fer una bifurcació arreu on es calcula l'*offset* al primer registre i es considera massa crític el problema que es generaria entre DBF clàssiques i esteses i, doncs, es prefereix sacrificar aquest byte d’emmagatzematge.

2.8 El fitxer NO conté la marca de final de fitxer (0x1A) que sovint es pot trobar en les DBF clàssiques. El fitxer contindrà els noms estesos de camps entre la marca de final de definició de camps (*Header Record Terminator* - 0x0D) i la primera fitxa. L’accés a aquest camps es farà a partir d’un *offset* i una mida (els noms estesos no contindran el ‘\0’ final) emmagatzemats en la capçalera dels camps (l’*offset* en el reservat2+7, de longitud 4 bytes, i la mida, en el reservat2+11 de longitud un byte. La màxima longitud d’un nom estès d’un camp seria, doncs, de 255 caràcters, però com es comenta més endavant, acaba essent de 128 caràcters, suficient en comparació amb el que suporten altres gestors de grans bases de dades (l’Oracle 9.i en suporta 30, i l’SQL Server 2000, 128).

2.9 S’avisarà amb un missatge en el cas que una DBF passi de no contenir noms estesos a contenir-los:

"El nom del camp que vols crear (%s) té característiques no compatibles amb la definició clàssica dels fitxers DBF (com ara accents, espais, longitud més gran que 10, etc). Recordeu que també podeu generar noms de format lliure des de la casella "Descriptor" o des del Gestor de Metadades (GeM+).

Desitges utilitzar el nom igualment? [Sí] [No]"

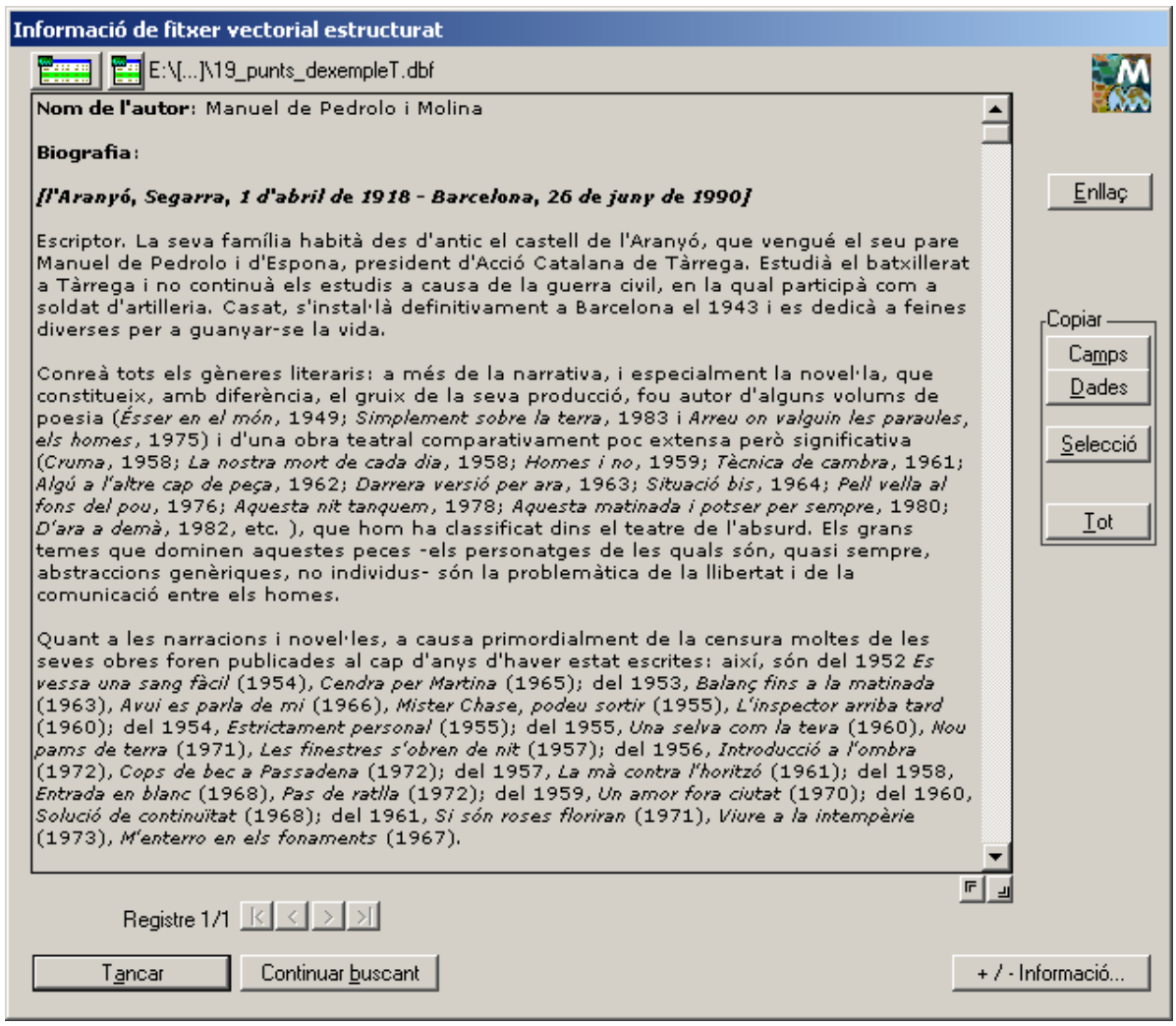
Els camps de C amb longituds esteses permeten posar en els camps textos especialment llargs i complexos, com ara codificació HTML, expressions xpath() per a l'accés a recursos de dades obertes (*open data*), etc. En el següent exemple extret d'una base hipotètica on se situa el lloc de naixement de diversos escriptors, la consulta permet accedir a la seva biografia (molt més extensa, uns 100 000 caràcters, que el que apareix a la captura, com podeu deduir per la grandària del botó de la barra de desplaçament vertical), en aquest cas extreta de l'Enciclopèdia Catalana a Internet. Naturalment també es podria afegir l'enllaç a una foto, enllaços a altres recursos a Internet o a una intranet, etc.

En l'exemple mostrat, el camp comença així:

```
<HTML><BR><BR><b><i>[l'Aranyó, Segarra, 1 d'abril de 1918 - Barcelona, 26 de juny de 1990]</i></b><BR><BR>Escriptor....
```

i acaba així:

```
...difusió de la seva obra, així com a la del seu ideari.</HTML>
```



3. Especificació del format “DBF estesa”.

El **primer byte** és **0x90**. Aquesta és la **marca de DBF estesa**. Millores posteriors podrien implicar numeracions hexadecimals successives.

Els **2 bytes** anomenats **reservat_1** (bytes 12-13, numerats des de 0) seran llegits conjuntament amb els 10 i 11 com un sol paquet de 4 bytes (*unsigned __int32*) que definirà el **nombre de bytes per registre**. L’extensió a un enter de 32 bits és necessària per a poder encabir, per exemple, diversos camps C amb una amplada important.

Els **4 bytes** ubicats a bytes 16-19 seran llegits conjuntament amb els ubicats a bytes 04-07 com un sol paquet de 8 bytes (*unsigned __int64*) que definirà el **nombre de registres**. L’extensió a un enter de 64 bits és necessària per a poder encabir, per exemple, els atributs de punts de fitxers lidar d’alta densitat sobre països molt grans.

Els **2 bytes** 30-31 seran llegits conjuntament amb els 8 i 9 com un sol paquet de 4 bytes que definirà **on s’inicia l’emmagatzematge dels registres**. A diferència de la DBF clàssica, on el **nombre de camps de la taula** determina on acaba la capçalera DBF, ara això ja no és així ja que cal preveure que encara hi haurà, després de la marca de final de la capçalera clàssica (*Header Record Terminator*, 0x0D), els noms estesos. El nombre de camps que suporta ara, doncs, la DBF estesa queda condicionat pels 32 bytes destinats a descriure cada camp i a aquest *offset* on comencen els registres, dintre el qual cal incloure, a més de la descripció de cada camp, la capçalera de 32 bytes, el byte final 0x0D que marca el final de la descripció dels camps i els noms estesos. Així doncs, de moment queda: (2147483648-32-1)/(32+128)= 13 421 772, o sigui **uns 13.4 milions de camps**.

En **camps especials** a les DBF esteses (els **C** a la versió actual, 0x90), el nombre de **bytes per camp** no es defineix al byte 16, sinó als bytes 21-24 (*unsigned __int32*) de cada paquet de 32 bytes que defineix cadascun dels camps. El 16è byte queda, en aquests casos, amb valor 0.

Els **noms estesos** en les noves taules DBF permeten una longitud de 128 caràcters i pràcticament qualsevol caràcter (cal notar que en altres grans gestors de bases de dades l’amplada del nom dels camps és igual o inferior a la proposa en la DBF estesa i, doncs, s’aconsegueix una màxima compatibilitat). Els únics caràcters que no es permeten són l’accent obert sol (sense accentuar cap lletra), els claudàtors, la dièresi i els caràcters no imprimibles (com ara el retorn de carro o el DEL), fent així la DBF compatible amb els noms de camps permesos en les bases de dades Oracle, MySQL o SQL Server. L'accés als noms estesos de camp es fa a través dels bytes 25, 26, 27, 28 i 29 de la capçalera del camp en qüestió. Més concretament, els bytes 7, 8, 9 i 10 defineixen l’*offset* on cal anar a buscar el

nom estès del camp. El byte 11 fa referència a la mida del nom, que com a màxim serà, com hem dit, de 128 caràcters. Permetre més caràcters (fins a 255) no sembla necessari quan ni l'SQL Server ni l'Oracle no depassen aquest valor i faria que disminuís el nombre total de camps possible a la taula. El **joc de caràcters** (ANSI, OEM, UTF-8) amb què s'escriu un nom estès és el coherent amb el definit al byte 29 de la capçalera de la taula (el mateix amb què s'escriu els camps C). Nota per al cas de la DBF clàssica: Tot i que els caràcters especials (lletres accentuades, etc) en els noms de camps no són conformes a l'estàndard de la DBF clàssica, l'ArcGIS i el QGIS creen i toleren accentuar noms de camps en DBF clàssiques. La filosofia al MiraMon és, en una DBF clàssica, no generar en els noms dels camps lletres accentuades, etc, ja que ens sembla massa lluny de l'estàndard, però en cas que hi hagi aquests caràcters estesos, tolerar-los i mostrar-los d'acord amb el codi de caràcters del byte 29 de la capçalera.

Per a informació addicional, consulteu Bases de dades al MiraMon.