

**ESTUDIO:**

**"CREACION DE UN BANCO DE  
DATOS HIDROGEOLOGICOS"**

**LISTADOS DE PROGRAMAS**

**TOMO I**

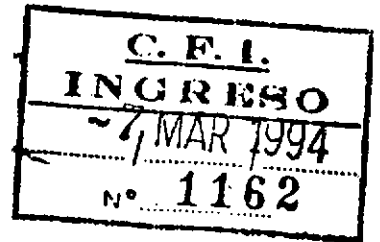
**EXPERTO: Ing. Daniel Fabián Gazzola**

0/0.151  
G II  
I

**Provincia del Neuquén**

Neuquén, 03 de Marzo de 1.994.

Señor  
Director  
Consejo Federal de inversiones  
Ing. Juan José Ciácerá  
S \_\_\_\_\_ / \_\_\_\_\_ D:



Me dirijo a Usted a los efectos de remitirle el listado completo de los programas que conforman el Sistema del Banco de Datos Hidrogeológicos, cuyo desarrollo oportunamente me fuera encomendado.

El presente envío constituye solo una parte de los informes que adeudo, al haber vencido los plazos de presentación respectivos, pero la magnitud de la tarea, incluyendo las correcciones y mejoras que me fueran solicitadas por los sectores destinatarios del sistema, han superado ampliamente, por su magnitud, los tiempos de desarrollo originalmente previstos, tal como podrá observar en los listados adjuntos, los que contienen más de 21.000 líneas de código de programación.

Aprovecho la oportunidad para manifestarle mi voluntad de culminar a la brevedad el presente estudio, cumplimentando todas las tareas e informes acordados, para lo cual continuare poniendo todo mi empeño y dedicación.

Saludo a Usted con atenta consideración.

DANIEL G. ZOLA  
ING. INDUSTRIAL  
C. F. I.  
MAT. 1

## 1. ARCHIVOS DE ENCABEZAMIENTO DE PROGRAMAS

1.1. ACONDAP.CH.....	I	1
1.2. AGUASUP.CH.....	I	2
1.3. AMETCAR.CH.....	I	3
1.4. ATIPOFTE.CH.....	I	4
1.5. ATIPOUSO.CH.....	I	5
1.6. CLIENTES.CH.....	I	6
1.7. CONDAP.CH.....	I	7
1.8. DEPTO.CH.....	I	8
1.9. DETERID.CH.....	I	9
1.10. ENSAADMI.CH.....	I	10
1.11. ENSABOMB.CH.....	I	11
1.12. ENSAHFDO.CH.....	I	12
1.13. ESTACID.CH.....	I	13
1.14. ESTHIDVA.CH.....	I	14
1.15. ESTHIDVE.CH.....	I	15
1.16. ESTHIDVM.CH.....	I	16
1.17. ESTMETVA.CH.....	I	17
1.18. ESTMETVE.CH.....	I	18
1.19. ESTMETVM.CH.....	I	19
1.20. GEOELDAT.CH.....	I	20
1.21. GEOELID.CH.....	I	21
1.22. GEOELRES.CH.....	I	22
1.23. GEOELSON.CH.....	I	23
1.24. IDFUENTE.CH.....	I	24
1.25. IDPOZO.CH.....	I	25
1.26. LOCAL.CH.....	I	26
1.27. MANANAFO.CH.....	I	27
1.28. MANANDAT.CH.....	I	28
1.29. MENU.CH.....	I	29
1.30. MUESTRAS.CH.....	I	39
1.31. POZOCRON.CH.....	I	40
1.32. POZODIPE.CH.....	I	41
1.33. POZODIPF.CH.....	I	42
1.34. POZOGEOF.CH.....	I	43
1.35. POZOGEOH.CH.....	I	44
1.36. POZOGEOL.CH.....	I	45
1.37. PROTIPO.CH.....	I	46
1.38. PROTIPOD.CH.....	I	47
1.39. RESCAR.CH.....	I	48
1.40. RESDETER.CH.....	I	49

## 2. ARCHIVOS DE PROGRAMAS

2.1. AGUASUP.PRG.....	I	50
2.2. AGUASUPI.PRG.....	I	54
2.3. ANALINF1.PRG.....	I	56
2.4. ANALINF2.PRG.....	I	67
2.5. ANALRES.PRG.....	I	72
2.6. ANALRES.D.PRG.....	I	77

2.7. AUXDEPTI.PRG.....	I	82
2.8. AUXDEPTO.PRG.....	I	84
2.9. AUXLOCAL.PRG.....	I	87
2.10. AUXLOCAL.PRG.....	I	89
2.12. CFUNC.C.....	I	92
2.13. CLIENTE1.PRG.....	I	95
2.14. CLIENTES.PRG.....	I	97
2.15. CONDAP.PRG.....	I	100
2.16. CONDAP1.PRG.....	I	105
2.17. DETERID.PRG.....	I	107
2.18. DETERIDI.PRG.....	I	111
2.19. ENSAADMI.PRG.....	I	113
2.20. ENSABOMB.PRG.....	I	117
2.21. ENSAHFDO.PRG.....	I	123
2.22. ESTACID.PRG.....	I	127
2.23. ESTACIDI.PRG.....	I	131
2.24. ESTHIDVA.PRG.....	I	133
2.25. ESTHIDVE.PRG.....	I	137
2.26. ESTHIDVI.PRG.....	I	142
2.27. ESTHIDVM.PRG.....	I	157
2.28. ESTMETVA.PRG.....	I	181
2.29. ESTMETVE.PRG.....	I	185
2.30. ESTMETVI.PRG.....	II	190
2.31. ESTMETVM.PRG.....	II	207
2.32. EXPOANAG.PRG.....	II	238
2.33. EXPOGEOE.PRG.....	II	247
2.34. GEOELDAT.PRG.....	II	253
2.35. GEOELID.PRG.....	II	259
2.36. GEOELIDI.PRG.....	II	263
2.37. GEOELREI.PRG.....	II	265
2.38. GEOELRES.PRG.....	II	271
2.39. GEOELSOI.PRG.....	II	283
2.40. GEOELSON.PRG.....	II	285
2.41. IDFUENTE.PRG.....	II	293
2.42. IDFUENTI.PRG.....	II	299
2.43. IDPOZO.PRG.....	II	301
2.44. IDPOZOI.PRG.....	II	305
2.45. IDPOZOIM.PRG.....	II	307
2.46. IDPOZOIN.PRG.....	II	309
2.47. MANANAFI.PRG.....	II	311
2.48. MANANAFO.PRG.....	II	313
2.49. MANANDAI.PRG.....	II	316
2.50. MANANDAT.PRG.....	II	318
2.51. MENU.PRG.....	II	322
2.52. MUESTRAD.PRG.....	II	327
2.53. MUESTRAI.PRG.....	II	332
2.54. MUESTRAS.PRG.....	II	334
2.55. PF01.PRG.....	II	345
2.56. PF02.PRG.....	II	366

2.57. POZOCRON.PRG .....	II 378
2.58. POZODIPE.PRG .....	II 382
2.59. POZODIPF.PRG .....	III 386
2.60. POZOFICH.PRG .....	III 391
2.61. POZOGEOF.PRG .....	III 412
2.62. POZOGEOH.PRG .....	III 416
2.63. POZOGEOH.PRG .....	III 420
2.64. PROTIPI.PRG .....	III 423
2.65. PROTIPO.PRG .....	III 425
2.66. PROTIPOD.PRG .....	III 428
2.67. PROTIPOI.PRG .....	III 434
2.68. RECUPERA.PRG .....	III 436
2.69. RESAPI.PRG .....	III 439
2.70. RESCAR.PRG .....	III 448
2.71. RESCARI.PRG .....	III 454
2.72. RESDETED.PRG .....	III 459
2.73. RESDETER.PRG .....	III 464
2.74. TABLAGWS.PRG .....	III 472
2.75. TABLGWSI.PRG .....	III 476
3. ARCHIVOS DE FORMATO DE REPORTES	
3.1. AGUASUP.FRM .....	III 478
3.2. ANALINF2.FRM .....	III 481
3.3. CLIENTES.FRM .....	III 484
3.4. CONDAP.FRM .....	III 486
3.5. DEPTO.FRM .....	III 488
3.6. DETERID.FRM .....	III 490
3.7. ESTACID.FRM .....	III 492
3.8. GEOELID.FRM .....	III 496
3.9. GEOELSON.FRM .....	III 499
3.10. IDFUENTE.FRM .....	III 502
3.11. IDPOZO.FRM .....	III 506
3.12. IDPOZOM.FRM .....	III 511
3.13. IDPOZON.FRM .....	III 513
3.14. LOCAL.FRM .....	III 516
3.15. MANANAFO.FRM .....	III 518
3.16. MANANDAT.FRM .....	III 520
3.17. MUESTRAI.FRM .....	III 523
3.18. PROTIPO.FRM .....	III 526
3.19. PROTIPOD.FRM .....	III 528
3.20. TABLAGWS.FRM .....	III 530
4. ARCHIVOS PARA COMPILACIÓN Y ENLACE DEL SISTEMA	
4.1. CFI.RMK .....	III 533
4.2. CFI.RLK .....	III 537

```
00001 : *****
00002 : *
00003 : *      Archivo : ACONDAP.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 21 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de usos.
00011 : *
00012 : *****
00013 :
00014 : PRIVATE aCondAp := { { "A", "D", "R", "G", "I", "S", "O" },;
00015 : { "Agua Potable", "Doméstico", "Riego", "Ganadero",;
00016 : "Industrial", "Recuperación Secundaria", "Otros" } }
00017 :
00018 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : AGUASUP.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 21 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definición de índices para la base de aguas
00011 : *      superficiales (AGUASUP.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndAgSup := { { "AGUASUP", NIL },;
00016 : { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : AMETCAR.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    21 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Definicion de metodos de caracterizacion
00011 : *      quimica de las aguas.
00012 : *
00013 : *****
00014 :
00015 : PRIVATE aMetCar := { {"P","S","W","O" },;
00016 :                      {"PIPER","STIFF","WILCOX","OTRO"};
00017 :                      }
00018 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo : ATIPOFTE.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    21 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Definición de fuentes hidrologicas.
00011 : *
00012 : *****
00013 :
00014 : PRIVATE aTipoFte := { { "PB", "PO", "PR", "PS", "PC",;
00015 :                        "AR", "AA", "AL", "AE", "AU", "AC",;
00016 :                        "MU", "MM", " " },;
00017 :                        { "Pozo de Bombeo", "Pozo de Observación",;
00018 :                        "Pozo de Reconocimiento", "Pozo Somero",;
00019 :                        "Pozo Cavado", "Rio", "Arroyo", "Lago",;
00020 :                        "Lago Artificial o Embalse", "Laguna",;
00021 :                        "Colector de Manantiales", "Manantial Unitario",;
00022 :                        "Manantial Multiple", " " } }
00023 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ATIPOUSO.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:      Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:      21 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definición de tipos de usos.
00011 : *
00012 : *****
00013 :
00014 : PRIVATE aTiposUso := { "ADRGISO","Agua Potable","Dom.stico","Riego", ;
00015 :      "Ganadero", "Industrial", "Recuperación Secundaria", "Otros" }
00016 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : CLIENTES.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    21 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definición de índices para la base auxiliar
00011 : *      de clientes (CLIENTES.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndClie := { { "CLIENTES", NIL },;
00016 :                    { "CODIGO", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : CONDAP.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 30 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de condi-
00011 : *      ciones de aptitud para usos (CONDAP.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndCondAp := { ( "CONDAP", NIL ),;
00016 :                      { "CODUSO+DETERM", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo :  DEPTO.CH
00004 : *      Sistema :  Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa:  menu.exe
00006 : *      Autor:     Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:     30 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Definición de índices para la base auxiliar
00011 : *      de Departamentos (DEPTO.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndDepto := { { "DEPTO", NIL } };
00016 :                  { "DEPTO", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : DETERID.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de defi-
00011 : *      nicion de determinaciones (DETERID.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndDeterID := { { "DETERID", NIL },;
00016 :                       { "DETERM", NIL } }
00017 :
```

Programa: ENSAADMI.CH - 1

```
00001 : *****
00002 : *
00003 : *      Archivo :  ENSAADMI.CH      *
00004 : *      Sistema :  Sistema Banco de Datos Hidrogeológicos *
00005 : *      Programa:  menu.exe          *
00006 : *      Autor:     Ing. Daniel Fabián GAZZOLA *
00007 : *      Fecha:     04 de Setiembre de 1.993 *
00008 : *
00009 : *      Descripción: *
00010 : *      Definicion de indices para la base de datos *
00011 : *      de ensayos de admision (ENSAADMI.DBF). *
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEAdmi := { { "ENSAADMI", NIL },;
00016 :                      { "DEPTO+TPOFUENTE+NROSERIE+DTOS(FECHA)+TIPO", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo :  ENSABOMB.CH      *
00004 : *      Sistema :  Sistema Banco de Datos Hidrogeológicos *
00005 : *      Programa:  menu.exe          *
00006 : *      Autor:     Ing. Daniel Fabián GAZZOLA *
00007 : *      Fecha:     15 de Julio de 1.993 *
00008 : *
00009 : *      Descripción: *
00010 : *      Definicion de indices para la base de datos *
00011 : *      de ensayos de bombeo (ENSABOMB.DBF). *
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEBomb := { { "ENSABOMB", NIL },;
00016 :                      { "DEPTO+TPOFUENTE+NROSERIE+DTOS(FECHA)+TIPO", NIL } }
00017 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo :  ENSAHFDO.CH      *
00004 : *      Sistema :  Sistema Banco de Datos Hidrogeológicos *
00005 : *      Programa:  menu.exe          *
00006 : *      Autor:     Ing. Daniel Fabián GAZZOLA *
00007 : *      Fecha:     02 de Julio de 1.993 *
00008 : *
00009 : *      Descripción: *
00010 : *      Definicion de indices para la base de ensayos *
00011 : *      hidraulicos de fin de obra (ENSAHFDO.DBF). *
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEHFDO := { { "ENSAHFDO", NIL },;
00016 :                      { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTACID.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:   Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:   30 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de iden-
00011 : *      tificacion de estaciones hidrometeor. (ESTACID.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEstId := { { "ESTACID", "ESTACIDT", NIL },;
00016 : *      { "DEPTO+TIPOEST+CODIGOEST",;
00017 : *      "TIPOEST+DEPTO+CODIGOEST", NIL } }
00018 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTHIDVA.CH      *
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos *
00005 : *      Programa: menu.exe        *
00006 : *      Autor:   Ing. Daniel Fabián GAZZOLA *
00007 : *      Fecha:   31 de Agosto de 1.993 *
00008 : *
00009 : *      Descripción: *
00010 : *      Definicion de indices para la base de valores *
00011 : *      anuales de estaciones hidrometeor. (ESTHIDVA.DBF). *
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEHVA := { { "ESTHIDVA", NIL },;
00016 :                    { "DEPTO+CODIGOEST", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTHIDVE.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    01 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de valores
00011 : *      extremos de estaciones hidromet. (ESTHIDVE.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEHVE := { { "ESTHIDVE", NIL },;
00016 :                    { "DEPTO+CODIGOEST", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTHIDVM.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    31 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de valores
00011 : *      medios mensuales de estaciones hidrometeor. (ESTHIDVM.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEHVM := { { "ESTHIDVM", NIL },;
00016 :                    { "DEPTO+CODIGOEST", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTMETVA.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 31 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de valores
00011 : *      medios anuales de estaciones meteor. (ESTMETVA.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEMVA := { { "ESTMETVA", NIL },;
00016 : { "DEPTO+CODIGOEST", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTMETVE.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 30 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de valores
00011 : *      extremos de estaciones meteor. (ESTMETVE.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEMVE := { { "ESTMETVE", NIL },;
00016 : { "DEPTO+CODIGOEST", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTMETVM.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 31 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de valores
00011 : *      medios mensuales de estaciones meteor. (ESTMETVM.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndEMVM := { { "ESTMETVM", NIL },;
00016 : *      { "DEPTO+CODIGOEST", NIL } }
00017 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo : GEOELDAT.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    22 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definición de índices para la base de datos
00011 : *      de estudios geoelectricos (GEOELDAT.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndGeoEID := { { "GEOELDAT", NIL },;
00016 :                      { "DEPTO+ESTUDIO+LINEAEST+NROSONDEO+" ;
00017 :                      "STR(DISTMN,8,2)+STR(DISTAB,8,2)", NIL } }
00018 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo :  GEOELID.CH
00004 : *      Sistema :  Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa:  menu.exe
00006 : *      Autor:     Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:     21 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de iden-
00011 : *      tificacion de estudios geoelectricos (GEOELID.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndGeoEII := { { "GEOELID", NIL },;
00016 :                       { "DEPTO+ESTUDIO", NIL } }
00017 :
```

Programa: GEOELRES.CH - 1

```
00001 : *****
00002 : *
00003 : *      Archivo : GEOELRES.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definición de índices para la base de resul-
00011 : *      tados de estudios geoelectricos (GEOELRES.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndGeoEIR := { { "GEOELRES", NIL },;
00016 :                      { "DEPTO+ESTUDIO+LINEAEST+NROSONDEO", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : GEOELSON.CH      *
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos *
00005 : *      Programa: menu.exe         *
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA *
00007 : *      Fecha:    21 de Agosto de 1.993 *
00008 : *
00009 : *      Descripción: *
00010 : *      Definicion de indices para la base de iden- *
00011 : *      tificacion de sondeos geoelectricos (GEOELSON.DBF). *
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndGeoEIS := { { "GEOELSON", NIL },;
00016 : *      { "DEPTO+ESTUDIO+LINEAEST+NROSONDEO", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo :  IDFUENTE.CH
00004 : *      Sistema :  Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa:  menu.exe
00006 : *      Autor:     Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:     01 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Definición de índices para la base de Fuentes
00011 : *      acuíferas (IDFUENTE.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndIdFte := { { "IDFUENTE", NIL },;
00016 :                      { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : IDPOZO.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 30 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de iden-
00011 : *      tificacion de pozos (IDPOZO.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndIdPozo := { { "IDPOZO", NIL },;
00016 : { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```

Programa: LOCAL.CH - I

```
00001 : *****
00002 : *
00003 : *      Archivo : LOCAL.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menú.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 30 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definición de índices para la base auxiliar
00011 : *      de localidades (LOCAL.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndLocal := { { "LOCAL", "LOCALCOD", NIL },;
00016 :                      { "DEPTO+CODLOC", "CODLOC", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo :  MANANAFO.CH      *
00004 : *      Sistema :  Sistema Banco de Datos Hidrogeológicos *
00005 : *      Programa:  menu.exe          *
00006 : *      Autor:     Ing. Daniel Fabián GAZZOLA *
00007 : *      Fecha:     03 de Julio de 1.993 *
00008 : *
00009 : *      Descripción: *
00010 : *      Definición de índices para la base de datos *
00011 : *      de aforos de manantiales (MANANAFO.DBF). *
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndMaAfo := { { "MANANAFO", NIL },;
00016 :                      { "DEPTO+TPOFUENTE+NROSERIE+DTOS(FECHA AFO)", NIL } }
00017 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo : MANANDAT.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    03 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de iden-
00011 : *      tificacion de manantiales (MANANDAT.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndMaDat := { { "MANANDAT", NIL },;
00016 :                      { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : MENU.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 14 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definición del árbol de menu.
00011 : *
00012 : *****
00013 :
00014 : asbm111 := { "AÚMuestras",;
00015 :             "Ingresa datos de muestras y solicitudes de análisis",;
00016 :             {|| muestras() }, .T.;
00017 :             }
00018 :
00019 : asbm112 := { "BÚResultados",;
00020 :             "Ingresa Resultados de las Determinaciones",;
00021 :             {|| resdeter() }, .T.;
00022 :             }
00023 :
00024 : asbm113 := { "CÚProtocolos",;
00025 :             "Definición de Protocolos Tipo",;
00026 :             {|| protipo() }, .T.;
00027 :             }
00028 :
00029 : asbm114 := { "DÚDeterminaciones",;
00030 :             "Definición de las Determinaciones",;
00031 :             {|| deterid() }, .T.;
00032 :             }
00033 :
00034 : asbm115 := { "EÚAptitud",;
00035 :             "Definición de los Parametros de Aptitud para Usos",;
00036 :             {|| condap() }, .T.;
00037 :             }
00038 :
00039 : asbm116 := { "AÚResultados",;
00040 :             "Resultados de Caracterización Química de las Aguas",;
00041 :             {|| rescar() }, .T.;
00042 :             }
00043 :
00044 : asbm1162 := { "BÚEquiv. GWS",;
00045 :             "Tabla de Equivalencias con el Sistema GWS",;
00046 :             {|| tablagws() }, .T.;
00047 :             }
00048 :
```

```
00049 : asbm116 := { "FúCaracterizaciøn .",;
00050 :             "Caracterizaciøn Quimica de las Aguas",;
00051 :             {|| INISUBMENU( { { asbm1161, asbm1162 },;
00052 :             40, 14 }, level3 ) }, .T.;
00053 :         }
00054 :
00055 : asbm11 := { "AúAn lisis de Aguas .",;
00056 :             "Datos de An lisis Fisicoquimico, Quimicos y "+;
00057 :             "Bacterioløgicos",;
00058 :             {|| INISUBMENU( { { asbm111, asbm112, asbm113, asbm114,;
00059 :             asbm115, asbm116 }, 30, 7 }, level3 ) }, .T.;
00060 :         }
00061 :
00062 : asbm12 := { "BúFuentes Acuíferas",;
00063 :             "Datos Identificatorios de las Fuentes Acuíferas",;
00064 :             {|| idfuente() }, .T.;
00065 :         }
00066 :
00067 : asbm13 := { "CúAguas Superficiales",;
00068 :             "Datos de las Fuentes Acuíferas Superficiales",;
00069 :             {|| aguasup() }, .T.;
00070 :         }
00071 :
00072 : asbm141 := { "AúIdentificaciøn",;
00073 :             "Datos Identificatorios de los Manantiales",;
00074 :             {|| manandat() }, .T.;
00075 :         }
00076 :
00077 : asbm142 := { "BúAforos",;
00078 :             "Datos de Aforos de Manantiales",;
00079 :             {|| mananafo() }, .T.;
00080 :         }
00081 :
00082 : asbm14 := { "DúManantiales .",;
00083 :             "Datos de Manantiales",;
00084 :             {|| INISUBMENU( { { asbm141, asbm142 },;
00085 :             30, 10 }, level3 ) }, .T.;
00086 :         }
00087 :
00088 : asbm151 := { "AúIdentificaciøn",;
00089 :             "Datos Identificatorios de las Estaciones "+;
00090 :             "Hidrometeoroløgicas",;
00091 :             {|| estacid() }, .T.;
00092 :         }
00093 :
00094 : asbm1521 := { "AúValores Medios Mensuales",;
00095 :             "Valores Medios Mensuales de Registros Meteoroløgicos",;
00096 :             {|| estmetvm() }, .T.;
```

```
00097 :      }
00098 :
00099 : asbm1522 := { "BúValores Medios Anuales",;
00100 :      "Valores Medios Anuales de Registros Meteorológicos",;
00101 :      {|| estmetva() }, .T.;
00102 :      }
00103 :
00104 : asbm1523 := { "CúValores Extremos Anuales",;
00105 :      "Valores Extremos Anuales de Registros Meteorológicos",;
00106 :      {|| estmetve() }, .T.;
00107 :      }
00108 :
00109 : asbm152 := { "BúRegistros Meteorológicos .",;
00110 :      "Registros de Estaciones Meteorológicas",;
00111 :      {|| INISUBMENU( { { asbm1521, asbm1522, asbm1523 },;
00112 :      30, 14 }, level3 ) }, .T.;
00113 :      }
00114 :
00115 : asbm1531 := { "AúValores Medios Mensuales",;
00116 :      "Valores Medios Mensuales de Registros Hidrométricos",;
00117 :      {|| esthidvm() }, .T.;
00118 :      }
00119 :
00120 : asbm1532 := { "BúValores Medios Anuales",;
00121 :      "Valores Medios Anuales de Registros Hidrométricos",;
00122 :      {|| esthidva() }, .T.;
00123 :      }
00124 :
00125 : asbm1533 := { "CúValores Extremos Anuales",;
00126 :      "Valores Extremos Anuales de Registros Hidrométricos",;
00127 :      {|| esthidve() }, .T.;
00128 :      }
00129 :
00130 : asbm153 := { "CúRegistros Hidrométricos .",;
00131 :      "Registros de Estaciones Hidrométricas",;
00132 :      {|| INISUBMENU( { { asbm1531, asbm1532, asbm1533 },;
00133 :      30, 15 }, level3 ) }, .T.;
00134 :      }
00135 :
00136 : asbm15 := { "EúEstaciones Hidrometeorológicas",;
00137 :      "Datos de Estaciones Hidrometeorológicas",;
00138 :      {|| INISUBMENU( { { asbm151, asbm152, asbm153 },;
00139 :      30, 11 }, level3 ) }, .T.;
00140 :      }
00141 :
00142 : asbm161 := { "AúIdentificación",;
00143 :      "Datos Identificatorios de los Pozos",;
00144 :      {|| idpozo() }, .T.;
```

```
00145 :      }
00146 :
00147 : asbm1621 := { "AúAdmisión",;
00148 :      "Datos de Ensayos de Admisión",;
00149 :      {|| ensaadmi() }, .T.;
00150 :      }
00151 :
00152 : asbm1622 := { "BúBombeo",;
00153 :      "Datos de Ensayos de Bombeo",;
00154 :      {|| ensabomb() }, .T.;
00155 :      }
00156 :
00157 : asbm1623 := { "CúFin de Obra",;
00158 :      "Datos de Ensayos Hidráulico de Fin de Obra",;
00159 :      {|| ensahfdo() }, .T.;
00160 :      }
00161 :
00162 : asbm162 := { "BúEnsayos      .",;
00163 :      "Datos de Ensayos en Pozos y Perforaciones",;
00164 :      {|| INISUBMENU( { { asbm1621, asbm1622, asbm1623 },;
00165 :                      40, 15 }, level3 ) }, .T.;
00166 :      }
00167 :
00168 : asbm163 := { "CúDiseño Pozo",;
00169 :      "Datos de Diseño del Pozo",;
00170 :      {|| pozodipe() }, .T.;
00171 :      }
00172 :
00173 : asbm164 := { "DúGeofísica",;
00174 :      "Datos Geofísicos del Pozo",;
00175 :      {|| pozogeof() }, .T.;
00176 :      }
00177 :
00178 : asbm165 := { "EúGeología",;
00179 :      "Datos Geológicos del Pozo",;
00180 :      {|| pozogeol() }, .T.;
00181 :      }
00182 :
00183 : asbm16 := { "FúPozos y Perforaciones .",;
00184 :      "Datos de Pozos y Perforaciones",;
00185 :      {|| INISUBMENU( { { asbm161, asbm162, asbm163,;
00186 :                      asbm164, asbm165 },;
00187 :                      30, 12 }, level3 ) }, .T.;
00188 :      }
00189 :
00190 : asbm171 := { "AúIdentificación Estudios",;
00191 :      "Datos Identificatorios de Estudios Geológicos",;
00192 :      {|| geoelid() }, .T.;
```

```
00193 :      }
00194 :
00195 : asbm172 := { "BúDatos de Sondeos",;
00196 :             "Datos de Sondeos Geoel.ctracos",;
00197 :             {|| geoelson() }, .T.;
00198 :             }
00199 :
00200 : asbm173 := { "CúResultados de Sondeos",;
00201 :             "Resultados de Sondeos Geoel.ctracos",;
00202 :             {|| geoelres() }, .T.;
00203 :             }
00204 :
00205 : asbm17 := { "GúEstudios Geoel.ctracos .",;
00206 :            "Datos de Estudios Geoel.ctracos",;
00207 :            {|| INISUBMENU( { { asbm171, asbm172, asbm173 },;
00208 :                           30, 13 }, level3 ) }, .T.;
00209 :            }
00210 :
00211 : asbm181 := { "AúDepartamentos",;
00212 :            "Identificaciøn de los Departamentos Provinciales",;
00213 :            {|| auxdepto() }, .T.;
00214 :            }
00215 :
00216 : asbm182 := { "BúLocalidades",;
00217 :            "Identificaciøn de las Localidades",;
00218 :            {|| auxlocal() }, .T.;
00219 :            }
00220 :
00221 : asbm183 := { "CúClientes",;
00222 :            "Identificaciøn de los Clientes",;
00223 :            {|| clientes() }, .T.;
00224 :            }
00225 :
00226 : asbm18 := { "HúArchivos Auxiliares .",;
00227 :            "Datos Auxiliares",;
00228 :            {|| INISUBMENU( { { asbm181, asbm182, asbm183 },;
00229 :                           30, 14 }, level3 ) }, .T.;
00230 :            }
00231 :
00232 : asbm1 := { { asbm11, asbm12, asbm13, asbm14, asbm15, asbm16, asbm17,;
00233 :            asbm18 }, 15, 5 }
00234 :
00235 : asbm211 := { "AúMuestras",;
00236 :            "Listado de Entrada de Estudios",;
00237 :            {|| muestrai() }, .T.;
00238 :            }
00239 :
00240 : asbm2121 := { "AúInforme 1",;
```

```
00241 :      "Informe Estadístico 1",;
00242 :      {|| analinf1() }, .T.;
00243 :    }
00244 :
00245 : asbm2122 := { "AúInforme 2",;
00246 :      "Informe Estadístico 2",;
00247 :      {|| analinf2() }, .T.;
00248 :    }
00249 :
00250 : asbm212 := { "BúEstadísticas      .",;
00251 :      "Informes Estadístico de los Analisis",;
00252 :      {|| INISUBMENU( { { asbm2121, asbm2122 }, 60, 10 },;
00253 :        level3 ) }, .T.;
00254 :    }
00255 :
00256 : asbm2131 := { "AúIdentificación",;
00257 :      "Listado de Identificación de Protocolos Tipo",;
00258 :      {|| protipi() }, .T.;
00259 :    }
00260 :
00261 : asbm2132 := { "BúConformación",;
00262 :      "Listado de Conformación de Protocolos Tipo",;
00263 :      {|| protipoi() }, .T.;
00264 :    }
00265 :
00266 : asbm213 := { "CúProtocolos      .",;
00267 :      "Listados de Protocolos Tipo",;
00268 :      {|| INISUBMENU( { { asbm2131, asbm2132 }, 60, 11 },;
00269 :        level3 ) }, .T.;
00270 :    }
00271 :
00272 : asbm214 := { "DúDeterminaciones",;
00273 :      "Definición de las Determinaciones",;
00274 :      {|| deteridi() }, .T.;
00275 :    }
00276 :
00277 : asbm2151 := { "AúParametros",;
00278 :      "Definición de los Parametros de Aptitud para Usos",;
00279 :      {|| condapi() }, .T.;
00280 :    }
00281 :
00282 : asbm2152 := { "BúResultados",;
00283 :      "Resultados de Aptitud para Usos de las Aguas",;
00284 :      {|| resapi() }, .T.;
00285 :    }
00286 :
00287 : asbm215 := { "EúAptitud      .",;
00288 :      "Informes de Aptitud de Aguas para Usos",;
```

```
00289 :          {|| INISUBMENU( { { asbm2151, asbm2152 }, 60, 13 },;  
00290 :          level3 ) }, .T.;;  
00291 :      }  
00292 :  
00293 : asbm2161 := { "AúResultados",;  
00294 :          "Resultados de Caracterizaciòn Quimica de las Aguas",;  
00295 :          {|| rescari() }, .T.;;  
00296 :      }  
00297 :  
00298 : asbm2162 := { "BúEquiv. GWS",;  
00299 :          "Tabla de Equivalencias con el Sistema GWS",;  
00300 :          {|| tablgwsi() }, .T.;;  
00301 :      }  
00302 :  
00303 : asbm216 := { "FúCaracterizaciòn .",;  
00304 :          "Caracterizaciòn Quimica de las Aguas",;  
00305 :          {|| INISUBMENU( { { asbm2161, asbm2162 }, 60, 14 },;  
00306 :          level3 ) }, .T.;;  
00307 :      }  
00308 :  
00309 : asbm21 := { "AúAn lisis de Aguas .",;  
00310 :          "Datos de An lisis Fisicoquimico, Quimicos y "+  
00311 :          "Bacteriológicos",;  
00312 :          {|| INISUBMENU( { { asbm211,asbm212,asbm213,asbm214,;  
00313 :              asbm215,asbm216 }, 50, 7 }, level3 ) }, .T.;;  
00314 :      }  
00315 :  
00316 : asbm22 := { "BúFuentes Acuíferas",;  
00317 :          "Datos Identificatorios de las Fuentes Acuíferas",;  
00318 :          {|| idfuenti() }, .T.;;  
00319 :      }  
00320 :  
00321 : asbm23 := { "CúAguas Superficiales",;  
00322 :          "Datos de las Fuentes Acuíferas Superficiales",;  
00323 :          {|| aguasupi() }, .T.;;  
00324 :      }  
00325 :  
00326 : asbm241 := { "AúIdentificaciòn",;  
00327 :          "Listado de Identificaciòn de los Manantiales",;  
00328 :          {|| manandai() }, .T.;;  
00329 :      }  
00330 :  
00331 : asbm242 := { "BúAforos",;  
00332 :          "Listado de Aforos de Manantiales",;  
00333 :          {|| mananafi() }, .T.;;  
00334 :      }  
00335 :  
00336 : asbm24 := { "DúManantiales .",;
```



```
00337 :      "Datos de Manantiales",;
00338 :      {|| INISUBMENU( { { asbm241, asbm242 },;
00339 :                      50, 10 }, level3 ) }, .T.;
00340 :      }
00341 :
00342 : asbm251 := { "AúIdentificaciøn",;
00343 :      "Datos Identificatorios de las Estaciones "+;
00344 :      "Hidrometeorológicas",;
00345 :      {|| estacidi() }, .T.;
00346 :      }
00347 :
00348 : asbm252 := { "BúRegistros Meteorológicos",;
00349 :      "Registros de Estaciones Meteorológicas",;
00350 :      {|| estmetvi() }, .T.;
00351 :      }
00352 :
00353 : asbm253 := { "CúRegistros Hidrométricas",;
00354 :      "Registros de Estaciones Hidrométricas",;
00355 :      {|| estmetvi() }, .T.;
00356 :      }
00357 :
00358 : asbm25 := { "EúEstaciones Hidrometeor.",;
00359 :      "Datos de Estaciones Hidrometeorológicas",;
00360 :      {|| INISUBMENU( { { asbm251, asbm252, asbm253 },;
00361 :                      50, 11 }, level3 ) }, .T.;
00362 :      }
00363 :
00364 : asbm261 := { "AúIdentificaciøn",;
00365 :      "Datos Identificatorios de los Pozos",;
00366 :      {|| idpozoi() }, .T.;
00367 :      }
00368 :
00369 : asbm262 := { "BúÍndice de Pozos",;
00370 :      "Impresiøn del Índice de Pozos",;
00371 :      {|| idpozoin() }, .T.;
00372 :      }
00373 :
00374 : asbm263 := { "CúFichas de Pozo",;
00375 :      "Impresiøn de las Fichas de Datos de Pozos",;
00376 :      {|| pozofich() }, .T.;
00377 :      }
00378 :
00379 : asbm264 := { "DúInf. para Mapa",;
00380 :      "Impresiøn de Informe para Mapa Hidrogeológico",;
00381 :      {|| idpozoinm() }, .T.;
00382 :      }
00383 :
00384 : asbm26 := { "FúPozos y Perforaciones .",;
00385 :      "Fichas de Pozos y Perforaciones",;
00386 :      {|| idpozoinm() }, .T.;
00387 :      }
```

```
00385 :      "Datos de Pozos y Perforaciones",;
00386 :      {|| INISUBMENU( { { asbm261, asbm262, asbm263, asbm264 },;
00387 :                      50, 12 }, level3 ) }, .T.;
00388 :      }
00389 :
00390 : asbm271 := { "AúIdentificaciøn Estudios",;
00391 :             "Listado de Estudios Geoel.ctrìcos",;
00392 :             {|| geoelidi() }, .T.;
00393 :             }
00394 :
00395 : asbm272 := { "BúDatos de Sondeos",;
00396 :             "Listado de Sondeos Geoel.ctrìcos",;
00397 :             {|| geoelsoi() }, .T.;
00398 :             }
00399 :
00400 : asbm273 := { "CúResultados de Sondeos",;
00401 :             "Resultados de Sondeos Geoel.ctrìcos",;
00402 :             {|| geoelrei() }, .T.;
00403 :             }
00404 :
00405 : asbm27 := { "GúEstudios Geoel.ctrìcos .",;
00406 :            "Datos de Estudios Geoel.ctrìcos",;
00407 :            {|| INISUBMENU( { { asbm271, asbm272, asbm273 },;
00408 :                          50, 13 }, level3 ) }, .T.;
00409 :            }
00410 :
00411 : asbm281 := { "AúDepartamentos",;
00412 :            "Identificaciøn de los Departamentos Provinciales",;
00413 :            {|| auxdepti() }, .T.;
00414 :            }
00415 :
00416 : asbm282 := { "BúLocalidades",;
00417 :            "Identificaciøn de las Localidades",;
00418 :            {|| auxlocai() }, .T.;
00419 :            }
00420 :
00421 : asbm283 := { "CúClientes",;
00422 :            "Identificaciøn de los Clientes",;
00423 :            {|| clientei() }, .T.;
00424 :            }
00425 :
00426 : asbm28 := { "HúArchivos Auxiliares .",;
00427 :            "Datos Auxiliares",;
00428 :            {|| INISUBMENU( { { asbm281, asbm282, asbm283 },;
00429 :                          50, 14 }, level3 ) }, .T.;
00430 :            }
00431 :
00432 : asbm2 := { { asbm21, asbm22, asbm23, asbm24, asbm25, asbm26, asbm27,;
```

```
00433 :          asbm28 }, 30, 5 }
00434 :
00435 : asbm31 := { "AúCaracterizaciøn Quimica de Aguas",;
00436 :          "Indices de Steff, Pepper y Schoeller Berkaloff",;
00437 :          {|| ExpoAnAg() }, .T.;
00438 :          }
00439 :
00440 : asbm32 := { "BúDatos de Estudios Geoel.ctrìcos",;
00441 :          "Crea archivos para programa de Curvas Geoel.ctrìcas",;
00442 :          {|| ExpoGoe() }, .T.;
00443 :          }
00444 :
00445 : asbm3  := { { asbm31, asbm32 }, 47, 5 }
00446 :
00447 : asbm41 := { "AúRecupera Caidas",;
00448 :          "Recupera el Sistema despues de Fallas de "+;
00449 :          "Energia o de la Computadora", ;
00450 :          {|| recupera() }, .T.;
00451 :          }
00452 :
00453 : asbm42 := { "BúReorganiza Archivos",;
00454 :          "Comprime los Archivos", ;
00455 :          {||  NIL  }, .F.;
00456 :          }
00457 :
00458 : asbm43 := { "CúActualiza Parametros",;
00459 :          "Actualiza los Parametros del Sistema",;
00460 :          {||  NIL  }, .F.;
00461 :          }
00462 :
00463 : asbm4  := { { asbm41 }, 64, 5 }
00464 :
00465 :
00466 : sbm    := { asbm1, asbm2, asbm3, asbm4 }
00467 :
```

Programa: MUESTRAS.CH - 1

```
00001 : *****
00002 : *
00003 : *      Archivo : MUESTRAS.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    18 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de datos
00011 : *      de las muestras de aguas (MUESTRA.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndMuestra := { ;
00016 : { "MUESTRAS", "MUESTRAF", "MUESTRAL", "MUESTRAC", NIL },;
00017 : { "PROTOCOLO", "DEPTO+TPOFUENTE+NROSERIE+PROTOCOLO",;
00018 : "DEPTO+CODLOC+CODSITIO", "SOLICITA+CODLOC+CODSITIO", NIL } }
00019 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : POZOCRON.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    12 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Definicion de indices para la base de crono-
00011 : *      metraje de perforacion (POZOCRON.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndCron := { { "POZOCRON", NIL },;
00016 :                   { "DEPTO+TPOFUENTE+NROSERIE+STR(PROFUN,3)", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : POZODIPE.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    11 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Definición de índices para la base de datos
00011 : *                  de diseño de perforaciones (POZODIPE.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndDiPe := { { "POZODIPE", NIL },;
00016 :                    { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```

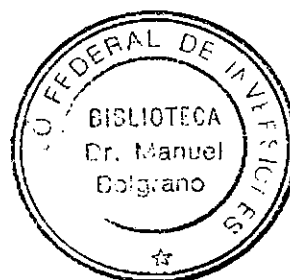
```
00001 : *****
00002 : *
00003 : *      Archivo : POZODIPF.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 11 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de datos
00011 : *      de filtros de pozos (POZODIPF.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndDiPF := { { "POZODIPF", NIL },;
00016 : { "DEPTO+TPOFUENTE+NROSERIE+STR(ALTINI,6,2)", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : POZOGEOF.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    12 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de datos
00011 : *      geofisicos de la perforacion (POZOGEOF.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndGeof := { { "POZOGEOF", NIL },;
00016 :                    { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo : POZOGEOH.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    12 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de des-
00011 : *      cripcion litologica de la perforacion (POZOGEOH.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndGeoH := { { "POZOGEOH", NIL },;
00016 :                    { "DEPTO+TPOFUENTE+NROSERIE+STR(PROFINI,6,2)", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : POZOGEOLOG.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    12 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Definición de índices para la base de datos
00011 : *                  litológicos de la perforación (POZOGEOLOG.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndGeol := { { "POZOGEOLOG", NIL },;
00016 :                   { "DEPTO+TPOFUENTE+NROSERIE", NIL } }
00017 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo : PROTIPO.CH *
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos *
00005 : *      Programa: menu.exe *
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA *
00007 : *      Fecha: 25 de Junio de 1.993 *
00008 : *
00009 : *      Descripción: *
00010 : *      Definicion de indices para la base de iden- *
00011 : *      tificaciøn de protocolos tipo (PROTIPO.DBF). *
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndProTipo := { { "PROTIPO", NIL },;
00016 : *      { "PROTIPO", NIL } }
00017 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : PROTIPOD.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 26 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de datos
00011 : *      de Protocolos Tipo (PROTIPOD.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndProTDet := { { "PROTIPOD", NIL },;
00016 : *      { "PROTIPO+DETERM", NIL } }
00017 :
```

Programa: RESCAR.CH - 1

```
00001 : *****
00002 : *
00003 : *      Archivo : RESCAR.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor: Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha: 12 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de resul-
00011 : *      tados de caracterizacion quimica de aguas (RESCAR.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndResCar := { { "RESCAR", NIL },;
00016 : { "PROTOCOLO+METCAR", NIL } }
00017 :
```

Programa: RESDETER.CH - 1

```
00001 : *****
00002 : *
00003 : *      Archivo : RESDETER.CH
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: menu.exe
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    29 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Definicion de indices para la base de resul-
00011 : *      tados de analisis (RESDETER.DBF).
00012 : *
00013 : *****
00014 :
00015 : LOCAL aIndResDet := { { "RESDETER", NIL },;
00016 : *      { "PROTOCOLO+DETERM", NIL } }
00017 :
```

```

00001 : *****
00002 : *
00003 : *      Archivo : AGUASUP.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: aguasup
00006 : *      Autor:   Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:   03 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo Maestro de Datos Identifi-
00011 : *                  catorios de Aguas Superficiales.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "IdFuente.Ch"
00017 : #include "AguaSup.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aIdFuente, aAguaSup
00022 : PRIVATE cmDepto
00023 : PRIVATE cmTpoFuente, cmNroSerie, cmDenomina
00024 : PRIVATE cmCuenca, cmSubcuenca, cmSubcuenca2
00025 : PRIVATE cmRegimen, cmArea, cmFormGeol
00026 :
00027 : DO Pantalla WITH "Datos de Aguas Superficiales", " AguaSup"
00028 :
00029 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00030 : aIdFuente   := OpenDbf( "IDFUENTE", aIndIdFte, 1 )
00031 : aAguaSup    := OpenDbf( "AGUASUP", aIndAgSup, 1 )
00032 :
00033 : *SET KEY -9 TO list_val // Programa Tecla de Función
00034 :
00035 : ABM( "Fuente", { || PAgSup0() }, { || PAgSup1() },;
00036 :      { || PAgSup2() }, { || PAgSup2() }, { || bAltaAgSup() },;
00037 :      { || dbDelete() }, { || bModiAgSup() }, NIL )
00038 :
00039 : SET KEY -9 TO // Programa Tecla de Función
00040 : RELEASE ALL LIKE cm*
00041 :
00042 : CloseDbf( "AguaSup" )
00043 : CloseDbf( "IDFUENTE" )
00044 : CloseDbf( "DEPTO" )
00045 :
00046 : RETURN
00047 :
00048 :

```

```
00049 : FUNCTION bAltaAgSup
00050 :
00051 : REPLACE DEPTO      WITH cmDepto,;
00052 :      TPOFUENTE     WITH cmTpoFuente,;
00053 :      NROSERIE       WITH cmNroSerie
00054 :
00055 : RETURN
00056 :
00057 :
00058 : Func bModiAgSup()
00059 :
00060 : REPLACE CUENCA      WITH cmCuenca,;
00061 :      SUBCUENCA      WITH cmSubCuenca,;
00062 :      SUBCUENCA2     WITH cmSub2Cuenca,;
00063 :      REGIMEN        WITH cmRegimen,;
00064 :      AREA           WITH cmArea,;
00065 :      FORMGEOL       WITH cmFormGeol
00066 :
00067 : Return
00068 :
00069 :
00070 : FUNCTION PAgSup0()
00071 :
00072 : @ 4, 1 SAY "Codigo Departamento:"
00073 : @ 5, 1 SAY "Tipo de Fuente.....:"
00074 : @ 6, 1 SAY "Número de la Fuente:"
00075 : @ 7, 1 SAY "Codigo Cuenca Prim.:"
00076 : @ 8, 1 SAY "Codigo Cuenca Sec.:"
00077 : @ 9, 1 SAY "Codigo Cuenca Terc.:"
00078 : @ 10, 1 SAY "Tipo de Regimen.....:"
00079 : @ 11, 1 SAY "Form.Geológicas:....:"
00080 :
00081 : RETURN
00082 :
00083 :
00084 : FUNCTION PAgSup1()
00085 :
00086 : PRIVATE KNUMERO1, KNUMERO2, KNUMERO3
00087 :
00088 : KNUMERO1 := " "
00089 : KNUMERO2 := " "
00090 : KNUMERO3 := 0
00091 : iniMAgSup()
00092 :
00093 : @ 5, 23 CLEAR TO 19, 79
00094 :
00095 : * 4, 22 SAY CHR(18)
00096 : @ 4, 23 GET KNUMERO1 PICTURE "99";
```



```
00097 :          VALID valexist( "DEPTO", KNUMERO1,;
00098 :          "Departamento Inexistente", { || showDepto( 4 ) } )
00099 : *   5, 22 SAY CHR(18)
00100 : @   5, 23 GET KNUMERO2 PICTURE "XX" && VALID valxxxxxx( @KNUMERO )
00101 : *   6, 22 SAY CHR(18)
00102 : @   6, 23 GET KNUMERO3 PICTURE "99999" ;
00103 :          VALID valexist( "IDFUENTE", KNUMERO1+KNUMERO2+STR(KNUMERO3,5)+;
00104 :          ".00", "Fuente Inexistente", { || showIdFte( 6 ) } )
00105 : READ
00106 : CLEAR GETS
00107 :
00108 : cmDepto      := RIGHT( "00"+ALLTRIM(KNUMERO1),2)
00109 : cmTpoFuente := KNUMERO2
00110 : IF LASTKEY() != 27
00111 :     cmNroSerie := STR( KNUMERO3, 5 )
00112 : ENDIF
00113 :
00114 : RETURN( cmDepto+cmTpoFuente+cmNroSerie )
00115 :
00116 :
00117 : FUNCTION PAgSup2()
00118 :
00119 : IF FOUND()
00120 :     loadAgSup()
00121 : ENDIF
00122 :
00123 : @   7, 23 GET  cmCuenca      PICTURE "99999"
00124 : @   8, 23 GET  cmSubCuenca   PICTURE "99999"
00125 : @   9, 23 GET  cmSub2Cuenca  PICTURE "99999"
00126 : @  10, 23 GET  cmRegimen
00127 : @  11, 23 GET  cmArea        PICTURE "99999.99"
00128 : @  12, 23 GET  cmFormgeol
00129 :
00130 : RETURN
00131 :
00132 :
00133 : FUNCTION iniMAgSup()
00134 :
00135 : cmDepto      := EVAL( bBlankField, aAguaSup, "DEPTO" )
00136 : cmTpoFuente  := EVAL( bBlankField, aAguaSup, "TPOFUENTE" )
00137 : cmNroSerie   := EVAL( bBlankField, aAguaSup, "NROSERIE" )
00138 : cmCuenca     := EVAL( bBlankField, aAguaSup, "CUENCA" )
00139 : cmSubCuenca  := EVAL( bBlankField, aAguaSup, "SUBCUENCA" )
00140 : cmSub2Cuenca := EVAL( bBlankField, aAguaSup, "SUBCUENCA2" )
00141 : cmRegimen    := 0
00142 : cmArea       := 0
00143 : cmFormGeo1   := EVAL( bBlankField, aAguaSup, "FORMGEOL" )
00144 :
```

```
00145 : Return
00146 :
00147 :
00148 : Func loadAgSup()
00149 :
00150 : cmCuenca      = CUENCA
00151 : cmSubCuenca  = SUBCUENCA
00152 : cmSub2Cuenca = SUBCUENCA2
00153 : cmRegimen    = REGIMEN
00154 : cmArea        = AREA
00155 : cmFormGeol   = FORMGEOL
00156 :
00157 : Return
00158 :
00159 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : AGUASUPI.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: aguasupi
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    15 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Emite Listado de Archivo de Identificación
00011 : *                  de Aguas Superficiales.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "AguaSup.Ch"
00017 :
00018 : PRIVATE aAguaSup, aDepto, cmForm
00019 : #include "aTipoFte.Ch"
00020 :
00021 : DO Pantalla WITH "Listado de Identif. de Aguas Superf.", "AguaSupI"
00022 : SETCOLOR( kClrstd )
00023 :
00024 : aDepto := OpenDbf( "DEPTO", aIndDepto, 1 )
00025 : aAguaSup := OpenDbf( "AGUASUP", aIndAgSup, 1 )
00026 : SET RELATION TO DEPTO INTO DEPTO
00027 :
00028 : SET DELETED ON
00029 : @ 4, 0 CLEAR TO 19,79
00030 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00031 :   @ 21, 0 CLEAR TO 22,79
00032 :   @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00033 :   * 22, 0 SAY "Pulse <Esc> para interrumpir"
00034 :   IF ISPRINTER()
00035 :     SET CONSOLE OFF
00036 :     SET PRINTER ON
00037 :     QQOUT( CHR(27)+CHR(64)+CHR(27)+CHR(77) )
00038 :     SET CONSOLE ON
00039 :     SET PRINTER OFF
00040 :     cmForm := kFormDir+"\AGUASUP"
00041 :     REPORT FORM &cmForm TO PRINT NOCONSOLE
00042 :   ELSE
00043 :     @ 21, 0 CLEAR TO 22,79
00044 :     ALERT( "Impresora con problemas", { "Cancela" } )
00045 :   ENDIF
00046 : ENDIF
00047 : @ 21, 0 CLEAR TO 22,79
00048 :
```

```
00049 : SET DELETED OFF
00050 : RELEASE ALL LIKE cm*
00051 :
00052 : DBCLEARRELATION()
00053 : CloseDbf( "AguaSup" )
00054 : CloseDbf( "DEPTO" )
00055 :
00056 : RETURN
00057 :
00058 :
```

```

00001 : *****
00002 : *
00003 : *      Archivo : ANALINF1.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: analinf1
00006 : *      Autor:   Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:   26 de Octubre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Informe Estadístico de Análisis.
00011 : *
00012 : *****
00013 :
00014 : #include "DeterId.Ch"
00015 : #include "Depto.Ch"
00016 : #include "Local.Ch"
00017 : #include "Clientes.Ch"
00018 : #include "Muestras.Ch"
00019 : #include "ResDeter.Ch"
00020 : #include "aTipoFte.Ch"
00021 :
00022 : #define MY_HSEP      "---"
00023 : #define MY_CSEP      " ; "
00024 :
00025 : bBlankField := { {aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00026 :
00027 : PRIVATE aDepto, aLocal, aClie, aIdFuente
00028 : PRIVATE aMuestras, aResDeter, aDeterId
00029 : PRIVATE bFiltro, aSel
00030 :
00031 : PRIVATE cmProtocolo, cmExpediente, cmTipoAnal, cmModulos, cmMonto
00032 : PRIVATE cmDepto, cmCodLoc, cmTpoFuente, cmNroSerie, cmCodSitio, cmSitio
00033 : PRIVATE cmFechaToma, cmHoraToma, cmTomador, cmSolicita, cmFechaRec
00034 : PRIVATE cmMuestrObs, cmNDeter
00035 :
00036 : PRIVATE KCLAVE1, KCLAVE2, KCLAVE3, KCLAVE4, KCLAVE5
00037 : PRIVATE KCLAVE6
00038 :
00039 : PRIVATE aSelParm
00040 : aSelParm := { ;
00041 :     { "KCLAVE1", "F", "CLIENTES", 1, ;
00042 :       { "CODIGO", "DENOMINA" },;
00043 :       { "Nro", "SOLICITANTE" }, { || NIL },;
00044 :       { || NIL }, "", 50 ;
00045 :     },;
00046 :     { "KCLAVE2", "F", "LOCAL", 2, ;
00047 :       { "CODLOC", "LOCALIDAD", "DEPTO->NOMDEPTO" },;
00048 :       { "Nº", "LOCALIDAD", "DEPARTAMENTO" }, { || NIL },;

```

```
00049 :      { || showLocal( 10 ), cmDepto := DEPTO },;
00050 :      { { "DEPTO", { || LOCAL->DEPTO } } }, 41 ;
00051 :      },;
00052 :      { "KCLAVE5", "V",;
00053 :      { { "FQ", "BA" }, { "FISICOQUIMICO", "BACTERIOLOGICO" } } ;
00054 :      },;
00055 :      { "KCLAVE6", "F", "DETERID", 1, ;
00056 :      { "DETERM", "DETERDES" },;
00057 :      { "Nº", "DENOMINACION" }, { || NIL },;
00058 :      { || NIL }, "", 50 ;
00059 :      );
00060 :      }
00061 :
00062 : DO Pantalla WITH "Estadísticas de Análisis I", "AnalInf1"
00063 : SETCOLOR( kclrstd )
00064 :
00065 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00066 : aLocal      := OpenDbf( "LOCAL", aIndLocal, 2 )
00067 : aClie       := OpenDbf( "CLIENTES", aIndClie, 1 )
00068 : aDeterId    := OpenDbf( "DETERID", aIndDeterId, 1 )
00069 : aResDeter   := OpenDbf( "RESDETER", aIndResDet, 1 )
00070 : aMuestras   := OpenDbf( "MUESTRA", aIndMuestra, 4 )
00071 :
00072 : SET KEY -9 TO list_val // Programa Tecla de Función
00073 :
00074 : DO WHILE .T.
00075 :     SELECT( "MUESTRA" )
00076 :     @ 4, 0 CLEAR TO 19,79
00077 :     @ 21, 0 CLEAR TO 22,79
00078 :     PAnallIO()
00079 :     bFiltro := PAnallI1()
00080 :     IF LASTKEY() == 27
00081 :         EXIT
00082 :     ENDIF
00083 :     PAnallI2( bFiltro )
00084 : ENDDO
00085 :
00086 : SET KEY -9 TO // Programa Tecla de Función
00087 : RELEASE ALL LIKE cm*
00088 :
00089 : CloseDbf( "MUESTRA" )
00090 : CloseDbf( "RESDETER" )
00091 : CloseDbf( "DETERID" )
00092 : CloseDbf( "CLIENTES" )
00093 : CloseDbf( "LOCAL" )
00094 : CloseDbf( "DEPTO" )
00095 :
00096 : RETURN
```

```
00097 :  
00098 :  
00099 : FUNCTION PAnallIO()  
00100 :  
00101 : @ 4, 1 SAY "Solicitante.....:"  
00102 : @ 5, 1 SAY "Código de Localidad:"  
00103 : @ 6, 1 SAY "Fecha Toma Desde....:"  
00104 : @ 7, 1 SAY "Fecha Toma Hasta....:"  
00105 : @ 8, 1 SAY "Tipo de Análisis....:"  
00106 : @ 9, 1 SAY "Parámetro.....:"  
00107 :  
00108 : RETURN  
00109 :  
00110 :  
00111 : FUNCTION PAnallII()  
00112 :  
00113 : LOCAL bFiltro  
00114 :  
00115 : KCLAVE1 := " "  
00116 : KCLAVE2 := " "  
00117 : KCLAVE3 := CTOD( " / / " )  
00118 : KCLAVE4 := CTOD( " / / " )  
00119 : KCLAVE5 := " "  
00120 : KCLAVE6 := " "  
00121 :  
00122 : @ 4, 23 CLEAR TO 19, 79  
00123 : @ 21, 0 SAY "Ingrese solo los datos conocidos"  
00124 : @ 22, 0 SAY "Pulse <Esc> para salir"  
00125 :  
00126 : @ 4, 22 SAY CHR(18)  
00127 : @ 4, 23 GET KCLAVE1 PICTURE "999" VALID IIF( KCLAVE1 == " ", .T.,;  
00128 :     valexist( "CLIENTES", (KCLAVE1:=RIGHT("000"+ALLTRIM(KCLAVE1),3)),;  
00129 :     "Solicitante Inexistente", { || showClie( 4 ) } ) )  
00130 : @ 5, 22 SAY CHR(18)  
00131 : @ 5, 23 GET KCLAVE2 PICTURE "999" VALID IIF( KCLAVE2 == " ", .T.,;  
00132 :     valexist( "LOCAL", (KCLAVE2:=RIGHT("000"+ALLTRIM(KCLAVE2),3)),;  
00133 :     "Localidad Inexistente", { || showLocal( 5 ),;  
00134 :     cmDepto := LOCAL->DEPTO } ) )  
00135 : @ 6, 23 GET KCLAVE3  
00136 : @ 7, 23 GET KCLAVE4  
00137 : @ 8, 22 SAY CHR(18)  
00138 : @ 8, 23 GET KCLAVE5 PICTURE "!!" VALID IIF( KCLAVE5 == " ", .T.,;  
00139 :     IIF( KCLAVE5 == "BA", .T., IIF( KCLAVE5 == "FQ", .T., .F. ) ) )  
00140 : @ 9, 22 SAY CHR(18)  
00141 : @ 9, 23 GET KCLAVE6 PICTURE "999" VALID IIF( KCLAVE6 == " ", .T.,;  
00142 :     valexist( "DETERID", (KCLAVE6:=RIGHT("000"+ALLTRIM(KCLAVE6),3)),;  
00143 :     "Parámetro Inexistente", { || showDetId( 9 ) } ) )  
00144 : READ
```

```
00145 : CLEAR GETS
00146 :
00147 : IF LASTKEY() != 27
00148 :     KCLAVE1 := RIGHT( "000"+ALLTRIM(KCLAVE1),3)
00149 :     KCLAVE1 := IIF( KCLAVE1 == "000", "", KCLAVE1 )
00150 :     KCLAVE2 := RIGHT( "000"+ALLTRIM(KCLAVE2),3)
00151 :     KCLAVE2 := IIF( KCLAVE2 == "000", "", KCLAVE2 )
00152 :     KCLAVE5 := ALLTRIM( KCLAVE5 )
00153 :     KCLAVE6 := RIGHT( "000"+ALLTRIM(KCLAVE6),3)
00154 :
00155 :     IF KCLAVE3 == CTOD( " / / " )
00156 :         IF KCLAVE4 == CTOD( " / / " )
00157 :             bFiltro := { || SOLICITA           = KCLAVE1 .AND. ;
00158 :                         CODLOC               = KCLAVE2 .AND. ;
00159 :                         TIPOANAL              = KCLAVE5 }
00160 :         ELSE
00161 :             bFiltro := { || SOLICITA           = KCLAVE1 .AND. ;
00162 :                         CODLOC               = KCLAVE2 .AND. ;
00163 :                         FECHATOMA            <= KCLAVE4 .AND. ;
00164 :                         TIPOANAL              = KCLAVE5 }
00165 :         ENDIF
00166 :     ELSE
00167 :         IF KCLAVE4 == CTOD( " / / " )
00168 :             bFiltro := { || SOLICITA           = KCLAVE1 .AND. ;
00169 :                         CODLOC               = KCLAVE2 .AND. ;
00170 :                         FECHATOMA            >= KCLAVE3 .AND. ;
00171 :                         TIPOANAL              = KCLAVE5 }
00172 :         ELSE
00173 :             bFiltro := { || SOLICITA           = KCLAVE1 .AND. ;
00174 :                         CODLOC               = KCLAVE2 .AND. ;
00175 :                         FECHATOMA            >= KCLAVE3 .AND. ;
00176 :                         FECHATOMA            <= KCLAVE4 .AND. ;
00177 :                         TIPOANAL              = KCLAVE5 }
00178 :         ENDIF
00179 :     ENDIF
00180 : ENDIF
00181 :
00182 : RETURN( bFiltro )
00183 :
00184 :
00185 : FUNCTION PAnallI2( bFiltro )
00186 :
00187 : LOCAL nReg
00188 : PRIVATE lImp, nPag, lSec1, lSec2
00189 : PRIVATE nTota, nTotb, nTotc
00190 : PRIVATE nSec1a, nSec1b, nSec1c
00191 : PRIVATE nSec2a, nSec2b, nSec2c
00192 :
```



```
00193 : nReg := 0
00194 : lImp := .T.
00195 : nPag := 0
00196 : lSec1 := .F.
00197 : lSec2 := .F.
00198 : nTota := nTotb := nTotc := 0
00199 : nSec1a := nSec1b := nSec1c := 0
00200 : nSec2a := nSec2b := nSec2c := 0
00201 :
00202 : SET ORDER TO
00203 : GO TOP
00204 : IF KCLAVE6 != "000"
00205 :     DBEVAL( { || nReg := nReg + FAnallI(KCLAVE6) }, bFiltro )
00206 : ELSE
00207 :     DBEVAL( { || ++nReg }, bFiltro )
00208 : ENDIF
00209 : SET ORDER TO 4
00210 :
00211 : IF nReg == 0
00212 :     ALERT( "No se hallaron registros" )
00213 : ELSE
00214 :     @ 4, 0 CLEAR TO 19,79
00215 :     @ 21, 0 CLEAR TO 22,79
00216 :     @ 4, 1 SAY "Registros hallados : "+STR( nReg, 5 )
00217 :     * 22, 0 SAY "Pulse <Esc> para cancelar"
00218 :
00219 :     IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00220 :         @ 21, 0 CLEAR TO 22,79
00221 :         @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00222 :         * 22, 0 SAY "Pulse <Esc> para interrumpir"
00223 :         lImp := ISPRINTER()
00224 :         cmCodLoc := "000"
00225 :         cmSolicita := "000"
00226 :         GO TOP
00227 :         IF lImp
00228 :             SET CONSOLE OFF
00229 :             SET PRINT ON
00230 :             ?? CHR(27)+CHR(64)+CHR(27)+CHR(77)
00231 :             SET PRINT OFF
00232 :             SET CONSOLE ON
00233 :             DBEVAL( { || IAnallI() }, bFiltro )
00234 :             SET DEVICE TO PRINTER
00235 :             IF lSec2
00236 :                 IF PROW() > 66
00237 :                     ++nPag
00238 :                     HAnallI( nPag )
00239 :             ENDIF
00240 :             @ PROW()+1, 10 SAY "*** Subsubtotal:"
```

```
00241 :      @ PROW(), 26 SAY "Muestras:"
00242 :      @ PROW(), 36 SAY nSec2a PICTURE "999999"
00243 :      @ PROW(), 44 SAY "Modulos:"
00244 :      @ PROW(), 53 SAY nSec2b PICTURE "999999"
00245 :      @ PROW(), 61 SAY "Determinaciones:"
00246 :      @ PROW(), 78 SAY nSec2c PICTURE "999999"
00247 :      nSec1a += nSec2a
00248 :      nSec1b += nSec2b
00249 :      nSec1c += nSec2c
00250 :      nSec2a := 0
00251 :      nSec2b := 0
00252 :      nSec2c := 0
00253 :      ENDIF
00254 :
00255 :      IF !Sec1
00256 :          IF PROW() > 66
00257 :              ++nPag
00258 :              HAnallI( nPag )
00259 :          ENDIF
00260 :          @ PROW()+1, 10 SAY "* Subtotal:"
00261 :          @ PROW(), 26 SAY "Muestras:"
00262 :          @ PROW(), 36 SAY nSec1a PICTURE "999999"
00263 :          @ PROW(), 44 SAY "Modulos:"
00264 :          @ PROW(), 53 SAY nSec1b PICTURE "999999"
00265 :          @ PROW(), 61 SAY "Determinaciones:"
00266 :          @ PROW(), 78 SAY nSec1c PICTURE "999999"
00267 :          nTota += nSec1a
00268 :          nTotb += nSec1b
00269 :          nTotc += nSec1c
00270 :          nSec1a := 0
00271 :          nSec1b := 0
00272 :          nSec1c := 0
00273 :      ENDIF
00274 :
00275 :      @ PROW()+1, 10 SAY "Total:"
00276 :      @ PROW(), 26 SAY "Muestras:"
00277 :      @ PROW(), 36 SAY nTota PICTURE "999999"
00278 :      @ PROW(), 44 SAY "Modulos:"
00279 :      @ PROW(), 53 SAY nTotb PICTURE "999999"
00280 :      @ PROW(), 61 SAY "Determinaciones:"
00281 :      @ PROW(), 78 SAY nTotc PICTURE "999999"
00282 :
00283 :      EJECT
00284 :      SET DEVICE TO SCREEN
00285 :      ENDIF
00286 :  ENDIF
00287 :  @ 21, 0 CLEAR TO 22,79
00288 : ENDIF
```

```
00289 :
00290 : RETURN
00291 :
00292 :
00293 : FUNCTION FAnallI( cmDeterm )
00294 :
00295 : Local cmProtocolo, lFlag
00296 :
00297 : cmProtocolo := PROTOCOLO
00298 : SELECT( "RESDETER" )
00299 : GO TOP
00300 : SEEK( cmProtocolo+cmDeterm )
00301 : lFlag := FOUND() .AND. !DELETED()
00302 : SELECT( "MUESTRA" )
00303 :
00304 :
00305 : RETURN( IIF( lFlag, 1, 0 ) )
00306 :
00307 :
00308 : FUNCTION IAnallI()
00309 :
00310 : LOCAL n, lFlag
00311 :
00312 : cmProtocolo := PROTOCOLO
00313 :
00314 : IF KCLAVE6 != "000"
00315 :   lFlag := ( FAnallI(KCLAVE6) == 1 )
00316 : ELSE
00317 :   lFlag := .T.
00318 : ENDIF
00319 :
00320 : SET DEVICE TO PRINTER
00321 :
00322 : IF lImp .AND. lFlag
00323 :   IF PROW() < 17 .OR. PROW() > 66
00324 :     ++nPag
00325 :     HAnallI( nPag )
00326 :   ENDIF
00327 :   LAnallI()
00328 : ENDIF
00329 :
00330 : SET DEVICE TO SCREEN
00331 :
00332 : RETURN
00333 :
00334 :
00335 : FUNCTION HAnallI( nPag )
00336 :
```

```
00337 : @ 1, 10 SAY "PROVINCIA DEL NEUQUEN"
00338 : @ 1, 82 SAY "Página"
00339 : @ 1, 89 SAY nPag PICTURE "999"
00340 : @ 2, 10 SAY "MINISTERIO DE LA PRODUCCION"
00341 : @ 2, 89 SAY DATE()
00342 : @ 3, 10 SAY "ENTE PROVINCIAL DE AGUA Y SANEAMIENTO"
00343 : @ 5, 33 SAY "LABORATORIO DE AGUAS"
00344 : @ 7, 28 SAY "INFORME ESTADISTICO DE ANALISIS"
00345 : @ 9, 10 SAY "CONDICIONES DE BUSQUEDA:"
00346 : @ 10, 10 SAY "Solicitante.....:"
00347 : IF LEN(KCLAVE1) > 0
00348 :     SELECT( "CLIENTES" )
00349 :     SEEK( KCLAVE1 )
00350 :     @ 10, 33 SAY "("+KCLAVE1+" ) "+DENOMINA
00351 :     SELECT( "MUESTRA" )
00352 : ELSE
00353 :     @ 10, 33 SAY "Todos"
00354 : ENDIF
00355 : @ 11, 10 SAY "Localidad.....:"
00356 : IF LEN(KCLAVE2) > 0
00357 :     SELECT( "LOCAL" )
00358 :     SEEK( KCLAVE2 )
00359 :     @ 11, 33 SAY "("+KCLAVE2+" ) "+LOCALIDAD
00360 :     SELECT( "MUESTRA" )
00361 : ELSE
00362 :     @ 11, 33 SAY "Todas"
00363 : ENDIF
00364 : @ 12, 10 SAY "Fecha Toma Desde...:"
00365 : IF KCLAVE3 == CTOD( " / / " )
00366 :     @ 12, 33 SAY "Primera"
00367 : ELSE
00368 :     @ 12, 33 SAY KCLAVE3
00369 : ENDIF
00370 : @ 13, 10 SAY "Fecha Toma Hasta...:"
00371 : IF KCLAVE4 == CTOD( " / / " )
00372 :     @ 13, 33 SAY "Ultima"
00373 : ELSE
00374 :     @ 13, 33 SAY KCLAVE4
00375 : ENDIF
00376 : @ 14, 10 SAY "Tipo de Análisis...:"
00377 : @ 14, 33 SAY IIF( KCLAVE5 == "BA", "BACTERIOLOGICO",;
00378 :     IIF( KCLAVE5 == "FQ", "FISICOQUIMICO", "Todos" ) )
00379 : @ 15, 10 SAY "Parámetro.....:"
00380 : IF KCLAVE6 != "000"
00381 :     SELECT( "DETERID" )
00382 :     SEEK( KCLAVE6 )
00383 :     @ 15, 33 SAY "("+KCLAVE6+" ) "+DETERDES
00384 :     SELECT( "MUESTRA" )
```

```
00385 : ELSE
00386 :   @ 15, 33 SAY "No especificado"
00387 : ENDIF
00388 :
00389 : @ 17, 10 SAY "-----"
00390 : @ 17, 20 SAY "-----"
00391 : @ 17, 29 SAY "-----"
00392 : @ 17, 50 SAY "-----"
00393 : @ 17, 62 SAY "-----"
00394 :
00395 : @ 18, 10 SAY "PROTOCOLO"
00396 : @ 18, 20 SAY " FECHA"
00397 : @ 18, 29 SAY " LUGAR DE EXTRACCION"
00398 : @ 18, 50 SAY "CALIDAD"
00399 : @ 18, 62 SAY "RESULTADO"
00400 :
00401 : @ 19, 10 SAY "-----"
00402 : @ 19, 20 SAY "-----"
00403 : @ 19, 29 SAY "-----"
00404 : @ 19, 50 SAY "-----"
00405 : @ 19, 62 SAY "-----"
00406 :
00407 : RETURN
00408 :
00409 :
00410 : FUNCTION LAnallI()
00411 :
00412 : LOCAL cCol3, nCol3, n
00413 :
00414 : IF ( cmCodLoc != CODLOC .OR. cmSolicita != SOLICITA ) .AND. !Sec2
00415 :   IF PROW() > 66
00416 :     ++nPag
00417 :     HAnallI( nPag )
00418 :   ENDIF
00419 :   @ PROW()+1, 10 SAY "** Subsubtotal:"
00420 :   @ PROW(), 26 SAY "Muestras:"
00421 :   @ PROW(), 36 SAY nSec2a PICTURE "999999"
00422 :   @ PROW(), 44 SAY "Modulos:"
00423 :   @ PROW(), 53 SAY nSec2b PICTURE "999999"
00424 :   @ PROW(), 61 SAY "Determinaciones:"
00425 :   @ PROW(), 78 SAY nSec2c PICTURE "999999"
00426 :   nSec1a += nSec2a
00427 :   nSec1b += nSec2b
00428 :   nSec1c += nSec2c
00429 :   nSec2a := 0
00430 :   nSec2b := 0
00431 :   nSec2c := 0
00432 : ENDIF
```

```
00433 :
00434 : IF cmSolicita != SOLICITA .AND. !Sec1
00435 :     IF PROW() > 66
00436 :         ++nPag
00437 :         HAnallI( nPag )
00438 :     ENDIF
00439 :     @ PROW()+1, 10 SAY "* Subtotal:"
00440 :     @ PROW(), 26 SAY "Muestras:"
00441 :     @ PROW(), 36 SAY nSec1a PICTURE "999999"
00442 :     @ PROW(), 44 SAY "Modulos:"
00443 :     @ PROW(), 53 SAY nSec1b PICTURE "999999"
00444 :     @ PROW(), 61 SAY "Determinaciones:"
00445 :     @ PROW(), 78 SAY nSec1c PICTURE "999999"
00446 :     nTota += nSec1a
00447 :     nTotb += nSec1b
00448 :     nTotc += nSec1c
00449 :     nSec1a := 0
00450 :     nSec1b := 0
00451 :     nSec1c := 0
00452 : ENDIF
00453 :
00454 : IF cmSolicita != SOLICITA .OR. !!Sec1
00455 :     IF PROW() > 66
00456 :         ++nPag
00457 :         HAnallI( nPag )
00458 :     ENDIF
00459 :     cmSolicita := SOLICITA
00460 :     SELECT( "CLIENTES" )
00461 :     SEEK( cmSolicita )
00462 :     @ PROW()+2, 10 SAY "** SOLICITANTE: (" +cmSolicita+" ) "+DENOMINA
00463 :     SELECT( "MUESTRA" )
00464 :     !Sec1 := .T.
00465 :     !Sec2 := .F.
00466 : ENDIF
00467 :
00468 : IF cmCodloc != CODLOC .OR. !!Sec2
00469 :     IF PROW() > 66
00470 :         ++nPag
00471 :         HAnallI( nPag )
00472 :     ENDIF
00473 :     cmCodLoc := CODLOC
00474 :     SELECT( "LOCAL" )
00475 :     SEEK( cmCodLoc )
00476 :     @ PROW()+2, 10 SAY "*** LOCALIDAD: (" +cmCodLoc+" ) "+LOCALIDAD
00477 :     SELECT( "MUESTRA" )
00478 :     !Sec2 := .T.
00479 : ENDIF
00480 :
```

Programa: ANALINF1.PRG - 11

```
00481 : ++nSec2a
00482 : nSec2b += MODULOS
00483 : nSec2c += NDETER
00484 :
00485 : @ PROW()+1, 10 SAY PROTOCOLO
00486 : @ PROW(), 20 SAY FECHAREC
00487 :
00488 : cCo13 := TRIM( SITIO )
00489 : nCo13 := MLCOUNT( cCo13, 20 )
00490 :
00491 : IF nCo13 > 1
00492 :   @ PROW(), 29 SAY MEMOLINE( cCo13, 20, 1 )
00493 : ELSE
00494 :   @ PROW(), 29 SAY cCo13
00495 : ENDIF
00496 : IF TIPOANAL == "BA"
00497 :   @ PROW(), 50 SAY LEFT( NOTAS, AT( " ", NOTAS ) );
00498 :   PICTURE "XXXXXXXXXXXX"
00499 : ENDIF
00500 : IF KCLAVE6 != "000"
00501 :   @ PROW(), 62 SAY RESDETER->VALORSIG
00502 :   @ PROW(), 65 SAY RESDETER->VALOR
00503 :   @ PROW(), 76 SAY RESDETER->UNIPRI
00504 : ENDIF
00505 :
00506 : FOR n := 2 TO nCo13
00507 :   @ PROW()+1, 29 SAY MEMOLINE( cCo13, 20, n )
00508 :   IF PROW() > 66
00509 :     ++nPag
00510 :     HAnallI( nPag )
00511 :   ENDIF
00512 : NEXT
00513 :
00514 : RETURN
00515 :
00516 :
```

```

00001 : *****
00002 : *
00003 : *      Archivo : ANALINF2.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: analinf2
00006 : *      Autor:   Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:   30 de Octubre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Informe Estadístico de Análisis 2.
00011 : *
00012 : *****
00013 :
00014 : #include "DeterId.Ch"
00015 : #include "Depto.Ch"
00016 : #include "Local.Ch"
00017 : #include "Clientes.Ch"
00018 : #include "Muestras.Ch"
00019 : #include "ResDeter.Ch"
00020 : #include "aTipoFte.Ch"
00021 :
00022 : #define MY_HSEP      "___"
00023 : #define MY_CSEP      " | "
00024 :
00025 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00026 :
00027 : PRIVATE aDepto, aLocal, aClie, aIdFuente
00028 : PRIVATE aMuestras, aResDeter, aDeterId
00029 : PRIVATE bFiltro, cHead
00030 :
00031 : PRIVATE cmProtocolo, cmExpediente, cmTipoAnal, cmModulos, cmMonto
00032 : PRIVATE cmDepto, cmCodLoc, cmTpoFuente, cmNroSerie, cmCodSitio, cmSitio
00033 : PRIVATE cmFechaToma, cmHoraToma, cmTomador, cmSolicita, cmFechaRec
00034 : PRIVATE cmMuestrObs, cmNDeter
00035 :
00036 : PRIVATE KCLAVE1, KCLAVE2, KCLAVE3
00037 :
00038 : PRIVATE aSelParm
00039 : aSelParm := { ;
00040 :     { "KCLAVE1", "F", "CLIENTES", 1, ;
00041 :       { "CODIGO", "DENOMINA" },;
00042 :       { "Nro", "SOLICITANTE" }, { || NIL },;
00043 :       { || NIL }, "", 50 ;
00044 :     };
00045 :   }
00046 :
00047 : DO Pantalla WITH "Estadísticas de Análisis 2", "AnalInf2"
00048 : SETCOLOR( kclrstd )

```



```
00049 :
00050 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00051 : aLocal      := OpenDbf( "LOCAL", aIndLocal, 2 )
00052 : aClie       := OpenDbf( "CLIENTES", aIndClie, 1 )
00053 : aDeterId    := OpenDbf( "DETERID", aIndDeterId, 1 )
00054 : aResDeter   := OpenDbf( "RESDETER", aIndResDet, 1 )
00055 : aMuestras   := OpenDbf( "MUESTRA", aIndMuestra, 4 )
00056 :
00057 : SET KEY -9 TO list_val // Programa Tecla de Función
00058 :
00059 : DO WHILE .T.
00060 :     SELECT( "MUESTRA" )
00061 :     @ 4, 0 CLEAR TO 19,79
00062 :     @ 21, 0 CLEAR TO 22,79
00063 :     PAnal2IO()
00064 :     bFiltro := PAnal2I1()
00065 :     IF LASTKEY() == 27
00066 :         EXIT
00067 :     ENDIF
00068 :     PAnal2I2( bFiltro )
00069 : ENDDO
00070 :
00071 : SET KEY -9 TO // Programa Tecla de Función
00072 : RELEASE ALL LIKE cm*
00073 :
00074 : CloseDbf( "MUESTRA" )
00075 : CloseDbf( "RESDETER" )
00076 : CloseDbf( "DETERID" )
00077 : CloseDbf( "CLIENTES" )
00078 : CloseDbf( "LOCAL" )
00079 : CloseDbf( "DEPTO" )
00080 :
00081 : RETURN
00082 :
00083 :
00084 : FUNCTION PAnal2IO()
00085 :
00086 : @ 4, 1 SAY "Solicitante.....:"
00087 : @ 5, 1 SAY "Fecha Toma Desde....:"
00088 : @ 6, 1 SAY "Fecha Toma Hasta....:"
00089 :
00090 : RETURN
00091 :
00092 :
00093 : FUNCTION PAnal2I1()
00094 :
00095 : LOCAL bFiltro
00096 :
```

```
00097 : KCLAVE1 := " "
00098 : KCLAVE2 := CTOD( " / / " )
00099 : KCLAVE3 := CTOD( " / / " )
00100 :
00101 : @ 4, 23 CLEAR TO 19, 79
00102 : @ 21, 0 SAY "Ingrese solo los datos conocidos"
00103 : @ 22, 0 SAY "Pulse <Esc> para salir"
00104 :
00105 : @ 4, 22 SAY CHR(18)
00106 : @ 4, 23 GET KCLAVE1 PICTURE "999" VALID IIF( KCLAVE1 == " ", .T.,;
00107 :     vaExist( "CLIENTES", (KCLAVE1:=RIGHT("000"+ALLTRIM(KCLAVE1),3)),;
00108 :     "Solicitante Inexistente", { || showClie( 4 ) } ) )
00109 : @ 5, 23 GET KCLAVE2
00110 : @ 6, 23 GET KCLAVE3
00111 : READ
00112 : CLEAR GETS
00113 :
00114 : IF LASTKEY() != 27
00115 :     KCLAVE1 := RIGHT( "000"+ALLTRIM(KCLAVE1),3)
00116 :     KCLAVE1 := IIF( KCLAVE1 == "000", "", KCLAVE1 )
00117 :
00118 :     IF KCLAVE2 == CTOD( " / / " )
00119 :         IF KCLAVE3 == CTOD( " / / " )
00120 :             cHead := "- "
00121 :             bFiltro := { || SOLICITA = KCLAVE1 }
00122 :         ELSE
00123 :             cHead := "Hasta el "+DTC( KCLAVE3 )
00124 :             bFiltro := { || SOLICITA = KCLAVE1 .AND. ;
00125 :                 FECHATOMA <= KCLAVE3 }
00126 :         ENDIF
00127 :     ELSE
00128 :         IF KCLAVE3 == CTOD( " / / " )
00129 :             cHead := "Desde el "+DTC( KCLAVE2 )
00130 :             bFiltro := { || SOLICITA = KCLAVE1 .AND. ;
00131 :                 FECHATOMA >= KCLAVE2 }
00132 :         ELSE
00133 :             cHead := "Desde el "+DTC(KCLAVE2)+" hasta el "+DTC(KCLAVE3)
00134 :             bFiltro := { || SOLICITA = KCLAVE1 .AND. ;
00135 :                 FECHATOMA >= KCLAVE2 .AND. ;
00136 :                 FECHATOMA <= KCLAVE3 }
00137 :         ENDIF
00138 :     ENDIF
00139 : ENDIF
00140 :
00141 : RETURN( bFiltro )
00142 :
00143 :
00144 : FUNCTION PAnal2I2( bFiltro )
```

```
00145 :
00146 : LOCAL nReg := 0
00147 : PRIVATE cForm, lImp := .T.
00148 :
00149 : GO TOP
00150 : SET DELETED ON
00151 : SET ORDER TO
00152 : DBEVAL( { || ++nReg }, bFiltro )
00153 : SET ORDER TO 4
00154 : SET DELETED OFF
00155 :
00156 : IF nReg == 0
00157 :     ALERT( "No se hallaron registros" )
00158 : ELSE
00159 :     @ 4, 0 CLEAR TO 19,79
00160 :     @ 21, 0 CLEAR TO 22,79
00161 :     @ 4, 1 SAY "Registros hallados : "+STR( nReg, 5 )
00162 :     * 21, 0 SAY "Seleccione los datos deseados"
00163 :     * 22, 0 SAY "Pulse <Esc> para cancelar"
00164 :
00165 :     IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00166 :         @ 21, 0 CLEAR TO 22,79
00167 :         @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00168 :         * 22, 0 SAY "Pulse <Esc> para interrumpir"
00169 :         IF ISPRINTER()
00170 :             SET CONSOLE OFF
00171 :             SET PRINTER ON
00172 :             QQOUT( CHR(27)+CHR(64) )
00173 :             SET CONSOLE ON
00174 :             SET PRINTER OFF
00175 :             GO TOP
00176 :             SET DELETED ON
00177 :             DBSETFILTER( bFiltro )
00178 :             DBSETRELATION( "CLIENTES", { || MUESTRA->SOLICITA } )
00179 :             DBSETRELATION( "LOCAL", { || MUESTRA->CODLOC } )
00180 :             cmForm := kFormDir+"\ANALINF2"
00181 :             REPORT FORM &cmForm TO PRINT NOCONSOLE HEADING (cHead)
00182 :             * REPORT FORM &cmForm TO PRINT NOCONSOLE
00183 :             DBCLEARRELATION()
00184 :             DBCLEARFILTER()
00185 :             SET DELETED OFF
00186 :         ELSE
00187 :             @ 21, 0 CLEAR TO 22,79
00188 :             ALERT( "Impresora con problemas", { "Cancela" } )
00189 :         ENDIF
00190 :     ENDIF
00191 :     @ 21, 0 CLEAR TO 22,79
00192 : ENDIF
```

00193 :  
00194 : RETURN  
00195 :  
00196 :

```
00001 : *****
00002 : *
00003 : *      Archivo : ANALRES.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: analres
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    29 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Ingresa Resultados de Análisis
00011 : *
00012 : *****
00013 :
00014 : #include "Muestras.Ch"
00015 : #include "ResDeter.Ch"
00016 :
00017 : #define MY_HSEP      "----"
00018 : #define MY_CSEP      " | "
00019 :
00020 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00021 :
00022 : PRIVATE aMuestras, aResDeter
00023 : PRIVATE cmProtocolo, cmExpediente, cmTipoAnal, cmModulos, cmMonto
00024 : PRIVATE cmDepto, cmCodLoc, cmTpoFuente, cmNroSerie, cmSitio
00025 : PRIVATE cmFechaToma, cmHoraToma, cmTomador, cmSolicita, cmFechaRec
00026 : PRIVATE cmMuestrObs
00027 :
00028 : DO Pantalla WITH "Identificación de las Muestras", "Muestras"
00029 :
00030 : aResDeter := OpenDbf( "RESETER", aIndResDet, 1 )
00031 : aMuestras := OpenDbf( "MUESTRA", aIndMuestra, 1 )
00032 :
00033 : *SET KEY -9 TO list_val // Programa Tecla de Función
00034 :
00035 : ABM( "Muestra", { || PMuest0() }, { || PMuest1() },;
00036 :      { || PMuest2() }, { || PMuest2() }, { || bAltaMuest() },;
00037 :      { || bBajaMuest() }, { || bModiMuest() }, { || MuestraD() } )
00038 :
00039 : SET KEY -9 TO // Programa Tecla de Función
00040 : RELEASE ALL LIKE cm*
00041 :
00042 : CloseDbf( "MUESTRA" )
00043 : CloseDbf( "RESETER" )
00044 :
00045 : RETURN
00046 :
00047 :
00048 : FUNCTION bAltaMuest
```

```
00049 :  
00050 : REPLACE PROTOCOLO WITH cmProtocolo  
00051 :  
00052 : RETURN  
00053 :  
00054 :  
00055 : FUNCTION bBajaMuest  
00056 :  
00057 : LOCAL nOldArea  
00058 :  
00059 : nOldArea := SELECT( "RESDETER" )  
00060 : GO TOP  
00061 : SEEK( cmProtocolo )  
00062 : DELETE WHILE PROTOCOLO == cmProtocolo  
00063 :  
00064 : SELECT( ALIAS( nOldArea ) )  
00065 : dbDelete()  
00066 :  
00067 : RETURN  
00068 :  
00069 :  
00070 : Func bModiMuest()  
00071 :  
00072 : REPLACE EXPEDIENTE WITH cmExpediente,;  
00073 :      TIPOANAL      WITH cmTipoAnal,;  
00074 :      MODULOS       WITH cmModulos,;  
00075 :      MONTO         WITH cmMonto,;  
00076 :      DEPTO        WITH cmDepto,;  
00077 :      CODLOC       WITH cmCodLoc,;  
00078 :      TPOFUENTE    WITH cmTpoFuente,;  
00079 :      NROSERIE     WITH cmNroSerie,;  
00080 :      SITIO        WITH cmSitio,;  
00081 :      FECHATOMA    WITH cmFechaToma,;  
00082 :      HORATOMA     WITH cmHoraToma,;  
00083 :      TOMADOR      WITH cmTomador,;  
00084 :      SOLICITA     WITH cmSolicita,;  
00085 :      FECHAREC     WITH cmFechaRec,;  
00086 :      MUESTROBS    WITH cmMuestroObs  
00087 :  
00088 : Return  
00089 :  
00090 :  
00091 : FUNCTION PMuesto()  
00092 :  
00093 : @ 4, 1 SAY "Número de Protocolo:"  
00094 : @ 5, 1 SAY "Expediente.....:"  
00095 : @ 6, 1 SAY "Tipo de Análisis..."  
00096 : @ 7, 1 SAY "Cantidad de Modulos:"
```

```
00097 : @ 8, 1 SAY "Monto Total.....:"
00098 : @ 9, 1 SAY "Código Depto.Prov.:"
00099 : @ 10, 1 SAY "Código de Localidad:"
00100 : @ 11, 1 SAY "Tipo de Fuente.....:"
00101 : @ 12, 1 SAY "Número de la.Fuente:"
00102 : @ 13, 1 SAY "Sitio Toma Muestra.:"
00103 : @ 14, 1 SAY "Fecha Toma Muestra.:"
00104 : @ 15, 1 SAY "Hora Toma Muestra.:"
00105 : @ 16, 1 SAY "Tomador de Muestra.:"
00106 : @ 17, 1 SAY "Solicitante.....:"
00107 : @ 18, 1 SAY "Fecha Recepción.....:"
00108 : @ 19, 1 SAY "Observaciones.....:"
00109 :
00110 : RETURN
00111 :
00112 :
00113 : FUNCTION PMuest1()
00114 :
00115 : PRIVATE KNUMERO
00116 :
00117 : KNUMERO := " "
00118 : iniMMuest()
00119 :
00120 : @ 5, 23 CLEAR TO 19, 79
00121 :
00122 : * 4, 22 SAY CHR(18)
00123 : @ 4, 23 GET KNUMERO PICTURE "999999" && VALID valxxxxxx( @KNUMERO )
00124 : READ
00125 : CLEAR GETS
00126 :
00127 : *cmCodCli := IIF( KCLIENT == " ", cmCodCli, KCLIENT )
00128 : cmProtocolo := RIGHT( "000000"+ALLTRIM(KNUMERO),6)
00129 :
00130 : RETURN( cmProtocolo )
00131 :
00132 :
00133 : FUNCTION PMuest2()
00134 :
00135 : IF FOUND()
00136 :     loadMuest()
00137 : ENDIF
00138 :
00139 : @ 5, 23 GET cmExpediente PICTURE "9999-999999/9999"
00140 : @ 6, 23 GET cmTipoAnal
00141 : @ 7, 23 GET cmModulos
00142 : @ 8, 23 GET cmMonto
00143 : @ 9, 23 GET cmDepto
00144 : @ 10, 23 GET cmCodLoc
```

```
00145 : @ 11, 23 GET cmTpoFuente
00146 : @ 12, 23 GET cmNroSerie
00147 : @ 13, 23 GET cmSitio
00148 : @ 14, 23 GET cmFechaToma
00149 : @ 15, 23 GET cmHoraToma
00150 : @ 16, 23 GET cmTomador
00151 : @ 17, 23 GET cmSolicita
00152 : @ 18, 23 GET cmFechaRec
00153 : @ 19, 23 GET cmMuestrObs PICTURE "@S56"
00154 :
00155 : RETURN
00156 :
00157 :
00158 : FUNCTION iniMMuest()
00159 :
00160 : cmExpediente := EVAL( bBlankField, aMuestras, "EXPEDIENTE" )
00161 : cmTipoAnal := EVAL( bBlankField, aMuestras, "TIPOANAL" )
00162 : cmModulos := 0
00163 : cmMonto := 0
00164 : cmDepto := EVAL( bBlankField, aMuestras, "DEPTO" )
00165 : cmCodLoc := EVAL( bBlankField, aMuestras, "CODLOC" )
00166 : cmTpoFuente := EVAL( bBlankField, aMuestras, "TPOFUENTE" )
00167 : cmNroSerie := EVAL( bBlankField, aMuestras, "NROSERIE" )
00168 : cmSitio := EVAL( bBlankField, aMuestras, "SITIO" )
00169 : cmFechaToma := CTOD( " / / " )
00170 : cmHoraToma := 0
00171 : cmTomador := EVAL( bBlankField, aMuestras, "TOMADOR" )
00172 : cmSolicita := EVAL( bBlankField, aMuestras, "SOLICITA" )
00173 : cmFechaRec := CTOD( " / / " )
00174 : cmMuestrObs := EVAL( bBlankField, aMuestras, "MUESTROBS" )
00175 :
00176 : Return
00177 :
00178 :
00179 : Func loadMuest()
00180 :
00181 : cmExpediente := EXPEDIENTE
00182 : cmTipoAnal := TIPOANAL
00183 : cmModulos := MODULOS
00184 : cmMonto := MONTO
00185 : cmDepto := DEPTO
00186 : cmCodLoc := CODLOC
00187 : cmTpoFuente := TPOFUENTE
00188 : cmNroSerie := NROSERIE
00189 : cmSitio := SITIO
00190 : cmFechaToma := FECHATOMA
00191 : cmHoraToma := HORATOMA
00192 : cmTomador := TOMADOR
```



```
00193 : cmSolicita      := SOLICITA
00194 : cmFechaRec      := FECHAREC
00195 : cmMuestrObs     := MUESTROBS
00196 :
00197 : Return
00198 :
00199 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ANALRES.D.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: analresd
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    29 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Ingresa Resultados de las Determinaciones
00011 : *      efectuadas sobre la Muestra.
00012 : *
00013 : *****
00014 :
00015 : #include "DeterId.Ch"
00016 : *include "ProTipo.Ch"
00017 : #include "ProTipod.Ch"
00018 :
00019 : #define MY_HSEP      "----"
00020 : #define MY_CSEP      " | "
00021 :
00022 : LOCAL cMemScrn
00023 : PRIVATE aDeterId, aProTipo, aProTipod
00024 : PRIVATE cmDeterm, cmDeterDes, cmUniPri, cmUniSec, cmFactor, cmFactorExp
00025 : PRIVATE cmValor, cmValorSig
00026 : PRIVATE cmProTipo, cmPTNomb
00027 :
00028 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00029 :
00030 : aDeterId := OpenDbf( "DETERID", aIndDeterId, 1 )
00031 : * aProTipo := OpenDbf( "PROTIPO", aIndProTipo, 1 )
00032 : * aProTipod := OpenDbf( "PROTIPOD", aIndProTDet, 1 )
00033 : SELECT( "RESDETER" )
00034 :
00035 : SET DELETED ON
00036 :
00037 : *SET KEY -9 TO list_val // Programa Tecla de Función
00038 :
00039 : cMemScrn := SAVESCREEN( 6, 0, 20, 79 )
00040 :
00041 : PMuesD00()
00042 :
00043 : ABMItem( 11, cmProtocolo, { || cmProtocolo == PROTOCOLO },;
00044 :         { || PMuesD0() }, { |x,y,z| PMuesD1(x,y,z) }, { || PMuesD2() },;
00045 :         { |x,y,z| PMuesD3(x,y,z) }, { || bAltaMuesD() },;
00046 :         { || dbDelete() }, { || bModiMuesD() } )
00047 :
00048 : @ 21, 0 CLEAR TO 22,79
```

```
00049 :  
00050 : SET KEY -9 TO           // Programa Tecla de Función  
00051 : RELEASE ALL LIKE cm*  
00052 :  
00053 : SET DELETED OFF  
00054 :  
00055 : RESTSCREEN( 6, 0, 20, 79, cMemScrn )  
00056 :  
00057 : CloseDbf( "DETERID" )  
00058 : SELECT( "MUESTRA" )  
00059 :  
00060 : RETURN  
00061 :  
00062 :  
00063 : FUNCTION bAltaMuesD  
00064 :  
00065 : APPEND BLANK  
00066 :  
00067 : REPLACE PROTOCOLO WITH cmProtocolo,;  
00068 :         DETERM      WITH cmDeterm,;  
00069 :         DETERDES     WITH cmDeterDes,;  
00070 :         UNIPRI       WITH cmUniPri,;  
00071 :         UNISEC       WITH cmUniSec,;  
00072 :         FACTOR       WITH cmFactor,;  
00073 :         FACTOREXP    WITH cmFactorExp  
00074 : iniMMuesD()  
00075 :  
00076 : RETURN( RECNO() )  
00077 :  
00078 :  
00079 : FUNCTION iniMMuesD()  
00080 :  
00081 : cmValor      := 0  
00082 : cmValorSig   := EVAL( bBlankField, aResDeter, "VALORSIG" )  
00083 :  
00084 : Return  
00085 :  
00086 :  
00087 : Func loadMuesD()  
00088 :  
00089 : cmDeterm      := DETERM  
00090 : cmDeterDes    := DETERDES  
00091 : cmUniPri      := UNIPRI  
00092 : cmUniSec      := UNISEC  
00093 : cmFactor      := FACTOR  
00094 : cmFactorExp   := FACTOREXP  
00095 : cmValor       := VALOR  
00096 : cmValorSig    := VALORSIG
```

```
00097 :  
00098 : Return  
00099 :  
00100 :  
00101 : Func bModiMuesD()  
00102 :  
00103 : REPLACE DETERM      WITH cmDeterm,;  
00104 :      DETERDES  WITH cmDeterDes,;  
00105 :      UNIPRI    WITH cmUniPri,;  
00106 :      UNISEC    WITH cmUniSec,;  
00107 :      FACTOR    WITH cmFactor,;  
00108 :      FACTOREXP WITH cmFactorExp  
00109 :  
00110 : Return  
00111 :  
00112 :  
00113 : FUNCTION PMuesD00()  
00114 :  
00115 : @ 6, 0 CLEAR TO 19,79  
00116 :  
00117 : @ 6, 4,20,76 BOX "+-+|---|"  
00118 :  
00119 : @ 6, 4, 8, 8 BOX "+-+|+-+|"  
00120 : @ 6,39, 8,50 BOX "---|+-+|"  
00121 : @ 6,61, 8,72 BOX "---|+-+|"  
00122 :  
00123 : @ 8, 8,20,39 BOX "+-+|---|"  
00124 : @ 8,50,20,61 BOX "+-+|---|"  
00125 : @ 8,72,20,76 BOX "+-+|---|"  
00126 :  
00127 : @ 7, 5 SAY "Det"  
00128 : @ 7,18 SAY "Descripción"  
00129 : @ 7,40 SAY "Unid.Prin."  
00130 : @ 7,51 SAY "Unid.Sec."  
00131 : @ 7,62 SAY "Factor"  
00132 : @ 7,73 SAY "Exp"  
00133 :  
00134 : Return NIL  
00135 :  
00136 :  
00137 : FUNCTION PMuesD0()  
00138 :  
00139 : @ 9, 5 CLEAR TO 19, 7  
00140 : @ 9, 9 CLEAR TO 19,38  
00141 : @ 9,40 CLEAR TO 19,49  
00142 : @ 9,51 CLEAR TO 19,60  
00143 : @ 9,62 CLEAR TO 19,71  
00144 : @ 9,73 CLEAR TO 19,75
```

```
00145 :
00146 : Return NIL
00147 :
00148 :
00149 : FUNCTION PMuesD1( nReg, nPriReg, cColor )
00150 :
00151 : Local n
00152 :
00153 : n := nReg - nPriReg
00154 :
00155 : loadMuesD()
00156 :
00157 : @ 9+n, 5 SAY cmDeterm      PICTURE "999"          COLOR cColor
00158 : @ 9+n, 9 SAY cmDeterDes    COLOR cColor
00159 : @ 9+n, 40 SAY cmUniPri     COLOR cColor
00160 : @ 9+n, 51 SAY cmUniSec     COLOR cColor
00161 : @ 9+n, 62 SAY cmFactor     PICTURE "99999.9999"    COLOR cColor
00162 : @ 9+n, 73 SAY cmFactorExp  PICTURE "999"          COLOR cColor
00163 :
00164 : Return NIL
00165 :
00166 :
00167 : FUNCTION PMuesD2()
00168 :
00169 : PRIVATE cOldclre, cMemscre
00170 : PRIVATE nTop := 5, nLeft := 4, nBottom := 17, nRight := 75
00171 :
00172 : cOldclre := SETCOLOR()
00173 : cMemscre := MakeShadBox( nTop, nLeft, nBottom, nRight )
00174 :
00175 : PBute2t( nTop, nLeft )
00176 : PBute2d( nTop, nLeft )
00177 :
00178 : SETCOLOR( cOldclre )
00179 : RESTSCREEN(nTop,nLeft,nBottom,nRight,cMemscre)
00180 :
00181 : RETURN
00182 :
00183 :
00184 : FUNCTION PMuesD3( nReg, nPriReg, cColor )
00185 :
00186 : LOCAL n
00187 :
00188 : n := nReg - nPriReg
00189 :
00190 : @ 9+n, 5 GET cmDeterm      PICTURE "999"          COLOR cColor;
00191 :          VALID vaExist( "DETERID", cmDeterm,;
00192 :          "Codigo de Determinación Inexistente", { || loadDetId() } ) )
```

```
00193 : @ 9+n, 9 GET cmDeterDes          COLOR cColor
00194 : @ 9+n,40 GET cmUniPri            COLOR cColor
00195 : @ 9+n,51 GET cmUniSec            COLOR cColor
00196 : @ 9+n,62 GET cmFactor    PICTURE "99999.9999" COLOR cColor
00197 : @ 9+n,73 GET cmFactorExp PICTURE "999"      COLOR cColor
00198 :
00199 : READ
00200 : CLEAR GETS
00201 :
00202 : RETURN
00203 :
00204 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : AUXDEPTI.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: auxdeptI
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Emite Listado del Archivo Auxiliar de
00011 : *      Departamentos.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 :
00017 : PRIVATE aDepto, cmForm
00018 :
00019 : DO Pantalla WITH "Listado de Departamentos Provinciales", "AuxDeptI"
00020 : SETCOLOR( kClrstd )
00021 :
00022 : aDepto := OpenDbf( "DEPTO", aIndDepto, 1 )
00023 :
00024 : SET DELETED ON
00025 : @ 4, 0 CLEAR TO 19,79
00026 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00027 :     @ 21, 0 CLEAR TO 22,79
00028 :     @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00029 :     * 22, 0 SAY "Pulse <Esc> para interrumpir"
00030 :     IF ISPRINTER()
00031 :         SET CONSOLE OFF
00032 :         SET PRINTER ON
00033 :         QQOUT( CHR(27)+CHR(64) )
00034 :         SET CONSOLE ON
00035 :         SET PRINTER OFF
00036 :         cmForm := kFormDir+"\DEPTO"
00037 :         REPORT FORM &cmForm TO PRINT NOCONSOLE
00038 :     ELSE
00039 :         @ 21, 0 CLEAR TO 22,79
00040 :         ALERT( "Impresora con problemas", { "Cancela" } )
00041 :     ENDIF
00042 : ENDIF
00043 : @ 21, 0 CLEAR TO 22,79
00044 :
00045 : SET DELETED OFF
00046 : RELEASE ALL LIKE cm*
00047 :
00048 : CloseDbf( "DEPTO" )
```

00049 :  
00050 : RETURN  
00051 :  
00052 :



```
00001 : *****
00002 : *
00003 : *      Archivo : AUXDEPTO.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: auxdepto
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo Auxiliar de Departamentos
00011 : *
00012 : *****
00013 :
00014 : #include "Depto.Ch"
00015 :
00016 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00017 :
00018 : PRIVATE aDepto
00019 : PRIVATE cmDepto, cmNomDepto
00020 :
00021 : DO Pantalla WITH "Departamentos Provinciales", "AuxDepto"
00022 :
00023 : aDepto := OpenDbf( "DEPTO", aIndDepto, 1 )
00024 :
00025 : *SET KEY -9 TO list val // Programa Tecla de Función
00026 :
00027 : ABM( "Departamento", { || PDepto0() }, { || PDepto1() },;
00028 :     { || PDepto2() }, { || PDepto2() }, { || bAltaDepto() },;
00029 :     { || dbDelete() }, { || bModiDepto() }, NIL )
00030 :
00031 : SET KEY -9 TO // Programa Tecla de Función
00032 : RELEASE ALL LIKE cm*
00033 :
00034 : CloseDbf( "DEPTO" )
00035 :
00036 : RETURN
00037 :
00038 :
00039 : FUNCTION bAltaDepto
00040 :
00041 : REPLACE DEPTO WITH cmDepto
00042 :
00043 : RETURN
00044 :
00045 :
00046 : Func bModiDepto()
00047 :
00048 : REPLACE NOMDEPTO WITH cmNomDepto
```

```
00049 :  
00050 : Return  
00051 :  
00052 :  
00053 : FUNCTION PDepto0()  
00054 :  
00055 : @ 4, 1 SAY "Codigo Departamento:"  
00056 : @ 5, 1 SAY "Nombre Departamento:"  
00057 :  
00058 : RETURN  
00059 :  
00060 :  
00061 : FUNCTION PDepto1()  
00062 :  
00063 : PRIVATE KNUMERO  
00064 :  
00065 : KNUMERO := " "  
00066 : iniMDepto()  
00067 :  
00068 : @ 4, 23 CLEAR TO 19, 79  
00069 :  
00070 : * 4, 22 SAY CHR(18)  
00071 : @ 4, 23 GET KNUMERO PICTURE "99" && VALID valxxxxxx( @KNUMERO )  
00072 : READ  
00073 : CLEAR GETS  
00074 :  
00075 :  
00076 : *cmCodCli := IIF( KCLIENT == " ", cmCodCli, KCLIENT )  
00077 : cmDepto := RIGHT( "00"+ALLTRIM(KNUMERO),2)  
00078 :  
00079 : RETURN( cmDepto )  
00080 :  
00081 :  
00082 : FUNCTION PDepto2()  
00083 :  
00084 : IF FOUND()  
00085 :     loadDepto()  
00086 : ENDIF  
00087 :  
00088 : @ 5, 23 GET cmNomDepto  
00089 :  
00090 : RETURN  
00091 :  
00092 :  
00093 : FUNCTION iniMDepto()  
00094 :  
00095 : cmNomDepto := EVAL( bBlankField, aDepto, "NOMDEPTO" )  
00096 :
```

```
00097 : Return
00098 :
00099 :
00100 : Func loadDepto()
00101 :
00102 : cmNomDepto := NOMDEPTO
00103 :
00104 : Return
00105 :
00106 :
00107 : FUNCTION showDepto( nlin )
00108 :
00109 : @ nlin, 30 SAY DEPTO->NOMDEPTO COLOR "N/W"
00110 :
00111 : RETURN
00112 :
00113 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : AUXLOCAL.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: auxlocal
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Emite Listado del Archivo Auxiliar de
00011 : *      Localidades.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "Local.Ch"
00017 :
00018 : PRIVATE aDepto, aLocal, cmForm
00019 :
00020 : DO Pantalla WITH "Listado de Localidades", "AuxLocal"
00021 : SETCOLOR( kClrstd )
00022 :
00023 : aDepto := OpenDbf( "DEPTO", aIndDepto, 1 )
00024 : aLocal := OpenDbf( "LOCAL", aIndLocal, 1 )
00025 :
00026 : SET DELETED ON
00027 : SET RELATION TO DEPTO INTO DEPTO
00028 :
00029 : @ 4, 0 CLEAR TO 19,79
00030 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00031 :   @ 21, 0 CLEAR TO 22,79
00032 :   @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00033 :   * 22, 0 SAY "Pulse <Esc> para interrumpir"
00034 :   IF ISPRINTER()
00035 :     SET CONSOLE OFF
00036 :     SET PRINTER ON
00037 :     QQOUT( CHR(27)+CHR(64) )
00038 :     SET CONSOLE ON
00039 :     SET PRINTER OFF
00040 :     cmForm := kFormDir+"\LOCAL"
00041 :     REPORT FORM &cmForm TO PRINT NOCONSOLE
00042 :   ELSE
00043 :     @ 21, 0 CLEAR TO 22,79
00044 :     ALERT( "Impresora con problemas", { "Cancela" } )
00045 :   ENDIF
00046 : ENDIF
00047 : @ 21, 0 CLEAR TO 22,79
00048 :
```

```
00049 : SET DELETED OFF
00050 : RELEASE ALL LIKE cm*
00051 :
00052 : DBCLEARRELATION()
00053 : CloseDbf( "LOCAL" )
00054 : CloseDbf( "DEPTO" )
00055 :
00056 : RETURN
00057 :
00058 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : AUXLOCAL.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: auxlocal
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo Auxiliar de Localidades
00011 : *
00012 : *****
00013 :
00014 : #include "Depto.Ch"
00015 : #include "Local.Ch"
00016 :
00017 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00018 :
00019 : PRIVATE aDepto, aLocal
00020 : PRIVATE cmDepto, cmCodLoc, cmLocalidad
00021 :
00022 : DO Pantalla WITH "Archivo Auxiliar de Localidades", "AuxLocal"
00023 :
00024 : aDepto    := OpenDbf( "DEPTO", aIndDepto, 1 )
00025 : aLocal    := OpenDbf( "LOCAL", aIndLocal, 1 )
00026 :
00027 : *SET KEY -9 TO list val // Programa Tecla de Función
00028 :
00029 : ABM( "Localidad", { || PLocal0() }, { || PLocal1() },;
00030 :      { || PLocal2() }, { || PLocal2() }, { || bAltaLocal() },;
00031 :      { || dbDelete() }, { || bModiLocal() }, NIL )
00032 :
00033 : SET KEY -9 TO // Programa Tecla de Función
00034 : RELEASE ALL LIKE cm*
00035 :
00036 : CloseDbf( "LOCAL" )
00037 : CloseDbf( "DEPTO" )
00038 :
00039 : RETURN
00040 :
00041 :
00042 : FUNCTION bAltaLocal
00043 :
00044 : REPLACE DEPTO WITH cmDepto,;
00045 :          CODLOC WITH cmCodLoc
00046 :
00047 : RETURN
00048 :
```

```
00049 :  
00050 : Func bModiLocal()  
00051 :  
00052 : REPLACE LOCALIDAD WITH cmLocalidad  
00053 :  
00054 : Return  
00055 :  
00056 :  
00057 : FUNCTION PLocal0()  
00058 :  
00059 : @ 4, 1 SAY "Codigo Departamento:"  
00060 : @ 5, 1 SAY "Codigo Localidad...:"  
00061 : @ 6, 1 SAY "Nombre Localidad...:"  
00062 :  
00063 : RETURN  
00064 :  
00065 :  
00066 : FUNCTION PLocal1()  
00067 :  
00068 : PRIVATE KNUMERO1, KNUMERO2  
00069 :  
00070 : KNUMERO1 := " "  
00071 : KNUMERO2 := " "  
00072 : iniMLocal()  
00073 :  
00074 : @ 5, 23 CLEAR TO 19, 79  
00075 :  
00076 : * 4, 22 SAY CHR(18)  
00077 : @ 4, 23 GET KNUMERO1 PICTURE "99";  
00078 :     VALID valexist( "DEPTO", KNUMERO1,;  
00079 :     "Departamento Inexistente", { || showDepto( 4 ) } )  
00080 : * 5, 22 SAY CHR(18)  
00081 : @ 5, 23 GET KNUMERO2 PICTURE "999" && VALID valxxxxxx( @KNUMERO )  
00082 : READ  
00083 : CLEAR GETS  
00084 :  
00085 : cmDepto := RIGHT( "00"+ALLTRIM(KNUMERO1),2)  
00086 : cmCodLoc := RIGHT( "000"+ALLTRIM(KNUMERO2),3)  
00087 :  
00088 : RETURN( cmDepto+cmCodLoc )  
00089 :  
00090 :  
00091 : FUNCTION PLocal2()  
00092 :  
00093 : IF FOUND()  
00094 :     loadLocal()  
00095 : ENDIF  
00096 :
```

```
*00097 : @ 6, 23 GET cmLocalidad
00098 :
00099 : RETURN
00100 :
00101 :
00102 : FUNCTION iniMLocal()
00103 :
00104 : cmLocalidad := EVAL( bBlankField, aLocal, "LOCALIDAD" )
00105 :
00106 : Return
00107 :
00108 :
00109 : Func loadLocal()
00110 :
00111 : cmLocalidad := LOCALIDAD
00112 :
00113 : Return
00114 :
00115 :
00116 : FUNCTION showLocal( nlin )
00117 :
00118 : @ nlin, 30 SAY LOCAL->LOCALIDAD COLOR "N/W"
00119 :
00120 : RETURN
00121 :
00122 :
```



```
00001 :
00002 : /* *****
00003 : *
00004 : *      Archivo : CFUNC.C
00005 : *      Sistema : Libreria de Procedimientos y Funciones.
00006 : *      Programa: cfunc.c
00007 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00008 : *      Fecha:    18 de Octubre de 1.993
00009 : *
00010 : *      Descripción:
00011 : *                  Libreria de funciones extendidas para
00012 : *                  CLIPPER 5.01
00013 : *
00014 : ***** */
00015 :
00016 : #include "extend.h"
00017 :
00018 : /* *****
00019 : *
00020 : * Función:      NumDAIEEE()
00021 : *
00022 : * Descripción:
00023 : *              Convierte un número dado en formato de punto flotante
00024 : *              a notación IEEE (4 Bytes)
00025 : *
00026 : * Parametros CLIPPER: [1] double      número en notación punto
00027 : *                      flotante.
00028 : *
00029 : * Retorna CLIPPER:    [1] string[4]  número en notación IEEE
00030 : *
00031 : ***** */
00032 :
00033 : CLIPPER NumDAIEEE()
00034 : {
00035 :     XDOUBLE numero;
00036 :     float flotante;
00037 :     char *_cadena;
00038 :     if (PCOUNT == 1 && ISNUM(1) )
00039 :     {
00040 :         numero = parnd(1);
00041 :         cadena = xalloc(4);
00042 :         flotante = numero;
00043 :         cadena = (char *)&flotante;
00044 :         retcrlen( cadena, 4 );
00045 :         xunlock();
00046 :     }
00047 :     else
00048 :         retc("Error");
```

```

00049 : }
00050 :
00051 :
00052 : /* *****
00053 : *
00054 : * Función:      NumLAIEEE()
00055 : *
00056 : * Descripción:
00057 : *      Convierte un número dado en formato entero largo
00058 : *      a notación IEEE (4 Bytes)
00059 : *
00060 : * Parametros CLIPPER:  [1] long      número en notación entero
00061 : *                      largo.
00062 : *
00063 : * Retorna CLIPPER:     [1] string[4] número en notación IEEE
00064 : *
00065 : ***** */
00066 :
00067 : CLIPPER NumLAIEEE()
00068 : {
00069 :     long numero;
00070 :     long largo;
00071 :     char * cadena;
00072 :     if (PCOUNT == 1 && ISNUM(1) )
00073 :     {
00074 :         numero = parnl(1);
00075 :         largo = numero;
00076 :         cadena = xalloc(4);
00077 :         cadena = (char *)& largo;
00078 :         retc( cadena, 4 );
00079 :         xunlock();
00080 :     }
00081 :     else
00082 :         retc("Error");
00083 : }
00084 :
00085 :
00086 : /* *****
00087 : *
00088 : * Función:      MakeDir()
00089 : *
00090 : * Descripción:
00091 : *      Crea un directorio.
00092 : *
00093 : * Parametros CLIPPER:  [1] string    nombre del directorio a
00094 : *                      crear.
00095 : *
00096 : * Retorna CLIPPER:     [1] int      codigo de resultado:

```

```
00097 : *                               -1 --> Si error          *
00098 : *                               0  --> Satisfactorio      *
00099 : *                               *
00100 : ***** */
00101 :
00102 : CLIPPER MakeDir()
00103 : {
00104 :     char *str;
00105 :     int i;
00106 :
00107 :     if (PCOUNT == 1 && ISCHAR(1) )
00108 :     {
00109 :         str = _parc(1);
00110 :         i = mkdir( str );
00111 :         retni( i );
00112 :     }
00113 :     else
00114 :         retni( -1 );
00115 : }
00116 :
00117 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : CLIENTE1.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: cliente1
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Emite Listado del Archivo Auxiliar de
00011 : *      Clientes.
00012 : *
00013 : *****
00014 :
00015 : #include "Clientes.Ch"
00016 :
00017 : PRIVATE aClie, cmForm
00018 :
00019 : DO Pantalla WITH "Listado de Clientes", "Cliente1"
00020 : SETCOLOR( kClrstd )
00021 :
00022 : aClie      := OpenDbf( "CLIENTES", aIndClie, 1 )
00023 :
00024 : SET DELETED ON
00025 : @ 4, 0 CLEAR TO 19,79
00026 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00027 :   @ 21, 0 CLEAR TO 22,79
00028 :   @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00029 :   * 22, 0 SAY "Pulse <Esc> para interrumpir"
00030 :   IF ISPRINTER()
00031 :     SET CONSOLE OFF
00032 :     SET PRINTER ON
00033 :     QQOUT( CHR(27)+CHR(64) )
00034 :     SET CONSOLE ON
00035 :     SET PRINTER OFF
00036 :     cmForm := kFormDir+"\CLIENTES"
00037 :     REPORT FORM &cmForm TO PRINT NOCONSOLE
00038 :   ELSE
00039 :     @ 21, 0 CLEAR TO 22,79
00040 :     ALERT( "Impresora con problemas", { "Cancela" } )
00041 :   ENDIF
00042 : ENDIF
00043 : @ 21, 0 CLEAR TO 22,79
00044 :
00045 : SET DELETED OFF
00046 : RELEASE ALL LIKE cm*
00047 :
00048 : CloseDbf( "CLIENTES" )
```

00049 :  
00050 : RETURN  
00051 :  
00052 :

Programa: CLIENTES.PRG - 1

```

00001 : *****
00002 : *
00003 : *      Archivo : CLIENTES.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: clientes
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    30 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo Auxiliar de Clientes.
00011 : *
00012 : *****
00013 :
00014 : #include "Clientes.Ch"
00015 :
00016 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00017 :
00018 : PRIVATE aClie
00019 : PRIVATE cmCodigo, cmDenomina
00020 :
00021 : DO Pantalla WITH "Clientes", "Clientes"
00022 :
00023 : aClie      := OpenDbf( "CLIENTES", aIndClie, 1 )
00024 :
00025 : *SET KEY -9 TO list_val // Programa Tecla de Función
00026 :
00027 : ABM( "Cliente", { || PClie0() }, { || PClie1() },;
00028 :      { || PClie2() }, { || PClie2() }, { || bAltaClie() },;
00029 :      { || dbDelete() }, { || bModiClie() }, NIL )
00030 :
00031 : SET KEY -9 TO // Programa Tecla de Función
00032 : RELEASE ALL LIKE cm*
00033 :
00034 : CloseDbf( "CLIENTES" )
00035 :
00036 : RETURN
00037 :
00038 :
00039 : FUNCTION bAltaClie
00040 :
00041 : REPLACE CODIGO WITH cmCodigo
00042 :
00043 : RETURN
00044 :
00045 :
00046 : Func bModiClie()
00047 :
00048 : REPLACE DENOMINA WITH cmDenomina

```

```
00049 :
00050 : Return
00051 :
00052 :
00053 : FUNCTION PClie0()
00054 :
00055 : @ 4, 1 SAY "Codigo de Cliente..:"
00056 : @ 5, 1 SAY "Denominación.....:"
00057 :
00058 : RETURN
00059 :
00060 :
00061 : FUNCTION PClie1()
00062 :
00063 : PRIVATE KNUMERO
00064 :
00065 : KNUMERO := " "
00066 : iniMClie()
00067 :
00068 : @ 4, 23 CLEAR TO 19, 79
00069 :
00070 : * 4, 22 SAY CHR(18)
00071 : @ 4, 23 GET KNUMERO PICTURE "999" && VALID valxxxxxx( @KNUMERO )
00072 : READ
00073 : CLEAR GETS
00074 :
00075 : cmCodigo := RIGHT( "000"+ALLTRIM(KNUMERO),3)
00076 :
00077 : RETURN( cmCodigo )
00078 :
00079 :
00080 : FUNCTION PClie2()
00081 :
00082 : IF FOUND()
00083 :     loadClie()
00084 : ENDIF
00085 :
00086 : @ 5, 23 GET cmDenomina
00087 :
00088 : RETURN
00089 :
00090 :
00091 : FUNCTION iniMClie()
00092 :
00093 : cmDenomina := EVAL( bBlankField, aClie, "DENOMINA" )
00094 :
00095 : Return
00096 :
```

Programa: CLIENTES.PRG - 3

```
00097 :  
00098 : Func loadClie()  
00099 :  
00100 : cmDenomina := DENOMINA  
00101 :  
00102 : Return  
00103 :  
00104 :  
00105 : FUNCTION showClie( nlin )  
00106 :  
00107 : @ nlin, 30 SAY CLIENTES->DENOMINA COLOR "N/W"  
00108 :  
00109 : RETURN  
00110 :  
00111 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo : CONDAP.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: condap
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    03 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Ingresar Condiciones de Aptitud del Agua para
00011 : *                  cada Uso.
00012 : *
00013 : *****
00014 :
00015 : #include "DeterId.Ch"
00016 : #include "CondAp.Ch"
00017 :
00018 : LOCAL cOldColor, RESP
00019 :
00020 : #define MY HSEP      "----"
00021 : #define MY CSEP      " ; "
00022 :
00023 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00024 :
00025 : #include "aCondAp.Ch"
00026 : PRIVATE aDeterId, aCondApB
00027 : PRIVATE cmDeterm, cmDeterDes, cmUniPri
00028 : PRIVATE cmCodUso, cmValorMin, cmValorMax
00029 :
00030 : #include "aTipoUso.Ch"
00031 :
00032 : PRIVATE aSelParm
00033 : aSelParm := { ;
00034 :     { "KCLAVE", "V", ;
00035 :       { { "A", "D", "R", "G", "I", "S", "O" }, aCondAp };
00036 :     },;
00037 :     { "CMDETERM", "F", "DETERID", 1, ;
00038 :       { "DETERM", "DETERDES" },;
00039 :       { "Nº", "DENOMINACION" }, { || NIL },;
00040 :       { || NIL }, "", 35 ;
00041 :     };
00042 :   }
00043 :
00044 :
00045 : DO Pantalla WITH "Parámetros de Aptitud de Aguas", " CondAp"
00046 :
00047 : aDeterId := OpenDbf( "DETERID", aIndDeterId, 1 )
00048 : aCondApB := OpenDbf( "CONDAP", aIndCondAp, 1 )
```

```
00049 :
00050 : SET DELETED ON
00051 :
00052 : SET KEY -9 TO list_val // Programa Tecla de Función
00053 :
00054 : cOldColor := SETCOLOR( kClrstd )
00055 : @ 21, 0 CLEAR TO 22,79
00056 : PCondA00()
00057 :
00058 : DO WHILE .T.
00059 :     RESP := " "
00060 :     IF (cmCodUso := PCondA01()) != " " .AND. LASTKEY() != 27
00061 :
00062 :         ABMItem( 11,cmCodUso, { || cmCodUso==CODUSO }, { || PCondA0() },;
00063 :             { |x,y,z| PCondA1(x,y,z) }, { || PCondA2() },;
00064 :             { |x,y,z| PCondA3(x,y,z) }, { || bAltaCondA() },;
00065 :             { || dbDelete() }, { || bModiCondA() } )
00066 :     ENDIF
00067 :
00068 :     @ 21, 0 CLEAR TO 22,79
00069 :     RESP = MenuABM( "OT", "" )
00070 :     IF RESP == "T"
00071 :         EXIT
00072 :     ENDIF
00073 : ENDDO
00074 :
00075 : @ 21, 0 CLEAR TO 22,79
00076 :
00077 : SET KEY -9 TO // Programa Tecla de Función
00078 : RELEASE ALL LIKE cm*
00079 :
00080 : SETCOLOR( cOldColor )
00081 : SET DELETED OFF
00082 :
00083 : CloseDbf( "CONDAP" )
00084 : CloseDbf( "DETERID" )
00085 :
00086 : RETURN
00087 :
00088 :
00089 : FUNCTION bAltaCondA
00090 :
00091 : APPEND BLANK
00092 :
00093 : iniMCondA()
00094 : REPLACE CODUSO WITH cmCodUso,;
00095 :     DETERM WITH cmDeterm,;
00096 :     VALORMIN WITH cmValorMin,;
```

```
00097 :          VALORMAX  WITH cmValorMax
00098 :
00099 : RETURN( RECNO() )
00100 :
00101 :
00102 : FUNCTION iniMConda()
00103 :
00104 : iniMDetId()
00105 : cmValorMin  := 0
00106 : cmValorMax  := 0
00107 :
00108 : Return
00109 :
00110 :
00111 : Func loadConda()
00112 :
00113 : cmDeterm    := DETERM
00114 : valexist( "DETERID", cmDeterm, "", { || loadDetId() } )
00115 : cmValorMin  := VALORMIN
00116 : cmValorMax  := VALORMAX
00117 :
00118 : Return
00119 :
00120 :
00121 : Func bModiConda()
00122 :
00123 : REPLACE DETERM    WITH cmDeterm,;
00124 :          VALORMIN  WITH cmValorMin,;
00125 :          VALORMAX  WITH cmValorMax
00126 :
00127 : Return
00128 :
00129 :
00130 : FUNCTION PCondaA00()
00131 :
00132 : @ 4, 0 CLEAR TO 19,79
00133 :
00134 : @ 4, 1 SAY "Código de Uso.....:"
00135 :
00136 : @ 6, 6,20,74 BOX "+-+!---!"
00137 :
00138 : @ 6, 6, 8,10 BOX "+--!+-+!"
00139 : @ 6,41, 8,52 BOX "---!+-+!"
00140 : @ 6,52, 8,63 BOX "---!+-+!"
00141 :
00142 : @ 8,10,20,41 BOX "+-+!---!"
00143 : @ 8,52,20,63 BOX "+-+!---!"
00144 : @ 8,63,20,74 BOX "+-!!--!"
```

```
00145 :
00146 : @ 7, 7 SAY "Det"
00147 : @ 7,20 SAY "Descripción"
00148 : @ 7,42 SAY "Unid.Prin."
00149 : @ 7,53 SAY "Valor Mín."
00150 : @ 7,64 SAY "Valor Máx."
00151 :
00152 : Return NIL
00153 :
00154 :
00155 : FUNCTION PConda01()
00156 :
00157 : PRIVATE KCLAVE
00158 :
00159 : KCLAVE := " "
00160 :
00161 : @ 4, 23 CLEAR TO 4, 79
00162 :
00163 : * 4, 22 SAY CHR(18)
00164 : * 4, 23 GET KCLAVE PICTURE "!" VALID KCLAVE$"ADRGISO"
00165 : @ 4, 23 GET KCLAVE PICTURE "!" VALID KCLAVE$aTiposUso[1]
00166 : READ
00167 : CLEAR GETS
00168 :
00169 : RETURN( KCLAVE )
00170 :
00171 :
00172 : FUNCTION PConda0()
00173 :
00174 : @ 9, 7 CLEAR TO 19, 9
00175 : @ 9,11 CLEAR TO 19,40
00176 : @ 9,42 CLEAR TO 19,51
00177 : @ 9,53 CLEAR TO 19,62
00178 : @ 9,64 CLEAR TO 19,73
00179 :
00180 : Return NIL
00181 :
00182 :
00183 : FUNCTION PConda1( nReg, nPriReg, cColor )
00184 :
00185 : Local n
00186 :
00187 : n := nReg - nPriReg
00188 :
00189 : loadConda()
00190 :
00191 : @ 9+n, 7 SAY cmDeterm PICTURE "999" COLOR cColor
00192 : @ 9+n,11 SAY cmDeterDes COLOR cColor
```

```
00193 : @ 9+n,42 SAY cmUniPri          COLOR cColor
00194 : @ 9+n,53 SAY cmValorMin  PICTURE "99999.9999" COLOR cColor
00195 : @ 9+n,64 SAY cmValorMax  PICTURE "99999.9999" COLOR cColor
00196 :
00197 : Return NIL
00198 :
00199 :
00200 : FUNCTION PConda2()
00201 :
00202 : PRIVATE cOldclre, cMemscre
00203 : PRIVATE nTop := 5, nLeft := 4, nBottom := 17, nRight := 75
00204 :
00205 : cOldclre := SETCOLOR()
00206 : cMemscre := MakeShadBox( nTop, nLeft, nBottom, nRight )
00207 :
00208 : PBute2t( nTop, nLeft )
00209 : PBute2d( nTop, nLeft )
00210 :
00211 : SETCOLOR( cOldclre )
00212 : RESTSCREEN(nTop,nLeft,nBottom,nRight,cMemscre)
00213 :
00214 : RETURN
00215 :
00216 :
00217 : FUNCTION PConda3( nReg, nPriReg, cColor )
00218 :
00219 : LOCAL n
00220 :
00221 : n := nReg - nPriReg
00222 :
00223 : @ 9+n, 7 GET cmDeterm  PICTURE "999"          COLOR cColor;
00224 :         VALID vaexist( "DETERID", cmDeterm,;
00225 :         "Codigo de Determinación Inexistente", { || loadDetId() } )
00226 : @ 9+n,11 GET cmDeterDes WHEN .F.          COLOR cColor
00227 : @ 9+n,42 GET cmUniPri  WHEN .F.          COLOR cColor
00228 : @ 9+n,53 GET cmValorMin PICTURE "99999.9999" COLOR cColor
00229 : @ 9+n,64 GET cmValorMax PICTURE "99999.9999" COLOR cColor
00230 :
00231 : READ
00232 : CLEAR GETS
00233 :
00234 : RETURN
00235 :
00236 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : CONDAPI.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: condapi
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    04 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Emite Listado de las Condiciones de Aptitud
00011 : *                  de las Aguas para cada tipo de uso.
00012 : *
00013 : *****
00014 :
00015 : #include "DeterId.Ch"
00016 : #include "CondAp.Ch"
00017 : #include "aCondAp.Ch"
00018 :
00019 : PRIVATE aDeterId, aCondAp, cmForm
00020 :
00021 : DO Pantalla WITH "Listado Condiciones Aptitud Aguas", " CondApI"
00022 : SETCOLOR( kclrstd )
00023 :
00024 : aDeterId := OpenDbf( "DETERID", aIndDeterId, 1 )
00025 : aCondAp  := OpenDbf( "CONDAP", aIndCondAp, 1 )
00026 : SET RELATION TO DETERM INTO DETERID
00027 :
00028 : SET DELETED ON
00029 : @ 4, 0 CLEAR TO 19,79
00030 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00031 :   @ 21, 0 CLEAR TO 22,79
00032 :   @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00033 :   * 22, 0 SAY "Pulse <Esc> para interrumpir"
00034 :   IF ISPRINTER()
00035 :     SET CONSOLE OFF
00036 :     SET PRINTER ON
00037 :     QQOUT( CHR(27)+CHR(64)+CHR(27)+CHR(77) )
00038 :     SET CONSOLE ON
00039 :     SET PRINTER OFF
00040 :     cmForm := kFormDir+"\CONDAP"
00041 :     REPORT FORM &cmForm TO PRINT NOCONSOLE
00042 :   ELSE
00043 :     @ 21, 0 CLEAR TO 22,79
00044 :     ALERT( "Impresora con problemas", { "Cancela" } )
00045 :   ENDIF
00046 : ENDIF
00047 : @ 21, 0 CLEAR TO 22,79
00048 :
```

```
00049 : SET DELETED OFF
00050 : RELEASE ALL LIKE cm*
00051 :
00052 : DBCLEARRELATION()
00053 : CloseDbf( "CONDAP" )
00054 : CloseDbf( "DETERID" )
00055 :
00056 : RETURN
00057 :
00058 :
```

```

00001 : *****
00002 : *
00003 : *      Archivo : DETERID.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: deterid
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    14 de Junio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Definiciones de Determinaciones
00011 : *
00012 : *****
00013 :
00014 : #include "DeterId.Ch"
00015 :
00016 : #define MY_HSEP      "---"
00017 : #define MY_CSEP      " | "
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDeterId
00022 : PRIVATE cmDeterm, cmDeterDes, cmUniPri, cmUniSec, cmFactor, cmFactorExp
00023 :
00024 : PRIVATE aSelParm
00025 : aSelParm := { ;
00026 :     { "KNUMERO", "F", "DETERID", 1, ;
00027 :     { "DETERM", "DETERDES" },;
00028 :     { "Nº", "DENOMINACION" }, { || NIL },;
00029 :     { || NIL }, "", 50 ;
00030 : };
00031 : }
00032 :
00033 : DO Pantalla WITH "Definición de Determinaciones", " DeterId"
00034 :
00035 : aDeterId := OpenDbf( "DETERID", aIndDeterId, 1 )
00036 :
00037 : SET KEY -9 TO list_val // Programa Tecla de Función
00038 :
00039 : ABM( "Determinación", { || PDetId0() }, { || PDetId1() },;
00040 :     { || PDetId2() }, { || PDetId2() }, { || bAltaDetId() },;
00041 :     { || dbDelete() }, { || bModiDetId() } )
00042 :
00043 : SET KEY -9 TO // Programa Tecla de Función
00044 : RELEASE ALL LIKE cm*
00045 :
00046 : CloseDbf( "DETERID" )
00047 :
00048 : RETURN

```



```
00049 :  
00050 :  
00051 : FUNCTION bAltaDetId  
00052 :  
00053 : REPLACE DETERM WITH cmDeterm  
00054 :  
00055 : RETURN  
00056 :  
00057 :  
00058 : Func bModiDetId()  
00059 :  
00060 : REPLACE DETERDES WITH cmDeterDes,;  
00061 :         UNIPRI WITH cmUniPri,;  
00062 :         UNISEC WITH cmUniSec,;  
00063 :         FACTOR WITH cmFactor,;  
00064 :         FACTOREXP WITH cmFactorExp  
00065 :  
00066 : Return  
00067 :  
00068 :  
00069 : FUNCTION PDetId0()  
00070 :  
00071 : @ 4, 1 SAY "Determinación.....:"  
00072 : @ 5, 1 SAY "Descripción.....:"  
00073 : @ 6, 1 SAY "Unidad Primaria.....:"  
00074 : @ 7, 1 SAY "Unidad Secundaria...:"  
00075 : @ 8, 1 SAY "Factor Conversión...:"  
00076 :  
00077 : RETURN  
00078 :  
00079 :  
00080 : FUNCTION PDetId1()  
00081 :  
00082 : PRIVATE KNUMERO  
00083 :  
00084 : KNUMERO := " "  
00085 : iniMDetId()  
00086 :  
00087 : @ 5, 23 CLEAR TO 19, 79  
00088 :  
00089 : @ 4, 22 SAY CHR(18)  
00090 : @ 4, 23 GET KNUMERO PICTURE "999" && VALID valxxxxxx( @KNUMERO )  
00091 : READ  
00092 : CLEAR GETS  
00093 :  
00094 : *cmCodCli := IIF( KCLIENT == " ", cmCodCli, KCLIENT )  
00095 : cmDeterm := RIGHT( "000"+ALLTRIM(KNUMERO),3)  
00096 : @ 4, 23 GET cmDeterm
```

```
00097 : CLEAR GETS
00098 :
00099 : RETURN( cmDeterm )
00100 :
00101 :
00102 : FUNCTION PDetId2()
00103 :
00104 : IF FOUND()
00105 :     loadDetId()
00106 : ENDIF
00107 :
00108 : @ 5, 23 GET cmDeterDes
00109 : @ 6, 23 GET cmUniPri
00110 : @ 7, 23 GET cmUniSec
00111 : @ 8, 23 GET cmFactor PICTURE "99999.9999"
00112 : @ 8, 33 SAY "x10~" && COLOR "N/W"
00113 : @ 8, 37 GET cmFactorExp PICTURE "999"
00114 :
00115 : RETURN
00116 :
00117 :
00118 : FUNCTION iniMDetId()
00119 :
00120 : cmDeterDes := EVAL( bBlankField, aDeterId, "DETERDES" )
00121 : cmUniPri   := EVAL( bBlankField, aDeterId, "UNIPRI" )
00122 : cmUniSec   := EVAL( bBlankField, aDeterId, "UNISEC" )
00123 : cmFactor   := 0
00124 : cmFactorExp := 0
00125 :
00126 : Return
00127 :
00128 :
00129 : Func loadDetId()
00130 :
00131 : cmDeterDes := DETERDES
00132 : cmUniPri   := UNIPRI
00133 : cmUniSec   := UNISEC
00134 : cmFactor   := FACTOR
00135 : cmFactorExp := FACTOREXP
00136 :
00137 : Return
00138 :
00139 :
00140 : FUNCTION showDetId( nlin )
00141 :
00142 : @ nlin, 30 SAY DETERDES COLOR "N/W"
00143 :
00144 : RETURN
```

00145 :

00146 :

```
00001 : *****
00002 : *
00003 : *      Archivo : DETERIDI.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: deteridi
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    31 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Emite Listado del Archivo de Determinaciones
00011 : *
00012 : *****
00013 :
00014 : #include "DeterId.Ch"
00015 :
00016 : PRIVATE aDeterId, cmForm
00017 :
00018 : DO Pantalla WITH "Listado de Determinaciones Definidas", "DeterIdI"
00019 : SETCOLOR( kclrstd )
00020 :
00021 : aDeterId := OpenDbf( "DETERID", aIndDeterId, 1 )
00022 :
00023 : SET DELETED ON
00024 : @ 4, 0 CLEAR TO 19,79
00025 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00026 :   @ 21, 0 CLEAR TO 22,79
00027 :   @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00028 :   * 22, 0 SAY "Pulse <Esc> para interrumpir"
00029 :   IF ISPRINTER()
00030 :     SET CONSOLE OFF
00031 :     SET PRINTER ON
00032 :     QQOUT( CHR(27)+CHR(64)+CHR(27)+CHR(77) )
00033 :     SET CONSOLE ON
00034 :     SET PRINTER OFF
00035 :     cmForm := kFormDir+"\DETERID"
00036 :     REPORT FORM &cmForm TO PRINT NOCONSOLE
00037 :   ELSE
00038 :     @ 21, 0 CLEAR TO 22,79
00039 :     ALERT( "Impresora con problemas", { "Cancela" } )
00040 :   ENDIF
00041 : ENDIF
00042 : @ 21, 0 CLEAR TO 22,79
00043 :
00044 : SET DELETED OFF
00045 : RELEASE ALL LIKE cm*
00046 :
00047 : CloseDbf( "DETERID" )
00048 :
```

00049 : RETURN  
00050 :  
00051 :



```

00001 : *****
00002 : *
00003 : *      Archivo : ENSAADMI.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: ensaadmi
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    04 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Datos de Ensayos de Admisión.
00011 : *
00012 : *****
00013 :
00014 : #include "Depto.Ch"
00015 : #include "IdFuente.Ch"
00016 : #include "IdPozo.Ch"
00017 : #include "EnsaAdmi.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aIdFuente, aIdPozo, aEAdmi
00022 : PRIVATE cmDepto
00023 : PRIVATE cmTpoFuente, cmNroSerie, cmDenomina
00024 : PRIVATE cmFecha, cmTipo, cmDuracion, cmPozos
00025 : PRIVATE cmProfund, cmDiamEns, cmCarga, cmCaudalAdm
00026 : PRIVATE cmNivelE, cmNivelD, cmDescenso, cmPermeab
00027 :
00028 : DO Pantalla WITH "Ensayo de Admisión", "EnsaAdmi"
00029 :
00030 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00031 : aIdFuente   := OpenDbf( "IDFUENTE", aIndIdFte, 1 )
00032 : aIdPozo     := OpenDbf( "IDPOZO", aIndIdPozo, 1 )
00033 : aEAdmi      := OpenDbf( "ENSAADMI", aIndEAdmi, 1 )
00034 :
00035 : *SET KEY -9 TO list_val // Programa Tecla de Función
00036 :
00037 : ABM( "Fuente", { || PEAdmi0() }, { || PEAdmi1() },;
00038 :      { || PEAdmi2() }, { || PEAdmi2() }, { || bAltaEAdmi() },;
00039 :      { || dbDelete() }, { || bModiEAdmi() }, NIL )
00040 :
00041 : SET KEY -9 TO // Programa Tecla de Función
00042 : RELEASE ALL LIKE cm*
00043 :
00044 : CloseDbf( "ENSAADMI" )
00045 : CloseDbf( "IDPOZO" )
00046 : CloseDbf( "IDFUENTE" )
00047 : CloseDbf( "DEPTO" )
00048 :

```

```
00049 : RETURN
00050 :
00051 :
00052 : FUNCTION bAltaEAdmi
00053 :
00054 : REPLACE DEPTO      WITH cmDepto,;
00055 :      TPOFUENTE  WITH cmTpoFuente,;
00056 :      NROSERIE   WITH cmNroSerie,;
00057 :      FECHA      WITH cmFecha,;
00058 :      TIPO       WITH cmTipo
00059 :
00060 : RETURN
00061 :
00062 :
00063 : Func bModiEAdmi()
00064 :
00065 : REPLACE DURACION  WITH cmDuracion,;
00066 :      POZOS        WITH cmPozos,;
00067 :      PROFUND      WITH cmProfund,;
00068 :      DIAMENS      WITH cmDiamEns,;
00069 :      CARGA        WITH cmCarga,;
00070 :      CAUDALADM    WITH cmCaudalAdm,;
00071 :      NIVELE       WITH cmNivele,;
00072 :      NIVELD       WITH cmNivelD,;
00073 :      DESCENSO     WITH cmDescenso,;
00074 :      PERMEAB      WITH cmPermeab
00075 :
00076 : Return
00077 :
00078 :
00079 : FUNCTION PEAdmiO()
00080 :
00081 : @ 4, 1 SAY "Codigo Departamento:"
00082 : @ 5, 1 SAY "Tipo de Fuente.....:"
00083 : @ 6, 1 SAY "Número de la Fuente:"
00084 : @ 7, 1 SAY "Fecha Inicio Ensayo:"
00085 : @ 8, 1 SAY "Tipo de Ensayo[C/V]:"
00086 : @ 9, 1 SAY "Duración.....[min]:"
00087 : @ 10, 1 SAY "Cantidad de Pozos..:"
00088 : @ 11, 1 SAY "Prof.Pozo Ensayado.:"
00089 : @ 12, 1 SAY "Diám.Cañería.E.[mm]:"
00090 : @ 13, 1 SAY "Carga Inicial...[m]:"
00091 : @ 14, 1 SAY "Caudal Ensayo.[l/h]:"
00092 : @ 15, 1 SAY "Nivel Estático..[m]:"
00093 : @ 16, 1 SAY "Nivel Dinámico..[m]:"
00094 : @ 17, 1 SAY "Descenso.....[m]:"
00095 : @ 18, 1 SAY "Permeabil..[m3/día]:"
00096 :
```

```
00097 : RETURN
00098 :
00099 :
00100 : FUNCTION PEAdmi()
00101 :
00102 : PRIVATE KNUMERO1, KNUMERO2, KNUMERO3, KNUMERO4, KNUMERO5
00103 :
00104 : KNUMERO1 := " "
00105 : KNUMERO2 := " "
00106 : KNUMERO3 := 0
00107 : KNUMERO4 := CTOD(" / / ")
00108 : KNUMERO5 := " "
00109 : iniMEAdmi()
00110 :
00111 : @ 4, 23 CLEAR TO 19, 79
00112 :
00113 : * 4, 22 SAY CHR(18)
00114 : @ 4, 23 GET KNUMERO1 PICTURE "99";
00115 :         VALID valexist( "DEPTO", KNUMERO1,;
00116 :         "Departamento Inexistente", { || showDepto( 4 ) } )
00117 : * 5, 22 SAY CHR(18)
00118 : @ 5, 23 GET KNUMERO2 PICTURE "!!"
00119 : * 6, 22 SAY CHR(18)
00120 : @ 6, 23 GET KNUMERO3 PICTURE "99999.99";
00121 :         VALID valexist( "IDFUENTE", KNUMERO1+KNUMERO2+STR(KNUMERO3,8,2),;
00122 :         "Fuente Inexistente", { || showIdFte( 6 ) } ) ;
00123 :         .AND. valexist( "IDPOZO", KNUMERO1+KNUMERO2+STR(KNUMERO3,8,2),;
00124 :         "Pozo Inexistente" )
00125 : * 7, 22 SAY CHR(18)
00126 : @ 7, 23 GET KNUMERO4
00127 : * 8, 22 SAY CHR(18)
00128 : @ 8, 23 GET KNUMERO5 PICTURE "!" VALID KNUMERO5$"CV"
00129 : READ
00130 : CLEAR GETS
00131 :
00132 : cmDepto      := RIGHT( "00"+ALLTRIM(KNUMERO1),2)
00133 : cmTpoFuente := KNUMERO2
00134 : cmNroSerie  := STR( KNUMERO3, 8, 2 )
00135 : cmFecha     := KNUMERO4
00136 : cmTipo      := KNUMERO5
00137 :
00138 : RETURN( cmDepto+cmTpoFuente+cmNroSerie+DTOS(cmFecha)+cmTipo )
00139 :
00140 :
00141 : FUNCTION PEAdmi2()
00142 :
00143 : IF FOUND()
00144 :     loadEAdmi()
```



```
00145 : ENDIF
00146 :
00147 : @ 9, 23 GET cmDuracion PICTURE "999"
00148 : @ 10, 23 GET cmPozos PICTURE "99"
00149 : @ 11, 23 GET cmProfund PICTURE "999.99"
00150 : @ 12, 23 GET cmDiamEns PICTURE "99.99"
00151 : @ 13, 23 GET cmCarga PICTURE "999.99"
00152 : @ 14, 23 GET cmCaudalAdm PICTURE "999.99" WHEN cmTipo=="C"
00153 : @ 15, 23 GET cmNivele PICTURE "999.99"
00154 : @ 16, 23 GET cmNivelD PICTURE "999.99"
00155 : @ 17, 23 GET cmDescenso PICTURE "99.99" WHEN cmTipo=="V"
00156 : @ 18, 23 GET cmPermeab PICTURE "999.9"
00157 :
00158 : RETURN
00159 :
00160 :
00161 : FUNCTION iniMEAdmi()
00162 :
00163 : cmDuracion := 0
00164 : cmPozos := 0
00165 : cmProfund := 0
00166 : cmDiamEns := 0
00167 : cmCarga := 0
00168 : cmCaudalAdm := 0
00169 : cmNivele := 0
00170 : cmNivelD := 0
00171 : cmDescenso := 0
00172 : cmPermeab := 0
00173 :
00174 : Return
00175 :
00176 :
00177 : Func loadEAdmi()
00178 :
00179 : cmDuracion := DURACION
00180 : cmPozos := POZOS
00181 : cmProfund := PROFUND
00182 : cmDiamEns := DIAMENS
00183 : cmCarga := CARGA
00184 : cmCaudalAdm := CAUDALADM
00185 : cmNivele := NIVELE
00186 : cmNivelD := NIVELD
00187 : cmDescenso := DESCENSO
00188 : cmPermeab := PERMEAB
00189 :
00190 : Return
00191 :
00192 :
```

```

00001 : *****
00002 : *
00003 : *      Archivo : ENSABOMB.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: ensabomb
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    06 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Datos de Ensayos de Bombeo.
00011 : *
00012 : *****
00013 :
00014 : #include "Depto.Ch"
00015 : #include "IdFuente.Ch"
00016 : #include "IdPozo.Ch"
00017 : #include "EnsaBomb.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aIdFuente, aIdPozo, aEBomb
00022 : PRIVATE cmDepto
00023 : PRIVATE cmTpoFuente, cmNroSerie, cmDenomina
00024 : PRIVATE cmFecha, cmTipo, cmDuracion, cmPozos, cmCaudal
00025 : PRIVATE cmNivelE, cmNivelD, cmDepresion, cmCaudalEsp
00026 : PRIVATE cmTranM1, cmTranM2, cmTranM3, cmTranM4, cmTranM5
00027 : PRIVATE cmTran1, cmTran2, cmTran3, cmTran4, cmTran5, cmTransmAdop
00028 : PRIVATE cmAlmM1, cmAlmM2, cmAlmM3, cmAlmM4
00029 : PRIVATE cmAlm1, cmAlm2, cmAlm3, cmAlm4, cmAlmacenAdo, cmPermeab
00030 :
00031 : DO Pantalla WITH "Ensayo de Bombeo", "EnsaBomb"
00032 :
00033 : aDepto    := OpenDbf( "DEPTO", aIndDepto, 1 )
00034 : aIdFuente := OpenDbf( "IDFUENTE", aIndIdFte, 1 )
00035 : aIdPozo   := OpenDbf( "IDPOZO", aIndIdPozo, 1 )
00036 : aEBomb    := OpenDbf( "ENSABOMB", aIndEBomb, 1 )
00037 :
00038 : *SET KEY -9 TO list_val // Programa Tecla de Función
00039 :
00040 : ABM( "Fuente", { || PEBomb0() }, { || PEBomb1() },;
00041 :      { || PEBomb2() }, { || PEBomb2() }, { || bAltaEBomb() },;
00042 :      { || dbDelete() }, { || bModiEBomb() }, NIL )
00043 :
00044 : SET KEY -9 TO // Programa Tecla de Función
00045 : RELEASE ALL LIKE cm*
00046 :
00047 : CloseDbf( "ENSABOMB" )
00048 : CloseDbf( "IDPOZO" )

```

```
00049 : CloseDbf( "IDFUENTE" )
00050 : CloseDbf( "DEPTO" )
00051 :
00052 : RETURN
00053 :
00054 :
00055 : FUNCTION bAltaEBomb
00056 :
00057 : REPLACE DEPTO      WITH cmDepto,;
00058 :      TPOFUENTE    WITH cmTpoFuente,;
00059 :      NROSERIE      WITH cmNroSerie,;
00060 :      FECHA         WITH cmFecha,;
00061 :      TIPO          WITH cmTipo
00062 :
00063 : RETURN
00064 :
00065 :
00066 : Func bModiEBomb()
00067 :
00068 : REPLACE DURACION    WITH cmDuracion,;
00069 :      POZOS          WITH cmPozos,;
00070 :      CAUDAL         WITH cmCaudal,;
00071 :      NIVELE         WITH cmNivelE,;
00072 :      NIVELD         WITH cmNivelD,;
00073 :      DEPRESION      WITH cmDepresion,;
00074 :      CAUDALESP      WITH cmCaudalEsp
00075 : REPLACE TRANSMET    WITH cmTranM1+cmTranM2+cmTranM3+cmTranM4+cmTranM5,;
00076 :      TRANSM         WITH STR(cmTran1,4)+STR(cmTran2,4)+STR(cmTran3,4)+;
00077 :                      STR(cmTran4,4)+STR(cmTran5,4).;
00078 :      TRANSMADOP     WITH cmTransmAdop
00079 : REPLACE ALMACENMET  WITH cmAlmM1+cmAlmM2+cmAlmM3+cmAlmM4,;
00080 :      ALMACEN        WITH STR(cmAlm1,10,8)+STR(cmAlm2,10,8)+;
00081 :                      STR(cmAlm3,10,8)+STR(cmAlm4,10,8).;
00082 :      ALMACENADO     WITH cmAlmacenado,;
00083 :      PERMEAB        WITH cmPermeab
00084 :
00085 : Return
00086 :
00087 :
00088 : FUNCTION PEBomb0()
00089 :
00090 : @ 4, 1 SAY "Codigo Departamento:"
00091 : @ 5, 1 SAY "Tipo de Fuente.....:"
00092 : @ 6, 1 SAY "Número de la Fuente:"
00093 : @ 7, 1 SAY "Fecha Inicio Ensayo:"
00094 : @ 8, 1 SAY "Tipo de Ensayo[C/V]:"
00095 : @ 9, 1 SAY "Duración.....[min]:"
00096 : @ 10, 1 SAY "Cantidad de Pozos..:"
```

```
00097 : @ 11, 1 SAY "Caudal Bomb..[m3/h]:"
00098 : @ 12, 1 SAY "Nivel Estático..[m]:"
00099 : @ 12, 41 SAY "Nivel Dinámico..[m]:"
00100 : @ 13, 1 SAY "Depresión.....[m]:"
00101 : @ 14, 1 SAY "Caudal Esp.[m3/h/m]:"
00102 : @ 15, 1 SAY "Met-Transm.[m2/día]:"
00103 : @ 16, 1 SAY "Transm.Adop[m2/día]:"
00104 : @ 17, 1 SAY "Metodo-Almacenam...:"
00105 : @ 18, 1 SAY "Almacenam. Adoptado:"
00106 : @ 19, 1 SAY "Permeabil..[m3/día]:"
00107 :
00108 : RETURN
00109 :
00110 :
00111 : FUNCTION PEBomb1()
00112 :
00113 : PRIVATE KNUMERO1, KNUMERO2, KNUMERO3, KNUMERO4, KNUMERO5
00114 :
00115 : KNUMERO1 := " "
00116 : KNUMERO2 := " "
00117 : KNUMERO3 := 0
00118 : KNUMERO4 := CTOD(" / / ")
00119 : KNUMERO5 := " "
00120 : iniMEBomb()
00121 :
00122 : @ 4, 23 CLEAR TO 11, 79
00123 : @ 12, 23 CLEAR TO 12, 40
00124 : @ 12, 63 CLEAR TO 12, 79
00125 : @ 13, 23 CLEAR TO 19, 79
00126 :
00127 : * 4, 22 SAY CHR(18)
00128 : @ 4, 23 GET KNUMERO1 PICTURE "99";
00129 :     VALID valexist( "DEPTO", KNUMERO1;;
00130 :     "Departamento Inexistente", { || showDepto( 4 ) } )
00131 : * 5, 22 SAY CHR(18)
00132 : @ 5, 23 GET KNUMERO2 PICTURE "!!"
00133 : * 6, 22 SAY CHR(18)
00134 : @ 6, 23 GET KNUMERO3 PICTURE "99999.99";
00135 :     VALID valexist( "IDFUENTE", KNUMERO1+KNUMERO2+STR(KNUMERO3,8,2),;
00136 :     "Fuente Inexistente", { || showIdFte( 6 ) } ) ;
00137 :     .AND. valexist( "IDPOZO", KNUMERO1+KNUMERO2+STR(KNUMERO3,8,2),;
00138 :     "Pozo Inexistente" )
00139 : * 7, 22 SAY CHR(18)
00140 : @ 7, 23 GET KNUMERO4
00141 : * 8, 22 SAY CHR(18)
00142 : @ 8, 23 GET KNUMERO5 PICTURE "!" VALID KNUMERO5$"CV"
00143 : READ
00144 : CLEAR GETS
```

```
00145 :  
00146 : cmDepto      := RIGHT( "00"+ALLTRIM(KNUMERO1),2)  
00147 : cmTpofuente := KNUMERO2  
00148 : cmNroSerie  := STR( KNUMERO3, 8, 2 )  
00149 : cmFecha     := KNUMERO4  
00150 : cmTipo      := KNUMERO5  
00151 :  
00152 : RETURN( cmDepto+cmTpoFuente+cmNroSerie+DTOS(cmFecha)+cmTipo )  
00153 :  
00154 :  
00155 : FUNCTION PEBomb2()  
00156 :  
00157 : IF FOUND()  
00158 :     loadEBomb()  
00159 : ENDIF  
00160 :  
00161 : @ 9, 23  GET  cmDuracion  PICTURE "999"  
00162 : @ 10, 23 GET  cmPozos    PICTURE "99"  
00163 : @ 11, 23 GET  cmCaudal   PICTURE "99.999"  
00164 : @ 12, 23 GET  cmNivelE   PICTURE "999.99"  
00165 : @ 12, 63 GET  cmNivelD   PICTURE "999.99"  
00166 : @ 13, 23 GET  cmDepresion PICTURE "99.99"    && WHEN cmTipo=="C"  
00167 : @ 14, 23 GET  cmCaudalEsp PICTURE "99.99"  
00168 : @ 15, 23 GET  cmTranM1   PICTURE "9" RANGE 0,9  
00169 : @ 15, 25 GET  cmTran1    PICTURE "9999"  
00170 : @ 15, 33 GET  cmTranM2   PICTURE "9" RANGE 0,9  
00171 : @ 15, 35 GET  cmTran2    PICTURE "9999"  
00172 : @ 15, 43 GET  cmTranM3   PICTURE "9" RANGE 0,9  
00173 : @ 15, 45 GET  cmTran3    PICTURE "9999"  
00174 : @ 15, 53 GET  cmTranM4   PICTURE "9" RANGE 0,9  
00175 : @ 15, 55 GET  cmTran4    PICTURE "9999"  
00176 : @ 15, 63 GET  cmTranM5   PICTURE "9" RANGE 0,9  
00177 : @ 15, 65 GET  cmTran5    PICTURE "9999"  
00178 : @ 16, 23 GET  cmTransmAdop PICTURE "9999"  
00179 : @ 17, 23 GET  cmAlmM1    PICTURE "9" RANGE 0,9  
00180 : @ 17, 25 GET  cmAlm1     PICTURE "9.99999999"  
00181 : @ 17, 37 GET  cmAlmM2    PICTURE "9" RANGE 0,9  
00182 : @ 17, 39 GET  cmAlm2     PICTURE "9.99999999"  
00183 : @ 17, 51 GET  cmAlmM3    PICTURE "9" RANGE 0,9  
00184 : @ 17, 53 GET  cmAlm3     PICTURE "9.99999999"  
00185 : @ 17, 65 GET  cmAlmM4    PICTURE "9" RANGE 0,9  
00186 : @ 17, 67 GET  cmAlm4     PICTURE "9.99999999"  
00187 : @ 18, 23 GET  cmAlmacenAdo PICTURE "9.99999999"  
00188 : @ 19, 23 GET  cmPermeab  PICTURE "999.9"  
00189 :  
00190 : RETURN  
00191 :  
00192 :
```

```
00193 : FUNCTION iniMEBomb()  
00194 :  
00195 : cmDuracion      := 0  
00196 : cmPozos        := 0  
00197 : cmCaudal       := 0  
00198 : cmNiveleE      := 0  
00199 : cmNivelD       := 0  
00200 : cmDepresion    := 0  
00201 : cmCaudalEsp    := 0  
00202 : cmTranM1       := 0  
00203 : cmTran1        := 0  
00204 : cmTranM2       := 0  
00205 : cmTran2        := 0  
00206 : cmTranM3       := 0  
00207 : cmTran3        := 0  
00208 : cmTranM4       := 0  
00209 : cmTran4        := 0  
00210 : cmTranM5       := 0  
00211 : cmTran5        := 0  
00212 : cmTransmAdop   := 0  
00213 : cmAlmM1        := 0  
00214 : cmAlm1         := 0  
00215 : cmAlmM2        := 0  
00216 : cmAlm2         := 0  
00217 : cmAlmM3        := 0  
00218 : cmAlm3         := 0  
00219 : cmAlmM4        := 0  
00220 : cmAlm4         := 0  
00221 : cmAlmacenAdo   := 0  
00222 : cmPermeab      := 0  
00223 :  
00224 : Return  
00225 :  
00226 :  
00227 : Func loadEBomb()  
00228 :  
00229 : cmDuracion      := DURACION  
00230 : cmPozos        := POZOS  
00231 : cmCaudal       := CAUDAL  
00232 : cmNiveleE      := NIVELE  
00233 : cmNivelD       := NIVELD  
00234 : cmDepresion    := DEPRESION  
00235 : cmCaudalEsp    := CAUDALESP  
00236 : cmTranM1       := VAL( SUBSTR( TRANSMET, 1, 1 ) )  
00237 : cmTran1        := VAL( SUBSTR( TRANSM, 1, 4 ) )  
00238 : cmTranM2       := VAL( SUBSTR( TRANSMET, 2, 1 ) )  
00239 : cmTran2        := VAL( SUBSTR( TRANSM, 5, 4 ) )  
00240 : cmTranM3       := VAL( SUBSTR( TRANSMET, 3, 1 ) )
```

```
00241 : cmTran3      := VAL( SUBSTR( TRANSM, 9, 4 ) )
00242 : cmTranM4     := VAL( SUBSTR( TRANSMET, 4, 1 ) )
00243 : cmTran4      := VAL( SUBSTR( TRANSM, 13, 4 ) )
00244 : cmTranM5     := VAL( SUBSTR( TRANSMET, 5, 1 ) )
00245 : cmTran5      := VAL( SUBSTR( TRANSM, 17, 4 ) )
00246 : cmTransmAdop := TRANSMADOP
00247 : cmAlmM1      := VAL( SUBSTR( ALMACENMET, 1, 1 ) )
00248 : cmAlm1       := VAL( SUBSTR( ALMACENMET, 1, 10 ) )
00249 : cmAlmM2      := VAL( SUBSTR( ALMACENMET, 2, 1 ) )
00250 : cmAlm2       := VAL( SUBSTR( ALMACENMET, 11, 10 ) )
00251 : cmAlmM3      := VAL( SUBSTR( ALMACENMET, 3, 1 ) )
00252 : cmAlm3       := VAL( SUBSTR( ALMACENMET, 21, 10 ) )
00253 : cmAlmM4      := VAL( SUBSTR( ALMACENMET, 4, 1 ) )
00254 : cmAlm4       := VAL( SUBSTR( ALMACENMET, 31, 10 ) )
00255 : cmAlmacenAdo := ALMACENADO
00256 : cmPermeab    := PERMEAB
00257 :
00258 : Return
00259 :
00260 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ENSAHFDO.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: ensahfdo
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    02 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Datos de Ensayos Hidráulicos de
00011 : *                  Fin de Obra de los Pozos.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "IdFuente.Ch"
00017 : #include "IdPozo.Ch"
00018 : #include "EnsaHFDO.Ch"
00019 :
00020 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00021 :
00022 : PRIVATE aDepto, aIdFuente, aIdPozo, aEHFDO
00023 : PRIVATE cmDepto
00024 : PRIVATE cmTpoFuente, cmNroSerie, cmDenomina
00025 : PRIVATE cmNivelE, cmNivelD, cmCaudal, cmDepresion
00026 : PRIVATE cmCaudalEsp, cmCaudalRec
00027 :
00028 : DO Pantalla WITH "Ensayo Hidráulico de Fin de Obra", "EnsaHFDO"
00029 :
00030 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00031 : aIdFuente   := OpenDbf( "IDFUENTE", aIndIdFte, 1 )
00032 : aIdPozo     := OpenDbf( "IDPOZO", aIndIdPozo, 1 )
00033 : aEHFDO      := OpenDbf( "ENSAHFDO", aIndEHFDO, 1 )
00034 :
00035 : *SET KEY -9 TO list_val // Programa Tecla de Función
00036 :
00037 : ABM( "Fuente", { || PEHFD00() }, { || PEHFD01() },;
00038 :      { || PEHFD02() }, { || PEHFD02() }, { || bAltaEHFDO() },;
00039 :      { || dbDelete() }, { || bModiEHFDO() }, NIL )
00040 :
00041 : SET KEY -9 TO // Programa Tecla de Función
00042 : RELEASE ALL LIKE cm*
00043 :
00044 : CloseDbf( "ENSAHFDO" )
00045 : CloseDbf( "IDPOZO" )
00046 : CloseDbf( "IDFUENTE" )
00047 : CloseDbf( "DEPTO" )
00048 :
```



```
00049 : RETURN
00050 :
00051 :
00052 : FUNCTION bAltaEHFDO
00053 :
00054 : REPLACE DEPTO      WITH cmDepto,;
00055 :      TPOFUENTE     WITH cmTpoFuente,;
00056 :      NROSERIE       WITH cmNroSerie
00057 :
00058 : RETURN
00059 :
00060 :
00061 : Func bModiEHFDO()
00062 :
00063 : REPLACE NIVELE      WITH cmNivelE,;
00064 :      NIVELD        WITH cmNivelD,;
00065 :      CAUDAL        WITH cmCaudal,;
00066 :      DEPRESION     WITH cmDepresion,;
00067 :      CAUDALESP     WITH cmCaudalEsp,;
00068 :      CAUDALREC     WITH cmCaudalRec
00069 :
00070 : Return
00071 :
00072 :
00073 : FUNCTION PEHFD00()
00074 :
00075 : @ 4, 1 SAY "Codigo Departamento:"
00076 : @ 5, 1 SAY "Tipo de Fuente.....:"
00077 : @ 6, 1 SAY "Número de la Fuente:"
00078 : @ 7, 1 SAY "Nivel Estático..[m]:"
00079 : @ 8, 1 SAY "Nivel Dinámico..[m]:"
00080 : @ 9, 1 SAY "Caudal.....[m3/h]:"
00081 : @ 10, 1 SAY "Depresión.....[m]:"
00082 : @ 11, 1 SAY "Caudal Esp.[m3/h/m]:"
00083 : @ 12, 1 SAY "Caudal Recom.[m3/h]:"
00084 :
00085 : RETURN
00086 :
00087 :
00088 : FUNCTION PEHFD01()
00089 :
00090 : PRIVATE KNUMERO1, KNUMERO2, KNUMERO3
00091 :
00092 : KNUMERO1 := " "
00093 : KNUMERO2 := " "
00094 : KNUMERO3 := 0
00095 : iniMEHFDO()
00096 :
```

```
00097 : @ 4, 23 CLEAR TO 19, 79
00098 :
00099 : * 4, 22 SAY CHR(18)
00100 : @ 4, 23 GET KNUMERO1 PICTURE "99";
00101 :         VALID vaexist( "DEPTO", KNUMERO1,;
00102 :         "Departamento Inexistente", { || showDepto( 4 ) } )
00103 : * 5, 22 SAY CHR(18)
00104 : @ 5, 23 GET KNUMERO2 PICTURE "!!"
00105 : * 6, 22 SAY CHR(18)
00106 : @ 6, 23 GET KNUMERO3 PICTURE "99999.99";
00107 :         VALID vaexist( "IDFUENTE", KNUMERO1+KNUMERO2+STR(KNUMERO3,8,2),;
00108 :         "Fuente Inexistente", { || showIdFte( 6 ) } ) ;
00109 :         .AND. vaexist( "IDPOZO", KNUMERO1+KNUMERO2+STR(KNUMERO3,8,2),;
00110 :         "Pozo Inexistente" )
00111 : READ
00112 : CLEAR GETS
00113 :
00114 : cmDepto      := RIGHT( "00"+ALLTRIM(KNUMERO1),2)
00115 : cmTpofuente := KNUMERO2
00116 : cmNroSerie  := STR( KNUMERO3, 8, 2 )
00117 :
00118 : RETURN( cmDepto+cmTpoFuente+cmNroSerie )
00119 :
00120 :
00121 : FUNCTION PEHFD02()
00122 :
00123 : IF FOUND()
00124 :     loadEHFD0()
00125 : ENDIF
00126 :
00127 : @ 7, 23 GET cmNivelE     PICTURE "99.99"
00128 : @ 8, 23 GET cmNivelD     PICTURE "99.99"
00129 : @ 9, 23 GET cmCaudal     PICTURE "99.999"
00130 : @ 10, 23 GET cmDepresion PICTURE "99.99"
00131 : @ 11, 23 GET cmCaudalEsp PICTURE "99.999"
00132 : @ 12, 23 GET cmCaudalRec PICTURE "99.999"
00133 :
00134 : RETURN
00135 :
00136 :
00137 : FUNCTION iniMEHFD0()
00138 :
00139 : cmNivelE      := 0
00140 : cmNivelD      := 0
00141 : cmCaudal      := 0
00142 : cmDepresion   := 0
00143 : cmCaudalEsp   := 0
00144 : cmCaudalRec   := 0
```

```
00145 :  
00146 : Return  
00147 :  
00148 :  
00149 : Func loadEHFD0()  
00150 :  
00151 : cmNivele      := NIVELE  
00152 : cmNivelD     := NIVELD  
00153 : cmCaudal     := CAUDAL  
00154 : cmDepresion  := DEPRESSION  
00155 : cmCaudalEsp  := CAUDALESP  
00156 : cmCaudalRec  := CAUDALREC  
00157 :  
00158 : Return  
00159 :  
00160 :
```

```

00001 : *****
00002 : *
00003 : *      Archivo : ESTACID.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: estacid
00006 : *      Autor:   Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:   03 de Julio de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo Maestro de Datos Identifi-
00011 : *                  catorios de Estaciones Hidrometeorológicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "Local.Ch"
00017 : #include "EstacId.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aLocal, aEstacId
00022 : PRIVATE cmDepto, cmCodLoc, cmLocalidad
00023 : PRIVATE cmTipoEst, cmCodigoEst, cmDenomina, cmUbicacion
00024 : PRIVATE cmCoordLat, cmCoordLon, cmCoordAlt, cmAnioInst
00025 : PRIVATE cmOrganismo, cmObserva
00026 :
00027 : DO Pantalla WITH "Datos de Estaciones Hidrometeorológicas", " EstacId"
00028 :
00029 : aDepto := OpenDbf( "DEPTO", aIndDepto, 1 )
00030 : aLocal := OpenDbf( "LOCAL", aIndLocal, 1 )
00031 : aEstacId := OpenDbf( "ESTACID", aIndEstId, 1 )
00032 :
00033 : *SET KEY -9 TO list_val // Programa Tecla de Función
00034 :
00035 : ABM( "Estación", { || PEstId0() }, { || PEstId1() },;
00036 :      { || PEstId2() }, { || PEstId2() }, { || bAltaEstId() },;
00037 :      { || dbDelete() }, { || bModiEstId() }, NIL )
00038 :
00039 : SET KEY -9 TO // Programa Tecla de Función
00040 : RELEASE ALL LIKE cm*
00041 :
00042 : CloseDbf( "ESTACID" )
00043 : CloseDbf( "LOCAL" )
00044 : CloseDbf( "DEPTO" )
00045 :
00046 : RETURN
00047 :
00048 :

```

```
00049 : FUNCTION bAltaEstId
00050 :
00051 : REPLACE DEPTO      WITH cmDepto,;
00052 :          TIPOEST    WITH cmTipoEst,;
00053 :          CODIGOEST   WITH cmCodigoEst
00054 :
00055 : RETURN
00056 :
00057 :
00058 : Func bModiEstId()
00059 :
00060 : REPLACE DENOMINA     WITH cmDenomina,;
00061 :          CODLOC      WITH cmCodLoc,;
00062 :          UBICACION   WITH cmUbicacion,;
00063 :          COORDLAT    WITH cmCoordLat,;
00064 :          COORDLON    WITH cmCoordLon,;
00065 :          COORDALT    WITH cmCoordAlt,;
00066 :          ANIOINST    WITH cmAnioInst,;
00067 :          ORGANISMO   WITH cmOrganismo,;
00068 :          OBSERVA     WITH cmObserva
00069 :
00070 : Return
00071 :
00072 :
00073 : FUNCTION PEstId0()
00074 :
00075 : @ 4, 1 SAY "Codigo Departamento:"
00076 : @ 5, 1 SAY "Tipo de Estación..."
00077 : @ 6, 1 SAY "Codigo de Estación:"
00078 : @ 7, 1 SAY "Denominación....."
00079 : @ 8, 1 SAY "Codigo de Localidad:"
00080 : @ 9, 1 SAY "Ubicación....."
00081 : @ 10, 1 SAY "Latitud Geográfica:"
00082 : @ 11, 1 SAY "Longitud Geográfica:"
00083 : @ 12, 1 SAY "Altura sobre el mar:"
00084 : @ 13, 1 SAY "Año de Instalación:"
00085 : @ 14, 1 SAY "Organismo Operador:"
00086 : @ 15, 1 SAY "Observaciones....."
00087 :
00088 : RETURN
00089 :
00090 :
00091 : FUNCTION PEstId1()
00092 :
00093 : PRIVATE KN1, KN2, KN3
00094 :
00095 : KN1 := " "
00096 : KN2 := " "
```

## Programa: ESTACID.PRG - 3

```

00097 : KN3 := 0
00098 : iniMEstId()
00099 :
00100 : @ 5, 23 CLEAR TO 19, 79
00101 :
00102 : * 4, 22 SAY CHR(18)
00103 : @ 4, 23 GET KN1 PICTURE "99";
00104 :         VALID vaexist( "DEPTO", RIGHT( "00"+ALLTRIM(KN1),2),;
00105 :         "Departamento Inexistente", { || showDepto( 4 ) } )
00106 : * 5, 22 SAY CHR(18)
00107 : @ 5, 23 GET KN2 PICTURE "!" VALID KN2$"AHM"
00108 : * 6, 22 SAY CHR(18)
00109 : @ 6, 23 GET KN3 PICTURE "99999.99" && VALID va|xxxxxx( @KN )
00110 : READ
00111 : CLEAR GETS
00112 :
00113 : cmDepto      := RIGHT( "00"+ALLTRIM(KN1),2)
00114 : cmTipoEst   := KN2
00115 : cmCodigoEst := STR( KN3, 8, 2 )
00116 :
00117 : RETURN( cmDepto+cmTipoEst+cmCodigoEst )
00118 :
00119 :
00120 : FUNCTION PEstId2()
00121 :
00122 : IF FOUND()
00123 :     loadEstId()
00124 : ENDIF
00125 :
00126 : @ 7, 23 GET cmDenomina
00127 : @ 8, 23 GET cmCodLoc PICTURE "999";
00128 :         VALID vaexist( "LOCAL", cmDepto+cmCodLoc,;
00129 :         "Localidad Inexistente", { || showLocal( 8 ) } )
00130 : @ 9, 23 GET cmUbicacion
00131 : @ 10, 23 GET cmCoordLat PICTURE "999.9999"
00132 : @ 11, 23 GET cmCoordLon PICTURE "999.9999"
00133 : @ 12, 23 GET cmcoordAlt PICTURE "9999"
00134 : @ 13, 23 GET cmAnioInst PICTURE "9999"
00135 : @ 14, 23 GET cmOrganismo
00136 : @ 15, 23 GET cmObserva
00137 :
00138 : RETURN
00139 :
00140 :
00141 : FUNCTION iniMEstId()
00142 :
00143 : cmDenomina := EVAL( bBlankField, aEstacId, "DENOMINA" )
00144 : cmCodLoc   := EVAL( bBlankField, aEstacId, "CODLOC" )

```

```
00145 : cmUbicacion := EVAL( bBlankField, aEstacId, "UBICACION" )
00146 : cmCoordLat  := 0
00147 : cmCoordLon  := 0
00148 : cmCoordAlt  := 0
00149 : cmAnioInst  := 0
00150 : cmOrganismo := EVAL( bBlankField, aEstacId, "ORGANISMO" )
00151 : cmObserva   := EVAL( bBlankField, aEstacId, "OBSERVA" )
00152 :
00153 : Return
00154 :
00155 :
00156 : Func loadEstId()
00157 :
00158 : cmDenomina := DENOMINA
00159 : cmCodLoc  := CODLOC
00160 : cmUbicacion := UBICACION
00161 : cmCoordLat := COORDLAT
00162 : cmCoordLon := COORDLON
00163 : cmCoordAlt := COORDALT
00164 : cmAnioInst := ANIOINST
00165 : cmOrganismo := ORGANISMO
00166 : cmObserva  := OBSERVA
00167 :
00168 : Return
00169 :
00170 :
00171 : FUNCTION showEstId( nlin )
00172 :
00173 : @ nlin, 30 SAY DENOMINA COLOR "N/W"
00174 :
00175 : RETURN
00176 :
00177 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTACIDI.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: estacidi
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    12 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Emite Listado de Identificación de las
00011 : *              Estaciones Hidrometeorologicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "Local.Ch"
00017 : #include "EstacId.Ch"
00018 :
00019 : PRIVATE aDepto, aLocal, aEstacId, cmForm
00020 :
00021 : DO Pantalla WITH "Listado Ident.Est.Hidrometeorologicas", "ProTipoI"
00022 : SETCOLOR( kClrstd )
00023 :
00024 : aDepto    := OpenDbf( "DEPTO", aIndDepto, 1 )
00025 : aLocal    := OpenDbf( "LOCAL", aIndLocal, 1 )
00026 : aEstacId  := OpenDbf( "ESTACID", aIndEstId, 2 )
00027 : SET RELATION TO DEPTO INTO DEPTO, TO DEPTO+CODLOC INTO LOCAL
00028 :
00029 : SET DELETED ON
00030 : @ 4, 0 CLEAR TO 19,79
00031 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00032 :     @ 21, 0 CLEAR TO 22,79
00033 :     @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00034 :     * 22, 0 SAY "Pulse <Esc> para interrumpir"
00035 :     IF ISPRINTER()
00036 :         SET CONSOLE OFF
00037 :         SET PRINTER ON
00038 :         QQOUT( CHR(27)+CHR(64)+CHR(27)+CHR(77) )
00039 :         SET CONSOLE ON
00040 :         SET PRINTER OFF
00041 :         cmForm := kFormDir+"\ESTACID"
00042 :         REPORT FORM &cmForm TO PRINT NOCONSOLE
00043 :     ELSE
00044 :         @ 21, 0 CLEAR TO 22,79
00045 :         ALERT( "Impresora con problemas", { "Cancela" } )
00046 :     ENDIF
00047 : ENDIF
00048 : @ 21, 0 CLEAR TO 22,79
```



```
00049 :  
00050 : SET DELETED OFF  
00051 : RELEASE ALL LIKE cm*  
00052 :  
00053 : DBCLEARRELATION()  
00054 : CloseDbf( "ESTACID" )  
00055 : CloseDbf( "LOCAL" )  
00056 : CloseDbf( "DEPTO" )  
00057 :  
00058 : RETURN  
00059 :  
00060 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTHIDVA.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: esthidva
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    01 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo de Valores anuales de
00011 : *                  Estaciones Hidrologicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "EstacId.Ch"
00017 : #include "EstHidVA.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aEstacId, aEstHidVA
00022 : PRIVATE cmDepto, cmCodigoEst, cmFechaUAct, cmPANio, cmUANio
00023 : PRIVATE cmCMoMed, cmAltMed, cmCSoMed, cmAlteElMed
00024 :
00025 : SET CENTURY ON
00026 :
00027 : DO Pantalla WITH "Valores Anuales de Est.Hidrológicas", "EstHidVA"
00028 :
00029 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00030 : aEstacId    := OpenDbf( "ESTACID", aIndEstId, 1 )
00031 : aEstHVA     := OpenDbf( "ESTHIDVA", aIndEHVA, 1 )
00032 :
00033 : *SET KEY -9 TO list_val // Programa Tecla de Función
00034 :
00035 : ABM( "Registro", { || PEHVAO() }, { || PEHVA1() },;
00036 :      { || PEHVA2() }, { || PEHVA2() }, { || bAltaEHVA() },;
00037 :      { || dbDelete() }, { || bModiEHVA() } )
00038 :
00039 : SET KEY -9 TO // Programa Tecla de Función
00040 : SET CENTURY OFF
00041 : RELEASE ALL LIKE cm*
00042 :
00043 : CloseDbf( "ESTHIDVA" )
00044 : CloseDbf( "ESTACID" )
00045 : CloseDbf( "DEPTO" )
00046 :
00047 : RETURN
00048 :
```

```
00049 :
00050 : FUNCTION bAltaEHVA
00051 :
00052 : REPLACE DEPTO      WITH cmDepto.;
00053 :      CODIGOEST WITH cmCodigoEst
00054 :
00055 : RETURN
00056 :
00057 :
00058 : FUNCTION PEHVA0()
00059 :
00060 : @ 4, 1 SAY "Codigo Departamento:"
00061 : @ 5, 1 SAY "Tipo de Estación..."
00062 : @ 6, 1 SAY "Codigo de Estación:"
00063 : @ 7, 1 SAY "Primer Año Serie..."
00064 : @ 8, 1 SAY "Ultimo Año Serie..."
00065 : @ 9, 1 SAY "Fecha Ult.Actualiz.:"
00066 : @ 10, 1 SAY "Cau.Mod.Medio[m3/s]:"
00067 : @ 11, 1 SAY "Elem.Medic.Alt(E/L):"
00068 : @ 12, 1 SAY "Altura.....[m]:"
00069 : @ 13, 1 SAY "Cau.Sól.Medio[Tm/a]:"
00070 :
00071 : RETURN
00072 :
00073 :
00074 : FUNCTION PEHVA1()
00075 :
00076 : PRIVATE KN1, KN2, KN3
00077 :
00078 : KN1 := " "
00079 : KN2 := " "
00080 : KN3 := 0
00081 : iniMEHVA()
00082 :
00083 : @ 4, 23 CLEAR TO 19, 79
00084 :
00085 : * 4, 22 SAY CHR(18)
00086 : @ 4, 23 GET KN1 PICTURE "99";
00087 :      VALID valexist( "DEPTO", KN1,;
00088 :      "Departamento Inexistente", { || showDepto( 4 ) } )
00089 : * 5, 22 SAY CHR(18)
00090 : @ 5, 23 GET KN2 PICTURE "!" VALID KN2$"AM"
00091 : * 6, 22 SAY CHR(18)
00092 : @ 6, 23 GET KN3 PICTURE "99999.99";
00093 :      VALID valexist( "ESTACID", RIGHT( "00"+ALLTRIM(KN1),2)+KN2+;
00094 :      STR( KN3, 8, 2 ),;
00095 :      "Estación Hidrológica Inexistente", NIL )
00096 :
```

```
00097 : READ
00098 : CLEAR GETS
00099 :
00100 : cmDepto      := RIGHT( "00"+ALLTRIM(KN1),2)
00101 : cmTipoEst    := KN2
00102 : cmCodigoEst := STR( KN3, 8, 2 )
00103 :
00104 : RETURN( cmDepto+cmCodigoEst )
00105 :
00106 :
00107 : FUNCTION PEHVA2()
00108 :
00109 : @ 7, 23 GET cmPANio      PICTURE "9999"
00110 : @ 8, 23 GET cmUANio     PICTURE "9999"
00111 : @ 9, 23 GET cmFechaUAct
00112 : @ 10, 23 GET cmCMoMed   PICTURE "999.99"
00113 : @ 11, 23 GET cmAltElMed PICTURE "!"      VALID cmAltElMed$"EL"
00114 : @ 12, 23 GET cmAltMed   PICTURE "999.99"
00115 : @ 13, 23 GET cmCSoMed   PICTURE "999.99"
00116 :
00117 : RETURN
00118 :
00119 :
00120 : Func bModiEHVA()
00121 :
00122 : REPLACE PANIO      WITH cmPANio,;
00123 :      UANIO        WITH cmUANio,;
00124 :      CMOMED       WITH cmCMoMed,;
00125 :      ALTELMED     WITH cmAltElMed,;
00126 :      ALTMED       WITH cmAltMed,;
00127 :      CSOMED       WITH cmCSoMed,;
00128 :      FECHAUACT    WITH cmFechaUact
00129 :
00130 : Return
00131 :
00132 :
00133 : FUNCTION iniMEHVA()
00134 :
00135 : cmPANio      := 0
00136 : cmUANio      := 0
00137 : cmCMoMed     := 0
00138 : cmAltElMed   := " "
00139 : cmAltMed     := 0
00140 : cmCSoMed     := 0
00141 : cmFechaUAct := CTOD( " / / " )
00142 :
00143 : Return
00144 :
```

```
00145 :  
00146 : FUNCTION loadMEHVA()  
00147 :  
00148 : cmPAnio      := PANIO  
00149 : cmUAnio      := UANIO  
00150 : cmCMoMed     := CMOMED  
00151 : cmAlte1Med   := ALTELMED  
00152 : cmAltMed     := ALTMED  
00153 : cmCSoMed     := CSOMED  
00154 : cmFechaUAct := FECHAUACT  
00155 :  
00156 : Return  
00157 :  
00158 :  
00159 :
```

## Programa: ESTHIDVE.PRG - 1

```

00001 : *****
00002 : *
00003 : *      Archivo : ESTHIDVE.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: esthidve
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    01 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *              Actualiza Archivo de Valores extremos de
00011 : *              Estaciones hidrologicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "EstacId.Ch"
00017 : #include "EstHidVE.Ch"
00018 :
00019 : bBlankField := { |aA, cField|, SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aEstacId, aEstHidVE
00022 : PRIVATE cmDepto, cmCodigoEst, cmFechaUAct, cmAnio
00023 : PRIVATE cmCMoMax, cmACMoMx, cmCMoMin, cmACMoMn
00024 : PRIVATE cmAltMax, cmAAltMx, cmAltMin, cmAAltMn, cmAltElMed
00025 : PRIVATE cmCSoMax, cmACSoMx, cmCSoMin, cmACSoMn
00026 :
00027 : SET CENTURY ON
00028 :
00029 : DO Pantalla WITH "Valores Extremos de Est.Hidrológicas", "EstHidVE"
00030 :
00031 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00032 : aEstacId    := OpenDbf( "ESTACID", aIndEstId, 1 )
00033 : aEstHVE     := OpenDbf( "ESTHIDVE", aIndEHVE, 1 )
00034 :
00035 : *SET KEY -9 TO list_val // Programa Tecla de Función
00036 :
00037 : ABM( "Registro", { || PEHVEO() }, { || PEHVE1() },;
00038 :      { || PEHVE2() }, { || PEHVE2() }, { || bAltaEHVE() },;
00039 :      { || dbDelete() }, { || bModiEHVE() } )
00040 :
00041 : SET KEY -9 TO // Programa Tecla de Función
00042 : SET CENTURY OFF
00043 : RELEASE ALL LIKE cm*
00044 :
00045 : CloseDbf( "ESTHIDVE" )
00046 : CloseDbf( "ESTACID" )
00047 : CloseDbf( "DEPTO" )
00048 :

```

```
00049 : RETURN
00050 :
00051 :
00052 : FUNCTION bAltaEIVE
00053 :
00054 : REPLACE DEPTO      WITH cmDepto,;
00055 :      CODIGOEST WITH cmCodigoEst
00056 :
00057 : RETURN
00058 :
00059 :
00060 : FUNCTION PEHVEO()
00061 :
00062 : @ 4, 1 SAY "Codigo Departamento:"
00063 : @ 5, 1 SAY "Tipo de Estación..."
00064 : @ 6, 1 SAY "Codigo de Estación:"
00065 : @ 7, 1 SAY "Fecha Ult.Actualiz.:"
00066 : @ 8, 1 SAY "Caud.Mod.Máx.[m3/s]:"
00067 : @ 8, 41 SAY "Año de Ocurrencia..."
00068 : @ 9, 1 SAY "Caud.Mod.Mín.[m3/s]:"
00069 : @ 9, 41 SAY "Año de Ocurrencia..."
00070 : @ 10, 1 SAY "Elem.Med.Altur(E/L):"
00071 : @ 11, 1 SAY "Altura Máxima...[m]:"
00072 : @ 11, 41 SAY "Año de Ocurrencia..."
00073 : @ 12, 1 SAY "Altura Mínima...[m]:"
00074 : @ 12, 41 SAY "Año de Ocurrencia..."
00075 : @ 13, 1 SAY "Caud.Sól.Máx.[Tm/a]:"
00076 : @ 13, 41 SAY "Año de Ocurrencia..."
00077 : @ 14, 1 SAY "Caud.Sól.Mín.[Tm/a]:"
00078 : @ 14, 41 SAY "Año de Ocurrencia..."
00079 :
00080 : RETURN
00081 :
00082 :
00083 : FUNCTION PEHVE1()
00084 :
00085 : PRIVATE KN1, KN2, KN3
00086 :
00087 : KN1 := " "
00088 : KN2 := " "
00089 : KN3 := 0
00090 : iniMEHVE()
00091 :
00092 : @ 4, 23 CLEAR TO 7, 79
00093 : @ 8, 23 CLEAR TO 19, 40
00094 : @ 8, 63 CLEAR TO 19, 79
00095 :
00096 : * 4, 22 SAY CHR(18)
```

## Programa: ESTHIDVE.PRG - 3

```

00097 : @ 4, 23 GET KN1 PICTURE "99";
00098 :         VALID valexist( "DEPTO", KN1,;
00099 :         "Departamento Inexistente", { || showDepto( 4 ) } )
00100 : * 5, 22 SAY CHR(18)
00101 : @ 5, 23 GET KN2 PICTURE "!" VALID KN2$"AM"
00102 : * 6, 22 SAY CHR(18)
00103 : @ 6, 23 GET KN3 PICTURE "99999.99";
00104 :         VALID valexist( "ESTACID", RIGHT( "00"+ALLTRIM(KN1),2)+KN2+;
00105 :         STR( KN3, 8, 2 ),;
00106 :         "Estación Hidrológica Inexistente", NIL )
00107 :
00108 : READ
00109 : CLEAR GETS
00110 :
00111 : cmDepto      := RIGHT( "00"+ALLTRIM(KN1),2)
00112 : cmTipoEst    := KN2
00113 : cmCodigoEst := STR( KN3, 8, 2 )
00114 :
00115 : RETURN( cmDepto+cmCodigoEst )
00116 :
00117 :
00118 : FUNCTION PEHVE2()
00119 :
00120 : @ 7, 23 GET cmFechaUAct
00121 : @ 8, 23 GET cmCMoMax      PICTURE "999.99"
00122 : @ 8, 63 GET cmACMoMx      PICTURE "9999"
00123 : @ 9, 23 GET cmCMoMin      PICTURE "999.99"
00124 : @ 9, 63 GET cmACMoMn      PICTURE "9999"
00125 : @ 10, 23 GET cmAltElMed    PICTURE "!" VALID cmAltElMed$"EL"
00126 : @ 11, 23 GET cmAltMax      PICTURE "999.99"
00127 : @ 11, 63 GET cmAAltMx      PICTURE "9999"
00128 : @ 12, 23 GET cmAltMin      PICTURE "999.99"
00129 : @ 12, 63 GET cmAAltMn      PICTURE "9999"
00130 : @ 13, 23 GET cmCSoMax      PICTURE "999.99"
00131 : @ 13, 63 GET cmACSoMx      PICTURE "9999"
00132 : @ 14, 23 GET cmCSoMin      PICTURE "999.99"
00133 : @ 14, 63 GET cmACSoMn      PICTURE "9999"
00134 :
00135 : RETURN
00136 :
00137 :
00138 : Func bModiEHVE()
00139 :
00140 : REPLACE CMOMAX      WITH cmCMoMax,;
00141 :         ACMOMX      WITH cmACMoMx,;
00142 :         CMOMIN      WITH cmCMoMin,;
00143 :         ACMOMN      WITH cmACMoMn,;
00144 :         ALTELMED    WITH cmAltElMed,;

```



```
00145 :      ALTMAX      WITH cmAltMax,;
00146 :      AALTMX      WITH cmAAltMx,;
00147 :      ALTMIN      WITH cmAltMin,;
00148 :      AALTMN      WITH cmAAltMn,;
00149 :      CSOMAX      WITH cmCSoMax,;
00150 :      ACSOMX      WITH cmACSoMx,;
00151 :      CSOMIN      WITH cmCSoMin,;
00152 :      ACSOMN      WITH cmACSoMn,;
00153 :      FECHAUACT WITH cmFechaUact
00154 :
00155 : Return
00156 :
00157 :
00158 : FUNCTION iniMEHVE()
00159 :
00160 : cmCMoMax      := 0
00161 : cmACMoMx      := 0
00162 : cmCMoMin      := 0
00163 : cmACMoMn      := 0
00164 : cmAltElMed    := " "
00165 : cmAltMax      := 0
00166 : cmAAltMx      := 0
00167 : cmAltMin      := 0
00168 : cmAAltMn      := 0
00169 : cmCSoMax      := 0
00170 : cmACSoMx      := 0
00171 : cmCSoMin      := 0
00172 : cmACSoMn      := 0
00173 : cmFechaUAct  := CTOD( " / / " )
00174 :
00175 : Return
00176 :
00177 :
00178 : FUNCTION loadMEHVE()
00179 :
00180 : cmCMoMax      := CMOMAX
00181 : cmACMoMx      := ACMOMX
00182 : cmCMoMin      := CMOMIN
00183 : cmACMoMn      := ACMOMN
00184 : cmAltElMed    := ALTELMED
00185 : cmAltMax      := ALTMAX
00186 : cmAAltMx      := AALTMX
00187 : cmAltMin      := ALTMIN
00188 : cmAAltMn      := AALTMN
00189 : cmCSoMax      := CSOMAX
00190 : cmACSoMx      := ACSOMX
00191 : cmCSoMin      := CSOMIN
00192 : cmACSoMn      := ACSOMN
```

```
00193 : cmFechaUAct := FECHAUACT
00194 :
00195 : Return
00196 :
00197 :
00198 :
```

```

00001 : *****
00002 : *
00003 : *      Archivo : ESTHIDVI.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: esthidvi
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    13 de Octubre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *      Imprime informe de todos los Valores de las
00011 : *      Estaciones Hidrologicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "EstacId.Ch"
00017 : #include "EstHidVM.Ch"
00018 : #include "EstHidVA.Ch"
00019 : #include "EstHidVE.Ch"
00020 :
00021 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00022 :
00023 : PRIVATE aDepto, aEstacId, aEstHidVM, aEstHidVA, aEstHidVE, cClave
00024 : PRIVATE cmDepto, cmCodigoEst, cmFechaUAct, cmAnio
00025 : PRIVATE cmMed1, cmMed2, cmMed3, cmMed4
00026 : PRIVATE cmMed5, cmMed6, cmMed7, cmMed8
00027 : PRIVATE cmMed9, cmMed10, cmMed11, cmMed12
00028 : PRIVATE cmMax1, cmMax2, cmMax3, cmMax4
00029 : PRIVATE cmMax5, cmMax6, cmMax7, cmMax8
00030 : PRIVATE cmMax9, cmMax10, cmMax11, cmMax12
00031 : PRIVATE cmAMx1, cmAMx2, cmAMx3, cmAMx4
00032 : PRIVATE cmAMx5, cmAMx6, cmAMx7, cmAMx8
00033 : PRIVATE cmAMx9, cmAMx10, cmAMx11, cmAMx12
00034 : PRIVATE cmMin1, cmMin2, cmMin3, cmMin4
00035 : PRIVATE cmMin5, cmMin6, cmMin7, cmMin8
00036 : PRIVATE cmMin9, cmMin10, cmMin11, cmMin12
00037 : PRIVATE cmAMn1, cmAMn2, cmAMn3, cmAMn4
00038 : PRIVATE cmAMn5, cmAMn6, cmAMn7, cmAMn8
00039 : PRIVATE cmAMn9, cmAMn10, cmAMn11, cmAMn12
00040 :
00041 : PRIVATE aSelParm
00042 : aSelParm := { ;
00043 :     { "KN1", "F", "DEPTO", 1, ;
00044 :     { "DEPTO", "NOMDEPTO" },;
00045 :     { "Nº", "DEPTO." }, { || NIL },;
00046 :     { || showDepto( 4 ) }, "", 22 ;
00047 :     },;
00048 :     { "KN2", "V",;

```

```
00049 :      { { "A", "H" }, { "Hidrometeorologica", "Hidrologica" } } ;
00050 :      },;
00051 :      { "KN3", "F", "ESTACID", 1, ;
00052 :      { "CODIGOEST", "DENOMINA" },;
00053 :      { "ESTACION", "DENOMINACION" },;
00054 :      { || KN1 == DEPTO .AND. KN2 == TPOFUENTE },;
00055 :      { || showEstId( 6 ) }, "", 43 ;
00056 :      };
00057 :      }
00058 :
00059 : SET CENTURY ON
00060 :
00061 : DO Pantalla WITH "Informe de Valores de Est.Hidrol.", "EstHidVI"
00062 :
00063 : cOldColor := SETCOLOR( kclrstd )
00064 :
00065 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00066 : aEstHVM     := OpenDbf( "ESTHIDVM", aIndEHVM, 1 )
00067 : aEstHVE     := OpenDbf( "ESTHIDVE", aIndEHVE, 1 )
00068 : aEstHVA     := OpenDbf( "ESTHIDVA", aIndEHVA, 1 )
00069 : aEstacId    := OpenDbf( "ESTACID", aIndEstId, 1 )
00070 :
00071 : SET RELATION TO DEPTO INTO DEPTO
00072 :
00073 : SET KEY -9 TO list val // Programa Tecla de Función
00074 :
00075 : DO WHILE .T.
00076 :     SELECT( "ESTACID" )
00077 :     @ 4, 0 CLEAR TO 19,79
00078 :     @ 21, 0 CLEAR TO 22,79
00079 :     PEHVIO()
00080 :     cClave := PEHVI1()
00081 :     IF LASTKEY() == 27
00082 :         EXIT
00083 :     ENDIF
00084 :     PEHVI2( cClave )
00085 : ENDDO
00086 :
00087 : SET KEY -9 TO // Programa Tecla de Función
00088 : SET CENTURY OFF
00089 : RELEASE ALL LIKE cm*
00090 :
00091 : SET RELATION TO
00092 :
00093 : CloseDbf( "ESTHIDVA" )
00094 : CloseDbf( "ESTHIDVE" )
00095 : CloseDbf( "ESTHIDVM" )
00096 : CloseDbf( "ESTACID" )
```

```
00097 : CloseDbf( "DEPTO" )
00098 :
00099 : SETCOLOR( cOldColor )
00100 :
00101 : RETURN
00102 :
00103 :
00104 : FUNCTION PEHVIO()
00105 :
00106 : @ 4, 1 SAY "Departamento.....:"
00107 : @ 5, 1 SAY "Tipo de Estación..."
00108 : @ 6, 1 SAY "Codigo de Estación :"
00109 :
00110 : RETURN
00111 :
00112 :
00113 : FUNCTION PEHVII()
00114 :
00115 : PRIVATE KN1, KN2, KN3
00116 :
00117 : KN1 := " "
00118 : KN2 := " "
00119 : KN3 := " "
00120 :
00121 : @ 4, 22 CLEAR TO 19, 79
00122 :
00123 : @ 4, 22 SAY CHR(18)
00124 : @ 4, 23 GET KN1 PICTURE "99";
00125 :         VALID valexist( "DEPTO", KN1,;
00126 :         "Departamento Inexistente", { || showDepto( 4 ) } )
00127 : @ 5, 22 SAY CHR(18)
00128 : @ 5, 23 GET KN2 PICTURE "!" VALID KN2$"AH"
00129 : @ 6, 22 SAY CHR(18)
00130 : @ 6, 23 GET KN3 PICTURE "99999.99";
00131 :         VALID valexist("ESTACID",RIGHT("00"+ALLTRIM(KN1),2)+KN2+KN3;
00132 :         "Estación Hidrológica Inexistente", NIL )
00133 :
00134 : READ
00135 : CLEAR GETS
00136 :
00137 : cmDepto      := RIGHT( "00"+ALLTRIM(KN1),2)
00138 : cmTipoEst    := KN2
00139 : cmCodigoEst  := KN3
00140 :
00141 : RETURN( cmDepto+cmCodigoEst )
00142 :
00143 :
00144 : FUNCTION PEHVI2( cClave )
```

```
00145 :
00146 : IF MenuABM( "SN", "[Confirma la impresión]" ) == "S"
00147 :   @ 21, 0 CLEAR TO 22,79
00148 :   @ 21, 0 SAY "Imprimiendo..." COLOR "N*/W"
00149 :   * 22, 0 SAY "Pulse <Esc> para interrumpir"
00150 :   IF ISPRINTER()
00151 :     IEncEH( 1 )
00152 :     IVaIEH( cClave )
00153 :   ELSE
00154 :     @ 21, 0 CLEAR TO 22,79
00155 :     ALERT( "Impresora con problemas", { "Cancela" } )
00156 :   ENDIF
00157 : ENDIF
00158 : @ 21, 0 CLEAR TO 22,79
00159 :
00160 : RETURN
00161 :
00162 :
00163 : FUNCTION IEncEH( nPag )
00164 :
00165 : SET DEVICE TO PRINTER
00166 :
00167 : @ 0, 11 SAY DATE()
00168 : @ 0, 88 SAY "Hoja:"
00169 : @ 0, 94 SAY nPag PICTURE "99"
00170 : @ 2, 42 SAY "PROVINCIA DEL NEUQUEN"
00171 : @ 3, 38 SAY "BANCO DE DATOS HIDROGEOLOGICOS"
00172 : @ 4, 38 SAY "===== "
00173 : @ 6, 11 SAY "Codigo Departamento:"
00174 : @ 6, 33 SAY DEPTO PICTURE "99"
00175 : @ 6, 45 SAY DEPTO->NOMDEPTO
00176 : @ 7, 11 SAY "Tipo de Estación...:"
00177 : @ 7, 33 SAY TIPOEST PICTURE "X"
00178 : @ 7, 45 SAY IIF( TIPOEST == "A", "Hidrometeorológica",;
00179 :                  "Hidrológica" )
00180 : @ 8, 11 SAY "Codigo de Estación.:"
00181 : @ 8, 33 SAY CODIGOEST PICTURE "99999.99"
00182 : @ 9, 11 SAY "Denominación.....:"
00183 : @ 9, 33 SAY DENOMINA
00184 : @ 11, 11 SAY REPLICATE("-",84)
00185 :
00186 : SET DEVICE TO SCREEN
00187 :
00188 : RETURN
00189 :
00190 :
00191 : FUNCTION IVaIEH( cClave )
00192 :
```

```
00193 : LOCAL IVM, IVA, IVE
00194 :
00195 : IVM := IVA := IVE := .F.
00196 :
00197 : SELECT( "ESTHIDVM" )
00198 : SEEK( cClave )
00199 : IVM := FOUND() .AND. !DELETED()
00200 :
00201 : SELECT( "ESTHIDVA" )
00202 : SEEK( cClave )
00203 : IVA := FOUND() .AND. !DELETED()
00204 :
00205 : SELECT( "ESTHIDVE" )
00206 : SEEK( cClave )
00207 : IVE := FOUND() .AND. !DELETED()
00208 :
00209 : SELECT( "ESTACID" )
00210 :
00211 : SET DEVICE TO PRINTER
00212 :
00213 : @ PROW()+2, 11 SAY "DATOS DE CAUDAL MODULO"
00214 :
00215 : @ PROW()+2, 11 SAY "Valor Medio Máximo :"
00216 : IF IVE
00217 :     @ PROW(), 33 SAY ESTHIDVE->CMOMAX
00218 :     @ PROW(), 40 SAY "m3/seg"
00219 :     @ PROW(), 50 SAY "Año..:"
00220 :     @ PROW(), 57 SAY ESTHIDVE->ACMOMX
00221 : ELSE
00222 :     @ PROW(), 33 SAY "Sin datos"
00223 : ENDIF
00224 :
00225 : @ PROW()+1, 11 SAY "Valor Medio Mínimo :"
00226 : IF IVE
00227 :     @ PROW(), 33 SAY ESTHIDVE->CMOMIN
00228 :     @ PROW(), 40 SAY "m3/seg"
00229 :     @ PROW(), 50 SAY "Año..:"
00230 :     @ PROW(), 57 SAY ESTHIDVE->ACMOMN
00231 : ELSE
00232 :     @ PROW(), 33 SAY "Sin datos"
00233 : ENDIF
00234 :
00235 : @ PROW()+1, 11 SAY "Valor Medio Anual..:"
00236 : IF IVA
00237 :     @ PROW(), 33 SAY ESTHIDVA->CMOMED
00238 :     @ PROW(), 40 SAY "m3/seg"
00239 :     @ PROW(), 50 SAY "Serie:"
00240 :     @ PROW(), 57 SAY ESTHIDVA->PANIO
```

```
00241 :    @ PROW(), 62 SAY "-"  
00242 :    @ PROW(), 64 SAY ESTHIDVA->UANIO  
00243 : ELSE  
00244 :    @ PROW(), 33 SAY "Sin datos"  
00245 : ENDIF  
00246 :  
00247 : @ PROW()+1, 11 SAY "Valores Mensuales [m3/seg]"  
00248 : IF 1VM  
00249 :    IValVH1( 1 )  
00250 :    IValVH2( 1 )  
00251 : ELSE  
00252 :    @ PROW(), 33 SAY "Sin datos"  
00253 : ENDIF  
00254 :  
00255 : @ PROW()+2, 11 SAY "DATOS DE ALTURA"  
00256 :  
00257 : @ PROW()+2, 11 SAY "Valor Medio Máximo :"  
00258 : IF 1VE  
00259 :    @ PROW(), 33 SAY ESTHIDVE->ALTMAX  
00260 :    @ PROW(), 40 SAY "m"  
00261 :    @ PROW(), 50 SAY "Año..:"  
00262 :    @ PROW(), 57 SAY ESTHIDVE->AALTMX  
00263 : ELSE  
00264 :    @ PROW(), 33 SAY "Sin datos"  
00265 : ENDIF  
00266 :  
00267 : @ PROW()+1, 11 SAY "Valor Medio Mínimo :"  
00268 : IF 1VE  
00269 :    @ PROW(), 33 SAY ESTMETVE->ALTMIN  
00270 :    @ PROW(), 40 SAY "m"  
00271 :    @ PROW(), 50 SAY "Año..:"  
00272 :    @ PROW(), 57 SAY ESTHIDVE->AALTMN  
00273 : ELSE  
00274 :    @ PROW(), 33 SAY "Sin datos"  
00275 : ENDIF  
00276 :  
00277 : @ PROW()+1, 11 SAY "Valor Medio Anual..:"  
00278 : IF 1VA  
00279 :    @ PROW(), 33 SAY ESTHIDVA->ALTMED  
00280 :    @ PROW(), 40 SAY "m"  
00281 :    @ PROW(), 50 SAY "Serie:"  
00282 :    @ PROW(), 57 SAY ESTHIDVA->PANIO  
00283 :    @ PROW(), 62 SAY "-"  
00284 :    @ PROW(), 64 SAY ESTHIDVA->UANIO  
00285 : ELSE  
00286 :    @ PROW(), 33 SAY "Sin datos"  
00287 : ENDIF  
00288 :
```



```
00289 : @ PROW()+1, 11 SAY "Valores Mensuales [m]"
00290 : IF 1VM
00291 :     IValVH1( 2 )
00292 :     IValVH2( 2 )
00293 : ELSE
00294 :     @ PROW(), 33 SAY "Sin datos"
00295 : ENDIF
00296 :
00297 : IF 1VM
00298 :     IEncEH( 2 )
00299 : ENDIF
00300 :
00301 : @ PROW()+2, 11 SAY "DATOS DE CAUDAL SOLIDO"
00302 :
00303 : @ PROW()+2, 11 SAY "Valor Medio Máximo :"
00304 : IF 1VE
00305 :     @ PROW(), 33 SAY ESTHIDVE->CSOMAX
00306 :     @ PROW(), 40 SAY "Tm/año"
00307 :     @ PROW(), 50 SAY "Año..:"
00308 :     @ PROW(), 57 SAY ESTHIDVE->ACSOMX
00309 : ELSE
00310 :     @ PROW(), 33 SAY "Sin datos"
00311 : ENDIF
00312 :
00313 : @ PROW()+1, 11 SAY "Valor Medio Mínimo :"
00314 : IF 1VE
00315 :     @ PROW(), 33 SAY ESTHIDVE->CSOMIN
00316 :     @ PROW(), 40 SAY "Tm/año"
00317 :     @ PROW(), 50 SAY "Año..:"
00318 :     @ PROW(), 57 SAY ESTHIDVE->ACSOMN
00319 : ELSE
00320 :     @ PROW(), 33 SAY "Sin datos"
00321 : ENDIF
00322 :
00323 : @ PROW()+1, 11 SAY "Valor Medio Anual..:"
00324 : IF 1VA
00325 :     @ PROW(), 33 SAY ESTHIDVA->CSOMED
00326 :     @ PROW(), 40 SAY "Tm/año"
00327 :     @ PROW(), 50 SAY "Serie:"
00328 :     @ PROW(), 57 SAY ESTHIDVA->PANIO
00329 :     @ PROW(), 62 SAY "-"
00330 :     @ PROW(), 64 SAY ESTHIDVA->UANIO
00331 : ELSE
00332 :     @ PROW(), 33 SAY "Sin datos"
00333 : ENDIF
00334 :
00335 : @ PROW()+1, 11 SAY "Valores Mensuales [Tm/mes]"
00336 : IF 1VM
```

```
00337 :    IValVH1( 3 )
00338 :    IValVH2( 3 )
00339 : ELSE
00340 :    @ PROW(), 33 SAY "Sin datos"
00341 : ENDIF
00342 :
00343 : SET DEVICE TO SCREEN
00344 :
00345 : EJECT
00346 :
00347 : RETURN
00348 :
00349 :
00350 : FUNCTION IValVH1( nParam )
00351 :
00352 : SELECT( "ESTHIDVM" )
00353 :
00354 : DO CASE
00355 :     CASE nParam == 1
00356 :         cmMed1 := CMOMED1
00357 :         cmMed2 := CMOMED2
00358 :         cmMed3 := CMOMED3
00359 :         cmMed4 := CMOMED4
00360 :         cmMed5 := CMOMED5
00361 :         cmMed6 := CMOMED6
00362 :         cmMed7 := CMOMED7
00363 :         cmMed8 := CMOMED8
00364 :         cmMed9 := CMOMED9
00365 :         cmMed10 := CMOMED10
00366 :         cmMed11 := CMOMED11
00367 :         cmMed12 := CMOMED12
00368 :
00369 :         cmMax1 := CMOMAX1
00370 :         cmMax2 := CMOMAX2
00371 :         cmMax3 := CMOMAX3
00372 :         cmMax4 := CMOMAX4
00373 :         cmMax5 := CMOMAX5
00374 :         cmMax6 := CMOMAX6
00375 :         cmMax7 := CMOMAX7
00376 :         cmMax8 := CMOMAX8
00377 :         cmMax9 := CMOMAX9
00378 :         cmMax10 := CMOMAX10
00379 :         cmMax11 := CMOMAX11
00380 :         cmMax12 := CMOMAX12
00381 :
00382 :         cmAMx1 := ACMOMX1
00383 :         cmAMx2 := ACMOMX2
00384 :         cmAMx3 := ACMOMX3
```

```
00385 :      cmAMx4      := ACMOMX4
00386 :      cmAMx5      := ACMOMX5
00387 :      cmAMx6      := ACMOMX6
00388 :      cmAMx7      := ACMOMX7
00389 :      cmAMx8      := ACMOMX8
00390 :      cmAMx9      := ACMOMX9
00391 :      cmAMx10     := ACMOMX10
00392 :      cmAMx11     := ACMOMX11
00393 :      cmAMx12     := ACMOMX12
00394 :
00395 :      cmMin1       := CMOMIN1
00396 :      cmMin2       := CMOMIN2
00397 :      cmMin3       := CMOMIN3
00398 :      cmMin4       := CMOMIN4
00399 :      cmMin5       := CMOMIN5
00400 :      cmMin6       := CMOMIN6
00401 :      cmMin7       := CMOMIN7
00402 :      cmMin8       := CMOMIN8
00403 :      cmMin9       := CMOMIN9
00404 :      cmMin10      := CMOMIN10
00405 :      cmMin11      := CMOMIN11
00406 :      cmMin12      := CMOMIN12
00407 :
00408 :      cmAMn1       := ACMOMN1
00409 :      cmAMn2       := ACMOMN2
00410 :      cmAMn3       := ACMOMN3
00411 :      cmAMn4       := ACMOMN4
00412 :      cmAMn5       := ACMOMN5
00413 :      cmAMn6       := ACMOMN6
00414 :      cmAMn7       := ACMOMN7
00415 :      cmAMn8       := ACMOMN8
00416 :      cmAMn9       := ACMOMN9
00417 :      cmAMn10      := ACMOMN10
00418 :      cmAMn11      := ACMOMN11
00419 :      cmAMn12      := ACMOMN12
00420 :
00421 :      CASE nParam == 2
00422 :      cmMed1       := ALTMED1
00423 :      cmMed2       := ALTMED2
00424 :      cmMed3       := ALTMED3
00425 :      cmMed4       := ALTMED4
00426 :      cmMed5       := ALTMED5
00427 :      cmMed6       := ALTMED6
00428 :      cmMed7       := ALTMED7
00429 :      cmMed8       := ALTMED8
00430 :      cmMed9       := ALTMED9
00431 :      cmMed10      := ALTMED10
00432 :      cmMed11      := ALTMED11
```

```
00433 :      cmMed12  := ALTMED12
00434 :
00435 :      cmMax1    := ALTMAX1
00436 :      cmMax2    := ALTMAX2
00437 :      cmMax3    := ALTMAX3
00438 :      cmMax4    := ALTMAX4
00439 :      cmMax5    := ALTMAX5
00440 :      cmMax6    := ALTMAX6
00441 :      cmMax7    := ALTMAX7
00442 :      cmMax8    := ALTMAX8
00443 :      cmMax9    := ALTMAX9
00444 :      cmMax10   := ALTMAX10
00445 :      cmMax11   := ALTMAX11
00446 :      cmMax12   := ALTMAX12
00447 :
00448 :      cmAMx1     := AALTMX1
00449 :      cmAMx2     := AALTMX2
00450 :      cmAMx3     := AALTMX3
00451 :      cmAMx4     := AALTMX4
00452 :      cmAMx5     := AALTMX5
00453 :      cmAMx6     := AALTMX6
00454 :      cmAMx7     := AALTMX7
00455 :      cmAMx8     := AALTMX8
00456 :      cmAMx9     := AALTMX9
00457 :      cmAMx10    := AALTMX10
00458 :      cmAMx11    := AALTMX11
00459 :      cmAMx12    := AALTMX12
00460 :
00461 :      cmMin1     := ALTMIN1
00462 :      cmMin2     := ALTMIN2
00463 :      cmMin3     := ALTMIN3
00464 :      cmMin4     := ALTMIN4
00465 :      cmMin5     := ALTMIN5
00466 :      cmMin6     := ALTMIN6
00467 :      cmMin7     := ALTMIN7
00468 :      cmMin8     := ALTMIN8
00469 :      cmMin9     := ALTMIN9
00470 :      cmMin10    := ALTMIN10
00471 :      cmMin11    := ALTMIN11
00472 :      cmMin12    := ALTMIN12
00473 :
00474 :      cmAMn1     := AALTMN1
00475 :      cmAMn2     := AALTMN2
00476 :      cmAMn3     := AALTMN3
00477 :      cmAMn4     := AALTMN4
00478 :      cmAMn5     := AALTMN5
00479 :      cmAMn6     := AALTMN6
00480 :      cmAMn7     := AALTMN7
```

```
00481 :      cmAMn8      := AALTMN8
00482 :      cmAMn9      := AALTMN9
00483 :      cmAMn10     := AALTMN10
00484 :      cmAMn11     := AALTMN11
00485 :      cmAMn12     := AALTMN12
00486 :
00487 :      CASE nParam == 3
00488 :      cmMed1      := CSOMED1
00489 :      cmMed2      := CSOMED2
00490 :      cmMed3      := CSOMED3
00491 :      cmMed4      := CSOMED4
00492 :      cmMed5      := CSOMED5
00493 :      cmMed6      := CSOMED6
00494 :      cmMed7      := CSOMED7
00495 :      cmMed8      := CSOMED8
00496 :      cmMed9      := CSOMED9
00497 :      cmMed10     := CSOMED10
00498 :      cmMed11     := CSOMED11
00499 :      cmMed12     := CSOMED12
00500 :
00501 :      cmMax1      := CSOMAX1
00502 :      cmMax2      := CSOMAX2
00503 :      cmMax3      := CSOMAX3
00504 :      cmMax4      := CSOMAX4
00505 :      cmMax5      := CSOMAX5
00506 :      cmMax6      := CSOMAX6
00507 :      cmMax7      := CSOMAX7
00508 :      cmMax8      := CSOMAX8
00509 :      cmMax9      := CSOMAX9
00510 :      cmMax10     := CSOMAX10
00511 :      cmMax11     := CSOMAX11
00512 :      cmMax12     := CSOMAX12
00513 :
00514 :      cmAMx1      := ACSOMX1
00515 :      cmAMx2      := ACSOMX2
00516 :      cmAMx3      := ACSOMX3
00517 :      cmAMx4      := ACSOMX4
00518 :      cmAMx5      := ACSOMX5
00519 :      cmAMx6      := ACSOMX6
00520 :      cmAMx7      := ACSOMX7
00521 :      cmAMx8      := ACSOMX8
00522 :      cmAMx9      := ACSOMX9
00523 :      cmAMx10     := ACSOMX10
00524 :      cmAMx11     := ACSOMX11
00525 :      cmAMx12     := ACSOMX12
00526 :
00527 :      cmMin1      := CSOMIN1
00528 :      cmMin2      := CSOMIN2
```

```
00577 :      @ PROW(), 67 SAY "Año "
00578 :      CASE nParam == 3
00579 :      @ PROW(), 23 SAY "C.Solid.Med."
00580 :      @ PROW(), 36 SAY "C.Solid.Máx."
00581 :      @ PROW(), 49 SAY "Año "
00582 :      @ PROW(), 54 SAY "C.Solid.Mín."
00583 :      @ PROW(), 67 SAY "Año "
00584 : ENDCASE
00585 :
00586 : @ PROW()+1, 10 SAY REPLICATE( "-", 43 )
00587 :
00588 : @ PROW()+1, 11 SAY "Enero"
00589 : @ PROW(), 26 SAY cmMed1  PICTURE "999.99"
00590 : @ PROW(), 39 SAY cmMax1  PICTURE "999.99"
00591 : @ PROW(), 49 SAY cmAMx1  PICTURE "9999"
00592 : @ PROW(), 57 SAY cmMin1  PICTURE "999.99"
00593 : @ PROW(), 67 SAY cmAMn1  PICTURE "9999"
00594 :
00595 : @ PROW()+1, 11 SAY "Febrero"
00596 : @ PROW(), 26 SAY cmMed2  PICTURE "999.99"
00597 : @ PROW(), 39 SAY cmMax2  PICTURE "999.99"
00598 : @ PROW(), 49 SAY cmAMx2  PICTURE "9999"
00599 : @ PROW(), 57 SAY cmMin2  PICTURE "999.99"
00600 : @ PROW(), 67 SAY cmAMn2  PICTURE "9999"
00601 :
00602 : @ PROW()+1, 11 SAY "Marzo"
00603 : @ PROW(), 26 SAY cmMed3  PICTURE "999.99"
00604 : @ PROW(), 39 SAY cmMax3  PICTURE "999.99"
00605 : @ PROW(), 49 SAY cmAMx3  PICTURE "9999"
00606 : @ PROW(), 57 SAY cmMin3  PICTURE "999.99"
00607 : @ PROW(), 67 SAY cmAMn3  PICTURE "9999"
00608 :
00609 : @ PROW()+1, 11 SAY "Abril"
00610 : @ PROW(), 26 SAY cmMed4  PICTURE "999.99"
00611 : @ PROW(), 39 SAY cmMax4  PICTURE "999.99"
00612 : @ PROW(), 49 SAY cmAMx4  PICTURE "9999"
00613 : @ PROW(), 57 SAY cmMin4  PICTURE "999.99"
00614 : @ PROW(), 67 SAY cmAMn4  PICTURE "9999"
00615 :
00616 : @ PROW()+1, 11 SAY "Mayo"
00617 : @ PROW(), 26 SAY cmMed5  PICTURE "999.99"
00618 : @ PROW(), 39 SAY cmMax5  PICTURE "999.99"
00619 : @ PROW(), 49 SAY cmAMx5  PICTURE "9999"
00620 : @ PROW(), 57 SAY cmMin5  PICTURE "999.99"
00621 : @ PROW(), 67 SAY cmAMn5  PICTURE "9999"
00622 :
00623 : @ PROW()+1, 11 SAY "Junio"
00624 : @ PROW(), 26 SAY cmMed6  PICTURE "999.99"
```

```
00625 : @ PROW(), 39 SAY cmMax6 PICTURE "999.99"
00626 : @ PROW(), 49 SAY cmAMx6 PICTURE "9999"
00627 : @ PROW(), 57 SAY cmMin6 PICTURE "999.99"
00628 : @ PROW(), 67 SAY cmAMn6 PICTURE "9999"
00629 :
00630 : @ PROW()+1, 11 SAY "Julio"
00631 : @ PROW(), 26 SAY cmMed7 PICTURE "999.99"
00632 : @ PROW(), 39 SAY cmMax7 PICTURE "999.99"
00633 : @ PROW(), 49 SAY cmAMx7 PICTURE "9999"
00634 : @ PROW(), 57 SAY cmMin7 PICTURE "999.99"
00635 : @ PROW(), 67 SAY cmAMn7 PICTURE "9999"
00636 :
00637 : @ PROW()+1, 11 SAY "Agosto"
00638 : @ PROW(), 26 SAY cmMed8 PICTURE "999.99"
00639 : @ PROW(), 39 SAY cmMax8 PICTURE "999.99"
00640 : @ PROW(), 49 SAY cmAMx8 PICTURE "9999"
00641 : @ PROW(), 57 SAY cmMin8 PICTURE "999.99"
00642 : @ PROW(), 67 SAY cmAMn8 PICTURE "9999"
00643 :
00644 : @ PROW()+1, 11 SAY "Setiembre"
00645 : @ PROW(), 26 SAY cmMed9 PICTURE "999.99"
00646 : @ PROW(), 39 SAY cmMax9 PICTURE "999.99"
00647 : @ PROW(), 49 SAY cmAMx9 PICTURE "9999"
00648 : @ PROW(), 57 SAY cmMin9 PICTURE "999.99"
00649 : @ PROW(), 67 SAY cmAMn9 PICTURE "9999"
00650 :
00651 : @ PROW()+1, 11 SAY "Octubre"
00652 : @ PROW(), 26 SAY cmMed10 PICTURE "999.99"
00653 : @ PROW(), 39 SAY cmMax10 PICTURE "999.99"
00654 : @ PROW(), 49 SAY cmAMx10 PICTURE "9999"
00655 : @ PROW(), 57 SAY cmMin10 PICTURE "999.99"
00656 : @ PROW(), 67 SAY cmAMn10 PICTURE "9999"
00657 :
00658 : @ PROW()+1, 11 SAY "Noviembre"
00659 : @ PROW(), 26 SAY cmMed11 PICTURE "999.99"
00660 : @ PROW(), 39 SAY cmMax11 PICTURE "999.99"
00661 : @ PROW(), 49 SAY cmAMx11 PICTURE "9999"
00662 : @ PROW(), 57 SAY cmMin11 PICTURE "999.99"
00663 : @ PROW(), 67 SAY cmAMn11 PICTURE "9999"
00664 :
00665 : @ PROW()+1, 11 SAY "Diciembre"
00666 : @ PROW(), 26 SAY cmMed12 PICTURE "999.99"
00667 : @ PROW(), 39 SAY cmMax12 PICTURE "999.99"
00668 : @ PROW(), 49 SAY cmAMx12 PICTURE "9999"
00669 : @ PROW(), 57 SAY cmMin12 PICTURE "999.99"
00670 : @ PROW(), 67 SAY cmAMn12 PICTURE "9999"
00671 :
00672 : @ PROW()+1, 10 SAY REPLICATE( "-", 43 )
```

00673 :  
00674 : RETURN  
00675 :  
00676 :



```
00001 : *****
00002 : *
00003 : *      Archivo : ESTHIDVM.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: esthidvm
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    31 de Agosto de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo de Valores mensuales de
00011 : *                  Estaciones Hidrologicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "EstacId.Ch"
00017 : #include "EstHidVM.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aEstacId, aEstHidVM
00022 : PRIVATE cmDepto, cmCodigoEst, cmFechaUAct, cmAnio, cmAltElMed
00023 : PRIVATE cmCMoMed1, cmCMoMed2, cmCMoMed3, cmCMoMed4
00024 : PRIVATE cmCMoMed5, cmCMoMed6, cmCMoMed7, cmCMoMed8
00025 : PRIVATE cmCMoMed9, cmCMoMed10, cmCMoMed11, cmCMoMed12
00026 : PRIVATE cmCMoMax1, cmCMoMax2, cmCMoMax3, cmCMoMax4
00027 : PRIVATE cmCMoMax5, cmCMoMax6, cmCMoMax7, cmCMoMax8
00028 : PRIVATE cmCMoMax9, cmCMoMax10, cmCMoMax11, cmCMoMax12
00029 : PRIVATE cmACMoMx1, cmACMoMx2, cmACMoMx3, cmACMoMx4
00030 : PRIVATE cmACMoMx5, cmACMoMx6, cmACMoMx7, cmACMoMx8
00031 : PRIVATE cmACMoMx9, cmACMoMx10, cmACMoMx11, cmACMoMx12
00032 : PRIVATE cmCMoMin1, cmCMoMin2, cmCMoMin3, cmCMoMin4
00033 : PRIVATE cmCMoMin5, cmCMoMin6, cmCMoMin7, cmCMoMin8
00034 : PRIVATE cmCMoMin9, cmCMoMin10, cmCMoMin11, cmCMoMin12
00035 : PRIVATE cmACMoMn1, cmACMoMn2, cmACMoMn3, cmACMoMn4
00036 : PRIVATE cmACMoMn5, cmACMoMn6, cmACMoMn7, cmACMoMn8
00037 : PRIVATE cmACMoMn9, cmACMoMn10, cmACMoMn11, cmACMoMn12
00038 : PRIVATE cmAltMed1, cmAltMed2, cmAltMed3, cmAltMed4
00039 : PRIVATE cmAltMed5, cmAltMed6, cmAltMed7, cmAltMed8
00040 : PRIVATE cmAltMed9, cmAltMed10, cmAltMed11, cmAltMed12
00041 : PRIVATE cmAltMax1, cmAltMax2, cmAltMax3, cmAltMax4
00042 : PRIVATE cmAltMax5, cmAltMax6, cmAltMax7, cmAltMax8
00043 : PRIVATE cmAltMax9, cmAltMax10, cmAltMax11, cmAltMax12
00044 : PRIVATE cmAAItMx1, cmAAItMx2, cmAAItMx3, cmAAItMx4
00045 : PRIVATE cmAAItMx5, cmAAItMx6, cmAAItMx7, cmAAItMx8
00046 : PRIVATE cmAAItMx9, cmAAItMx10, cmAAItMx11, cmAAItMx12
00047 : PRIVATE cmAltMin1, cmAltMin2, cmAltMin3, cmAltMin4
00048 : PRIVATE cmAltMin5, cmAltMin6, cmAltMin7, cmAltMin8
```

## Programa: ESTHIDVM.PR - 2

```

00049 : PRIVATE cmAltMin9, cmAltMin10, cmAltMin11, cmAltMin12
00050 : PRIVATE cmAAltMn1, cmAAltMn2, cmAAltMn3, cmAAltMn4
00051 : PRIVATE cmAAltMn5, cmAAltMn6, cmAAltMn7, cmAAltMn8
00052 : PRIVATE cmAAltMn9, cmAAltMn10, cmAAltMn11, cmAAltMn12
00053 : PRIVATE cmCSoMed1, cmCSoMed2, cmCSoMed3, cmCSoMed4
00054 : PRIVATE cmCSoMed5, cmCSoMed6, cmCSoMed7, cmCSoMed8
00055 : PRIVATE cmCSoMed9, cmCSoMed10, cmCSoMed11, cmCSoMed12
00056 : PRIVATE cmCSoMax1, cmCSoMax2, cmCSoMax3, cmCSoMax4
00057 : PRIVATE cmCSoMax5, cmCSoMax6, cmCSoMax7, cmCSoMax8
00058 : PRIVATE cmCSoMax9, cmCSoMax10, cmCSoMax11, cmCSoMax12
00059 : PRIVATE cmACSoMx1, cmACSoMx2, cmACSoMx3, cmACSoMx4
00060 : PRIVATE cmACSoMx5, cmACSoMx6, cmACSoMx7, cmACSoMx8
00061 : PRIVATE cmACSoMx9, cmACSoMx10, cmACSoMx11, cmACSoMx12
00062 : PRIVATE cmCSoMin1, cmCSoMin2, cmCSoMin3, cmCSoMin4
00063 : PRIVATE cmCSoMin5, cmCSoMin6, cmCSoMin7, cmCSoMin8
00064 : PRIVATE cmCSoMin9, cmCSoMin10, cmCSoMin11, cmCSoMin12
00065 : PRIVATE cmACSoMn1, cmACSoMn2, cmACSoMn3, cmACSoMn4
00066 : PRIVATE cmACSoMn5, cmACSoMn6, cmACSoMn7, cmACSoMn8
00067 : PRIVATE cmACSoMn9, cmACSoMn10, cmACSoMn11, cmACSoMn12
00068 :
00069 : SET CENTURY ON
00070 :
00071 : DO Pantalla WITH "Valores Mensuales de Est.Hidrológicas", "EstHidVM"
00072 :
00073 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00074 : aEstacId    := OpenDbf( "ESTACID", aIndEstId, 1 )
00075 : aEstHVM     := OpenDbf( "ESTHIDVM", aIndEHVM, 1 )
00076 :
00077 : *SET KEY -9 TO list_val // Programa Tecla de Función
00078 :
00079 : ABM( "Registro", { || PEHVM0() }, { || PEHVM1() },;
00080 :      { || PEHVM2() }, { || PEHVM3() }, { || bAltaEHVM() },;
00081 :      { || dbDelete() }, { || bModiEHVM() } )
00082 :
00083 : SET KEY -9 TO // Programa Tecla de Función
00084 : SET CENTURY OFF
00085 : RELEASE ALL LIKE cm*
00086 :
00087 : CloseDbf( "ESTHIDVM" )
00088 : CloseDbf( "ESTACID" )
00089 : CloseDbf( "DEPTO" )
00090 :
00091 : RETURN
00092 :
00093 :
00094 : FUNCTION bAltaEHVM
00095 :
00096 : REPLACE DEPTO WITH cmDepto,;

```

```
00097 :          CODIGOEST WITH cmCodigoEst
00098 :
00099 : RETURN
00100 :
00101 :
00102 : FUNCTION PEHVM()
00103 :
00104 : @ 4, 1 SAY "Departamento:"
00105 : @ 4, 20 SAY "Estación Tipo:"
00106 : @ 4, 38 SAY "Codigo:"
00107 : @ 4, 57 SAY "Actualiz.al:"
00108 :
00109 : @ 5, 18, 20, 61 BOX "+-+|---|"
00110 : @ 5, 18, 7, 30 BOX "---|+-+|"
00111 : @ 7, 30, 20, 37 BOX "+-+|---|"
00112 : @ 5, 37, 7, 44 BOX "---|+-+|"
00113 : @ 7, 44, 20, 49 BOX "+-+|---|"
00114 : @ 5, 49, 7, 56 BOX "---|+-+|"
00115 : @ 7, 56, 20, 61 BOX "+-+|---|"
00116 :
00117 : @ 6, 19 SAY " Mes"
00118 : @ 8, 19 SAY "Enero"
00119 : @ 9, 19 SAY "Febrero"
00120 : @ 10, 19 SAY "Marzo"
00121 : @ 11, 19 SAY "Abril"
00122 : @ 12, 19 SAY "Mayo"
00123 : @ 13, 19 SAY "Junio"
00124 : @ 14, 19 SAY "Julio"
00125 : @ 15, 19 SAY "Agosto"
00126 : @ 16, 19 SAY "Setiembre"
00127 : @ 17, 19 SAY "Octubre"
00128 : @ 18, 19 SAY "Noviembre"
00129 : @ 19, 19 SAY "Diciembre"
00130 :
00131 : RETURN
00132 :
00133 :
00134 : FUNCTION PEHVM1()
00135 :
00136 : PRIVATE KN1, KN2, KN3
00137 :
00138 : KN1 := " "
00139 : KN2 := " "
00140 : KN3 := 0
00141 : iniMEHVM()
00142 :
00143 : @ 4, 16 CLEAR TO 4, 17
00144 : @ 4, 35 CLEAR TO 4, 35
```

```
00145 : @ 4, 47 CLEAR TO 4, 54
00146 : @ 4, 70 CLEAR TO 4, 79
00147 :
00148 : * 4, 15 SAY CHR(18)
00149 : @ 4, 16 GET KN1 PICTURE "99";
00150 :         VALID valexist( "DEPTO", KN1,;
00151 :         "Departamento Inexistente", NIL ) && { || showDepto( 4 ) } )
00152 : * 4, 34 SAY CHR(18)
00153 : @ 4, 35 GET KN2 PICTURE "!" VALID KN2$"AM"
00154 : * 4, 46 SAY CHR(18)
00155 : @ 4, 47 GET KN3 PICTURE "99999.99";
00156 :         VALID valexist( "ESTACID", RIGHT( "00"+ALLTRIM(KN1),2)+KN2+;
00157 :         STR( KN3, 8, 2 ),;
00158 :         "Estación Hidrológica Inexistente", NIL )
00159 :
00160 : READ
00161 : CLEAR GETS
00162 :
00163 : cmDepto      := RIGHT( "00"+ALLTRIM(KN1),2)
00164 : cmTipoEst    := KN2
00165 : cmCodigoEst := STR( KN3, 8, 2 )
00166 :
00167 : RETURN( cmDepto+cmCodigoEst )
00168 :
00169 :
00170 : FUNCTION PEHVM2()
00171 :
00172 : @ 4, 70 GET cmFechaUAct
00173 : CLEAR GETS
00174 : SeleEHVM( .F. )
00175 :
00176 : RETURN
00177 :
00178 :
00179 : FUNCTION PEHVM3()
00180 :
00181 : @ 4, 70 GET cmFechaUAct
00182 : READ
00183 : CLEAR GETS
00184 : SeleEHVM( .T. )
00185 :
00186 : RETURN
00187 :
00188 :
00189 : Func SeleEHVM( lGet )
00190 :
00191 : LOCAL c01dColor, opc
00192 :
```

```
00193 : IF FOUND()
00194 :     loadMEHVM()
00195 : ENDIF
00196 : cmAnio := YEAR( cmFechaUAct )
00197 :
00198 : cOldColor := SETCOLOR()
00199 :
00200 : opc := 1
00201 :
00202 : DO WHILE .T.
00203 :     SETCOLOR( kclrmnu3 )
00204 :     @ 21, 1 SAY "Elija:"
00205 :     SETCOLOR( kclrmnu2 )
00206 :     @ 21, 9 PROMPT "Caudal Modulo"
00207 :     *         MESSAGE "Registros de Caudal Modulo"
00208 :     @ 21, 24 PROMPT "Altura"
00209 :     *         MESSAGE "Registros de Altura"
00210 :     @ 21, 31 PROMPT "Sólido (Caudal)"
00211 :     *         MESSAGE "Registros de Caudal Sólido"
00212 :     @ 21, 48 PROMPT "Sale"
00213 :
00214 :     MENU TO opc
00215 :
00216 :     PEHVM3CLS()
00217 :
00218 :     DO CASE
00219 :         CASE opc == 1    && Caudal Modulo
00220 :             PEHVM3C()
00221 :         CASE opc == 2    && Altura
00222 :             PEHVM3A()
00223 :         CASE opc == 3    && Caudal Solido
00224 :             PEHVM3S()
00225 :         CASE opc == 4 .OR. opc == 0    && Sale
00226 :             EXIT
00227 :     ENDCASE
00228 :     IF !Get
00229 :         READ
00230 :     ENDIF
00231 :     CLEAR GETS
00232 :
00233 : ENDDO
00234 :
00235 : SETCOLOR( cOldColor )
00236 : @ 21,0 CLEAR TO 22,79
00237 : SET MESSAGE TO 21 CENTER
00238 :
00239 : RETURN
00240 :
```

```
00241 :  
00242 : FUNCTION PEHVM3CLS()  
00243 :  
00244 : @ 6, 0 CLEAR TO 6, 17  
00245 : * 8, 19 CLEAR TO 19, 29  
00246 : @ 8, 31 CLEAR TO 19, 36  
00247 : @ 8, 38 CLEAR TO 19, 43  
00248 : @ 8, 45 CLEAR TO 19, 48  
00249 : @ 8, 50 CLEAR TO 19, 55  
00250 : @ 8, 57 CLEAR TO 19, 60  
00251 :  
00252 : RETURN  
00253 :  
00254 :  
00255 : FUNCTION PEHVM3C()  
00256 :  
00257 : @ 6, 31 SAY "C.Med."  
00258 : @ 6, 38 SAY "C.Máx."  
00259 : @ 6, 45 SAY "Año "  
00260 : @ 6, 50 SAY "C.Mín."  
00261 : @ 6, 57 SAY "Año "  
00262 :  
00263 : @ 8, 31 GET cmCMoMed1 PICTURE "999.99"  
00264 : @ 8, 45 GET cmACMoMx1 PICTURE "9999" WHEN MxA(cmCMoMed1,cmCMoMax1)  
00265 : @ 8, 38 GET cmCMoMax1 PICTURE "999.99" WHEN MxV(cmCMoMed1)  
00266 : @ 8, 57 GET cmACMoMn1 PICTURE "9999" WHEN MnA(cmCMoMed1,cmCMoMin1)  
00267 : @ 8, 50 GET cmCMoMin1 PICTURE "999.99" WHEN MnV(cmCMoMed1)  
00268 :  
00269 : @ 9, 31 GET cmCMoMed2 PICTURE "999.99"  
00270 : @ 9, 45 GET cmACMoMx2 PICTURE "9999" WHEN MxA(cmCMoMed2,cmCMoMax2)  
00271 : @ 9, 38 GET cmCMoMax2 PICTURE "999.99" WHEN MxV(cmCMoMed2)  
00272 : @ 9, 57 GET cmACMoMn2 PICTURE "9999" WHEN MnA(cmCMoMed2,cmCMoMin2)  
00273 : @ 9, 50 GET cmCMoMin2 PICTURE "999.99" WHEN MnV(cmCMoMed2)  
00274 :  
00275 : @ 10, 31 GET cmCMoMed3 PICTURE "999.99"  
00276 : @ 10, 45 GET cmACMoMx3 PICTURE "9999" WHEN MxA(cmCMoMed3,cmCMoMax3)  
00277 : @ 10, 38 GET cmCMoMax3 PICTURE "999.99" WHEN MxV(cmCMoMed3)  
00278 : @ 10, 57 GET cmACMoMn3 PICTURE "9999" WHEN MnA(cmCMoMed3,cmCMoMin3)  
00279 : @ 10, 50 GET cmCMoMin3 PICTURE "999.99" WHEN MnV(cmCMoMed3)  
00280 :  
00281 : @ 11, 31 GET cmCMoMed4 PICTURE "999.99"  
00282 : @ 11, 45 GET cmACMoMx4 PICTURE "9999" WHEN MxA(cmCMoMed4,cmCMoMax4)  
00283 : @ 11, 38 GET cmCMoMax4 PICTURE "999.99" WHEN MxV(cmCMoMed4)  
00284 : @ 11, 57 GET cmACMoMn4 PICTURE "9999" WHEN MnA(cmCMoMed4,cmCMoMin4)  
00285 : @ 11, 50 GET cmCMoMin4 PICTURE "999.99" WHEN MnV(cmCMoMed4)  
00286 :  
00287 : @ 12, 31 GET cmCMoMed5 PICTURE "999.99"  
00288 : @ 12, 45 GET cmACMoMx5 PICTURE "9999" WHEN MxA(cmCMoMed5,cmCMoMax5)
```

## Programa: ESTHIDVM.PRG - 7

```

00289 : @ 12, 38 GET cmCMoMax5 PICTURE "999.99" WHEN MxV(cmCMoMed5)
00290 : @ 12, 57 GET cmACMoMn5 PICTURE "9999" WHEN MnA(cmCMoMed5,cmCMoMin5)
00291 : @ 12, 50 GET cmCMoMin5 PICTURE "999.99" WHEN MnV(cmCMoMed5)
00292 :
00293 : @ 13, 31 GET cmCMoMed6 PICTURE "999.99"
00294 : @ 13, 45 GET cmACMoMx6 PICTURE "9999" WHEN MxA(cmCMoMed6,cmCMoMax6)
00295 : @ 13, 38 GET cmCMoMax6 PICTURE "999.99" WHEN MxV(cmCMoMed6)
00296 : @ 13, 57 GET cmACMoMn6 PICTURE "9999" WHEN MnA(cmCMoMed6,cmCMoMin6)
00297 : @ 13, 50 GET cmCMoMin6 PICTURE "999.99" WHEN MnV(cmCMoMed6)
00298 :
00299 : @ 14, 31 GET cmCMoMed7 PICTURE "999.99"
00300 : @ 14, 45 GET cmACMoMx7 PICTURE "9999" WHEN MxA(cmCMoMed7,cmCMoMax7)
00301 : @ 14, 38 GET cmCMoMax7 PICTURE "999.99" WHEN MxV(cmCMoMed7)
00302 : @ 14, 57 GET cmACMoMn7 PICTURE "9999" WHEN MnA(cmCMoMed7,cmCMoMin7)
00303 : @ 14, 50 GET cmCMoMin7 PICTURE "999.99" WHEN MnV(cmCMoMed7)
00304 :
00305 : @ 15, 31 GET cmCMoMed8 PICTURE "999.99"
00306 : @ 15, 45 GET cmACMoMx8 PICTURE "9999" WHEN MxA(cmCMoMed8,cmCMoMax8)
00307 : @ 15, 38 GET cmCMoMax8 PICTURE "999.99" WHEN MxV(cmCMoMed8)
00308 : @ 15, 57 GET cmACMoMn8 PICTURE "9999" WHEN MnA(cmCMoMed8,cmCMoMin8)
00309 : @ 15, 50 GET cmCMoMin8 PICTURE "999.99" WHEN MnV(cmCMoMed8)
00310 :
00311 : @ 16, 31 GET cmCMoMed9 PICTURE "999.99"
00312 : @ 16, 45 GET cmACMoMx9 PICTURE "9999" WHEN MxA(cmCMoMed9,cmCMoMax9)
00313 : @ 16, 38 GET cmCMoMax9 PICTURE "999.99" WHEN MxV(cmCMoMed9)
00314 : @ 16, 57 GET cmACMoMn9 PICTURE "9999" WHEN MnA(cmCMoMed9,cmCMoMin9)
00315 : @ 16, 50 GET cmCMoMin9 PICTURE "999.99" WHEN MnV(cmCMoMed9)
00316 :
00317 : @ 17, 31 GET cmCMoMed10 PICTURE "999.99"
00318 : @ 17, 45 GET cmACMoMx10 PICTURE "9999" WHEN MxA(cmCMoMed10,cmCMoMax10)
00319 : @ 17, 38 GET cmCMoMax10 PICTURE "999.99" WHEN MxV(cmCMoMed10)
00320 : @ 17, 57 GET cmACMoMn10 PICTURE "9999" WHEN MnA(cmCMoMed10,cmCMoMin10)
00321 : @ 17, 50 GET cmCMoMin10 PICTURE "999.99" WHEN MnV(cmCMoMed10)
00322 :
00323 : @ 18, 31 GET cmCMoMed11 PICTURE "999.99"
00324 : @ 18, 45 GET cmACMoMx11 PICTURE "9999" WHEN MxA(cmCMoMed11,cmCMoMax11)
00325 : @ 18, 38 GET cmCMoMax11 PICTURE "999.99" WHEN MxV(cmCMoMed11)
00326 : @ 18, 57 GET cmACMoMn11 PICTURE "9999" WHEN MnA(cmCMoMed11,cmCMoMin11)
00327 : @ 18, 50 GET cmCMoMin11 PICTURE "999.99" WHEN MnV(cmCMoMed11)
00328 :
00329 : @ 19, 31 GET cmCMoMed12 PICTURE "999.99"
00330 : @ 19, 45 GET cmACMoMx12 PICTURE "9999" WHEN MxA(cmCMoMed12,cmCMoMax12)
00331 : @ 19, 38 GET cmCMoMax12 PICTURE "999.99" WHEN MxV(cmCMoMed12)
00332 : @ 19, 57 GET cmACMoMn12 PICTURE "9999" WHEN MnA(cmCMoMed12,cmCMoMin12)
00333 : @ 19, 50 GET cmCMoMin12 PICTURE "999.99" WHEN MnV(cmCMoMed12)
00334 :
00335 : RETURN
00336 :

```

```
00337 :
00338 : FUNCTION PEHVM3A()
00339 :
00340 : @ 6, 31 SAY "A.Med."
00341 : @ 6, 38 SAY "A.Máx."
00342 : @ 6, 45 SAY "Año "
00343 : @ 6, 50 SAY "A.Mín."
00344 : @ 6, 57 SAY "Año "
00345 :
00346 : @ 6, 0 SAY "Elem.Med.(E/L):"
00347 : @ 6, 16 GET cmAltElMed PICTURE "!" VALID cmAltElMed$"EL"
00348 :
00349 : @ 8, 31 GET cmAltMed1 PICTURE "999.99"
00350 : @ 8, 45 GET cmAAItMx1 PICTURE "9999" WHEN MxA(cmAltMed1,cmAltMax1)
00351 : @ 8, 38 GET cmAltMax1 PICTURE "999.99" WHEN MxV(cmAltMed1)
00352 : @ 8, 57 GET cmAAItMn1 PICTURE "9999" WHEN MnA(cmAltMed1,cmAltMin1)
00353 : @ 8, 50 GET cmAltMin1 PICTURE "999.99" WHEN MnV(cmAltMed1)
00354 :
00355 : @ 9, 31 GET cmAltMed2 PICTURE "999.99"
00356 : @ 9, 45 GET cmAAItMx2 PICTURE "9999" WHEN MxA(cmAltMed2,cmAltMax2)
00357 : @ 9, 38 GET cmAltMax2 PICTURE "999.99" WHEN MxV(cmAltMed2)
00358 : @ 9, 57 GET cmAAItMn2 PICTURE "9999" WHEN MnA(cmAltMed2,cmAltMin2)
00359 : @ 9, 50 GET cmAltMin2 PICTURE "999.99" WHEN MnV(cmAltMed2)
00360 :
00361 : @ 10, 31 GET cmAltMed3 PICTURE "999.99"
00362 : @ 10, 45 GET cmAAItMx3 PICTURE "9999" WHEN MxA(cmAltMed3,cmAltMax3)
00363 : @ 10, 38 GET cmAltMax3 PICTURE "999.99" WHEN MxV(cmAltMed3)
00364 : @ 10, 57 GET cmAAItMn3 PICTURE "9999" WHEN MnA(cmAltMed3,cmAltMin3)
00365 : @ 10, 50 GET cmAltMin3 PICTURE "999.99" WHEN MnV(cmAltMed3)
00366 :
00367 : @ 11, 31 GET cmAltMed4 PICTURE "999.99"
00368 : @ 11, 45 GET cmAAItMx4 PICTURE "9999" WHEN MxA(cmAltMed4,cmAltMax4)
00369 : @ 11, 38 GET cmAltMax4 PICTURE "999.99" WHEN MxV(cmAltMed4)
00370 : @ 11, 57 GET cmAAItMn4 PICTURE "9999" WHEN MnA(cmAltMed4,cmAltMin4)
00371 : @ 11, 50 GET cmAltMin4 PICTURE "999.99" WHEN MnV(cmAltMed4)
00372 :
00373 : @ 12, 31 GET cmAltMed5 PICTURE "999.99"
00374 : @ 12, 45 GET cmAAItMx5 PICTURE "9999" WHEN MxA(cmAltMed5,cmAltMax5)
00375 : @ 12, 38 GET cmAltMax5 PICTURE "999.99" WHEN MxV(cmAltMed5)
00376 : @ 12, 57 GET cmAAItMn5 PICTURE "9999" WHEN MnA(cmAltMed5,cmAltMin5)
00377 : @ 12, 50 GET cmAltMin5 PICTURE "999.99" WHEN MnV(cmAltMed5)
00378 :
00379 : @ 13, 31 GET cmAltMed6 PICTURE "999.99"
00380 : @ 13, 45 GET cmAAItMx6 PICTURE "9999" WHEN MxA(cmAltMed6,cmAltMax6)
00381 : @ 13, 38 GET cmAltMax6 PICTURE "999.99" WHEN MxV(cmAltMed6)
00382 : @ 13, 57 GET cmAAItMn6 PICTURE "9999" WHEN MnA(cmAltMed6,cmAltMin6)
00383 : @ 13, 50 GET cmAltMin6 PICTURE "999.99" WHEN MnV(cmAltMed6)
00384 :
```



```
00385 : @ 14, 31 GET cmAltMed7 PICTURE "999.99"
00386 : @ 14, 45 GET cmAAItMx7 PICTURE "9999" WHEN MxA(cmAltMed7,cmAltMax7)
00387 : @ 14, 38 GET cmAltMax7 PICTURE "999.99" WHEN MxV(cmAltMed7)
00388 : @ 14, 57 GET cmAAItMn7 PICTURE "9999" WHEN MnA(cmAltMed7,cmAltMin7)
00389 : @ 14, 50 GET cmAltMin7 PICTURE "999.99" WHEN MnV(cmAltMed7)
00390 :
00391 : @ 15, 31 GET cmAltMed8 PICTURE "999.99"
00392 : @ 15, 45 GET cmAAItMx8 PICTURE "9999" WHEN MxA(cmAltMed8,cmAltMax8)
00393 : @ 15, 38 GET cmAltMax8 PICTURE "999.99" WHEN MxV(cmAltMed8)
00394 : @ 15, 57 GET cmAAItMn8 PICTURE "9999" WHEN MnA(cmAltMed8,cmAltMin8)
00395 : @ 15, 50 GET cmAltMin8 PICTURE "999.99" WHEN MnV(cmAltMed8)
00396 :
00397 : @ 16, 31 GET cmAltMed9 PICTURE "999.99"
00398 : @ 16, 45 GET cmAAItMx9 PICTURE "9999" WHEN MxA(cmAltMed9,cmAltMax9)
00399 : @ 16, 38 GET cmAltMax9 PICTURE "999.99" WHEN MxV(cmAltMed9)
00400 : @ 16, 57 GET cmAAItMn9 PICTURE "9999" WHEN MnA(cmAltMed9,cmAltMin9)
00401 : @ 16, 50 GET cmAltMin9 PICTURE "999.99" WHEN MnV(cmAltMed9)
00402 :
00403 : @ 17, 31 GET cmAltMed10 PICTURE "999.99"
00404 : @ 17, 45 GET cmAAItMx10 PICTURE "9999" WHEN MxA(cmAltMed10,cmAltMax10)
00405 : @ 17, 38 GET cmAltMax10 PICTURE "999.99" WHEN MxV(cmAltMed10)
00406 : @ 17, 57 GET cmAAItMn10 PICTURE "9999" WHEN MnA(cmAltMed10,cmAltMin10)
00407 : @ 17, 50 GET cmAltMin10 PICTURE "999.99" WHEN MnV(cmAltMed10)
00408 :
00409 : @ 18, 31 GET cmAltMed11 PICTURE "999.99"
00410 : @ 18, 45 GET cmAAItMx11 PICTURE "9999" WHEN MxA(cmAltMed11,cmAltMax11)
00411 : @ 18, 38 GET cmAltMax11 PICTURE "999.99" WHEN MxV(cmAltMed11)
00412 : @ 18, 57 GET cmAAItMn11 PICTURE "9999" WHEN MnA(cmAltMed11,cmAltMin11)
00413 : @ 18, 50 GET cmAltMin11 PICTURE "999.99" WHEN MnV(cmAltMed11)
00414 :
00415 : @ 19, 31 GET cmAltMed12 PICTURE "999.99"
00416 : @ 19, 45 GET cmAAItMx12 PICTURE "9999" WHEN MxA(cmAltMed12,cmAltMax12)
00417 : @ 19, 38 GET cmAltMax12 PICTURE "999.99" WHEN MxV(cmAltMed12)
00418 : @ 19, 57 GET cmAAItMn12 PICTURE "9999" WHEN MnA(cmAltMed12,cmAltMin12)
00419 : @ 19, 50 GET cmAltMin12 PICTURE "999.99" WHEN MnV(cmAltMed12)
00420 :
00421 : RETURN
00422 :
00423 :
00424 : FUNCTION PEHVM3S()
00425 :
00426 : @ 6, 31 SAY "S.Med."
00427 : @ 6, 38 SAY "S.Máy."
00428 : @ 6, 45 SAY "Año "
00429 : @ 6, 50 SAY "S.Mín."
00430 : @ 6, 57 SAY "Año "
00431 :
00432 : @ 8, 31 GET cmCSoMed1 PICTURE "999.99"
```

```

00433 : @ 8, 45 GET cmACSoMx1 PICTURE "9999" WHEN MxA(cmCSoMed1,cmCSoMax1)
00434 : @ 8, 38 GET cmCSoMax1 PICTURE "999.99" WHEN MxV(cmCSoMed1)
00435 : @ 8, 57 GET cmACSoMn1 PICTURE "9999" WHEN MnA(cmCSoMed1,cmCSoMin1)
00436 : @ 8, 50 GET cmCSoMin1 PICTURE "999.99" WHEN MnV(cmCSoMed1)
00437 :
00438 : @ 9, 31 GET cmCSoMed2 PICTURE "999.99"
00439 : @ 9, 45 GET cmACSoMx2 PICTURE "9999" WHEN MxA(cmCSoMed2,cmCSoMax2)
00440 : @ 9, 38 GET cmCSoMax2 PICTURE "999.99" WHEN MxV(cmCSoMed2)
00441 : @ 9, 57 GET cmACSoMn2 PICTURE "9999" WHEN MnA(cmCSoMed2,cmCSoMin2)
00442 : @ 9, 50 GET cmCSoMin2 PICTURE "999.99" WHEN MnV(cmCSoMed2)
00443 :
00444 : @ 10, 31 GET cmCSoMed3 PICTURE "999.99"
00445 : @ 10, 45 GET cmACSoMx3 PICTURE "9999" WHEN MxA(cmCSoMed3,cmCSoMax3)
00446 : @ 10, 38 GET cmCSoMax3 PICTURE "999.99" WHEN MxV(cmCSoMed3)
00447 : @ 10, 57 GET cmACSoMn3 PICTURE "9999" WHEN MnA(cmCSoMed3,cmCSoMin3)
00448 : @ 10, 50 GET cmCSoMin3 PICTURE "999.99" WHEN MnV(cmCSoMed3)
00449 :
00450 : @ 11, 31 GET cmCSoMed4 PICTURE "999.99"
00451 : @ 11, 45 GET cmACSoMx4 PICTURE "9999" WHEN MxA(cmCSoMed4,cmCSoMax4)
00452 : @ 11, 38 GET cmCSoMax4 PICTURE "999.99" WHEN MxV(cmCSoMed4)
00453 : @ 11, 57 GET cmACSoMn4 PICTURE "9999" WHEN MnA(cmCSoMed4,cmCSoMin4)
00454 : @ 11, 50 GET cmCSoMin4 PICTURE "999.99" WHEN MnV(cmCSoMed4)
00455 :
00456 : @ 12, 31 GET cmCSoMed5 PICTURE "999.99"
00457 : @ 12, 45 GET cmACSoMx5 PICTURE "9999" WHEN MxA(cmCSoMed5,cmCSoMax5)
00458 : @ 12, 38 GET cmCSoMax5 PICTURE "999.99" WHEN MxV(cmCSoMed5)
00459 : @ 12, 57 GET cmACSoMn5 PICTURE "9999" WHEN MnA(cmCSoMed5,cmCSoMin5)
00460 : @ 12, 50 GET cmCSoMin5 PICTURE "999.99" WHEN MnV(cmCSoMed5)
00461 :
00462 : @ 13, 31 GET cmCSoMed6 PICTURE "999.99"
00463 : @ 13, 45 GET cmACSoMx6 PICTURE "9999" WHEN MxA(cmCSoMed6,cmCSoMax6)
00464 : @ 13, 38 GET cmCSoMax6 PICTURE "999.99" WHEN MxV(cmCSoMed6)
00465 : @ 13, 57 GET cmACSoMn6 PICTURE "9999" WHEN MnA(cmCSoMed6,cmCSoMin6)
00466 : @ 13, 50 GET cmCSoMin6 PICTURE "999.99" WHEN MnV(cmCSoMed6)
00467 :
00468 : @ 14, 31 GET cmCSoMed7 PICTURE "999.99"
00469 : @ 14, 45 GET cmACSoMx7 PICTURE "9999" WHEN MxA(cmCSoMed7,cmCSoMax7)
00470 : @ 14, 38 GET cmCSoMax7 PICTURE "999.99" WHEN MxV(cmCSoMed7)
00471 : @ 14, 57 GET cmACSoMn7 PICTURE "9999" WHEN MnA(cmCSoMed7,cmCSoMin7)
00472 : @ 14, 50 GET cmCSoMin7 PICTURE "999.99" WHEN MnV(cmCSoMed7)
00473 :
00474 : @ 15, 31 GET cmCSoMed8 PICTURE "999.99"
00475 : @ 15, 45 GET cmACSoMx8 PICTURE "9999" WHEN MxA(cmCSoMed8,cmCSoMax8)
00476 : @ 15, 38 GET cmCSoMax8 PICTURE "999.99" WHEN MxV(cmCSoMed8)
00477 : @ 15, 57 GET cmACSoMn8 PICTURE "9999" WHEN MnA(cmCSoMed8,cmCSoMin8)
00478 : @ 15, 50 GET cmCSoMin8 PICTURE "999.99" WHEN MnV(cmCSoMed8)
00479 :
00480 : @ 16, 31 GET cmCSoMed9 PICTURE "999.99"

```

## Programa: ESTHIDVM.PRG - 11

```
00481 : @ 16, 45 GET cmACSoMx9  PICTURE "9999"  WHEN MxA(cmCSoMed9,cmCSoMax9)
00482 : @ 16, 38 GET cmCSoMax9  PICTURE "999.99"  WHEN MxV(cmCSoMed9)
00483 : @ 16, 57 GET cmACSoMn9  PICTURE "9999"  WHEN MnA(cmCSoMed9,cmCSoMin9)
00484 : @ 16, 50 GET cmCSoMin9  PICTURE "999.99"  WHEN MnV(cmCSoMed9)
00485 :
00486 : @ 17, 31 GET cmCSoMed10 PICTURE "999.99"
00487 : @ 17, 45 GET cmACSoMx10 PICTURE "9999"  WHEN MxA(cmCSoMed10,cmCSoMax10)
00488 : @ 17, 38 GET cmCSoMax10 PICTURE "999.99"  WHEN MxV(cmCSoMed10)
00489 : @ 17, 57 GET cmACSoMn10 PICTURE "9999"  WHEN MnA(cmCSoMed10,cmCSoMin10)
00490 : @ 17, 50 GET cmCSoMin10 PICTURE "999.99"  WHEN MnV(cmCSoMed10)
00491 :
00492 : @ 18, 31 GET cmCSoMed11 PICTURE "999.99"
00493 : @ 18, 45 GET cmACSoMx11 PICTURE "9999"  WHEN MxA(cmCSoMed11,cmCSoMax11)
00494 : @ 18, 38 GET cmCSoMax11 PICTURE "999.99"  WHEN MxV(cmCSoMed11)
00495 : @ 18, 57 GET cmACSoMn11 PICTURE "9999"  WHEN MnA(cmCSoMed11,cmCSoMin11)
00496 : @ 18, 50 GET cmCSoMin11 PICTURE "999.99"  WHEN MnV(cmCSoMed11)
00497 :
00498 : @ 19, 31 GET cmCSoMed12 PICTURE "999.99"
00499 : @ 19, 45 GET cmACSoMx12 PICTURE "9999"  WHEN MxA(cmCSoMed12,cmCSoMax12)
00500 : @ 19, 38 GET cmCSoMax12 PICTURE "999.99"  WHEN MxV(cmCSoMed12)
00501 : @ 19, 57 GET cmACSoMn12 PICTURE "9999"  WHEN MnA(cmCSoMed12,cmCSoMin12)
00502 : @ 19, 50 GET cmCSoMin12 PICTURE "999.99"  WHEN MnV(cmCSoMed12)
00503 :
00504 : RETURN
00505 :
00506 :
00507 : Func bModiEHVM()
00508 :
00509 : REPLACE CMOMED1  WITH cmCMoMed1,;
00510 :          CMOMED2  WITH cmCMoMed2,;
00511 :          CMOMED3  WITH cmCMoMed3,;
00512 :          CMOMED4  WITH cmCMoMed4,;
00513 :          CMOMED5  WITH cmCMoMed5,;
00514 :          CMOMED6  WITH cmCMoMed6,;
00515 :          CMOMED7  WITH cmCMoMed7,;
00516 :          CMOMED8  WITH cmCMoMed8,;
00517 :          CMOMED9  WITH cmCMoMed9,;
00518 :          CMOMED10 WITH cmCMoMed10,;
00519 :          CMOMED11 WITH cmCMoMed11,;
00520 :          CMOMED12 WITH cmCMoMed12
00521 :
00522 : REPLACE CMOMAX1  WITH cmCMoMax1,;
00523 :          CMOMAX2  WITH cmCMoMax2,;
00524 :          CMOMAX3  WITH cmCMoMax3,;
00525 :          CMOMAX4  WITH cmCMoMax4,;
00526 :          CMOMAX5  WITH cmCMoMax5,;
00527 :          CMOMAX6  WITH cmCMoMax6,;
00528 :          CMOMAX7  WITH cmCMoMax7,;
```

```
00529 :          CMOMAX8   WITH cmCMoMax8,;
00530 :          CMOMAX9   WITH cmCMoMax9,;
00531 :          CMOMAX10  WITH cmCMoMax10,;
00532 :          CMOMAX11  WITH cmCMoMax11,;
00533 :          CMOMAX12  WITH cmCMoMax12
00534 :
00535 : REPLACE ACMOMX1   WITH cmACMoMx1,;
00536 :          ACMOMX2   WITH cmACMoMx2,;
00537 :          ACMOMX3   WITH cmACMoMx3,;
00538 :          ACMOMX4   WITH cmACMoMx4,;
00539 :          ACMOMX5   WITH cmACMoMx5,;
00540 :          ACMOMX6   WITH cmACMoMx6,;
00541 :          ACMOMX7   WITH cmACMoMx7,;
00542 :          ACMOMX8   WITH cmACMoMx8,;
00543 :          ACMOMX9   WITH cmACMoMx9,;
00544 :          ACMOMX10  WITH cmACMoMx10,;
00545 :          ACMOMX11  WITH cmACMoMx11,;
00546 :          ACMOMX12  WITH cmACMoMx12
00547 :
00548 : REPLACE CMOMIN1   WITH cmCMoMin1,;
00549 :          CMOMIN2   WITH cmCMoMin2,;
00550 :          CMOMIN3   WITH cmCMoMin3,;
00551 :          CMOMIN4   WITH cmCMoMin4,;
00552 :          CMOMIN5   WITH cmCMoMin5,;
00553 :          CMOMIN6   WITH cmCMoMin6,;
00554 :          CMOMIN7   WITH cmCMoMin7,;
00555 :          CMOMIN8   WITH cmCMoMin8,;
00556 :          CMOMIN9   WITH cmCMoMin9,;
00557 :          CMOMIN10  WITH cmCMoMin10,;
00558 :          CMOMIN11  WITH cmCMoMin11,;
00559 :          CMOMIN12  WITH cmCMoMin12
00560 :
00561 : REPLACE ACMOMN1   WITH cmACMoMn1,;
00562 :          ACMOMN2   WITH cmACMoMn2,;
00563 :          ACMOMN3   WITH cmACMoMn3,;
00564 :          ACMOMN4   WITH cmACMoMn4,;
00565 :          ACMOMN5   WITH cmACMoMn5,;
00566 :          ACMOMN6   WITH cmACMoMn6,;
00567 :          ACMOMN7   WITH cmACMoMn7,;
00568 :          ACMOMN8   WITH cmACMoMn8,;
00569 :          ACMOMN9   WITH cmACMoMn9,;
00570 :          ACMOMN10  WITH cmACMoMn10,;
00571 :          ACMOMN11  WITH cmACMoMn11,;
00572 :          ACMOMN12  WITH cmACMoMn12
00573 :
00574 : REPLACE ALTMED1   WITH cmAltMed1,;
00575 :          ALTMED2   WITH cmAltMed2,;
00576 :          ALTMED3   WITH cmAltMed3,;
```

```
00577 :      ALTMED4   WITH cmAltMed4,;
00578 :      ALTMED5   WITH cmAltMed5,;
00579 :      ALTMED6   WITH cmAltMed6,;
00580 :      ALTMED7   WITH cmAltMed7,;
00581 :      ALTMED8   WITH cmAltMed8,;
00582 :      ALTMED9   WITH cmAltMed9,;
00583 :      ALTMED10  WITH cmAltMed10,;
00584 :      ALTMED11  WITH cmAltMed11,;
00585 :      ALTMED12  WITH cmAltMed12
00586 :
00587 : REPLACE ALTMAX1   WITH cmAltMax1,;
00588 :      ALTMAX2   WITH cmAltMax2,;
00589 :      ALTMAX3   WITH cmAltMax3,;
00590 :      ALTMAX4   WITH cmAltMax4,;
00591 :      ALTMAX5   WITH cmAltMax5,;
00592 :      ALTMAX6   WITH cmAltMax6,;
00593 :      ALTMAX7   WITH cmAltMax7,;
00594 :      ALTMAX8   WITH cmAltMax8,;
00595 :      ALTMAX9   WITH cmAltMax9,;
00596 :      ALTMAX10  WITH cmAltMax10,;
00597 :      ALTMAX11  WITH cmAltMax11,;
00598 :      ALTMAX12  WITH cmAltMax12
00599 :
00600 : REPLACE AALTMX1   WITH cmAAltMx1,;
00601 :      AALTMX2   WITH cmAAltMx2,;
00602 :      AALTMX3   WITH cmAAltMx3,;
00603 :      AALTMX4   WITH cmAAltMx4,;
00604 :      AALTMX5   WITH cmAAltMx5,;
00605 :      AALTMX6   WITH cmAAltMx6,;
00606 :      AALTMX7   WITH cmAAltMx7,;
00607 :      AALTMX8   WITH cmAAltMx8,;
00608 :      AALTMX9   WITH cmAAltMx9,;
00609 :      AALTMX10  WITH cmAAltMx10,;
00610 :      AALTMX11  WITH cmAAltMx11,;
00611 :      AALTMX12  WITH cmAAltMx12
00612 :
00613 : REPLACE ALTMIN1   WITH cmAltMin1,;
00614 :      ALTMIN2   WITH cmAltMin2,;
00615 :      ALTMIN3   WITH cmAltMin3,;
00616 :      ALTMIN4   WITH cmAltMin4,;
00617 :      ALTMIN5   WITH cmAltMin5,;
00618 :      ALTMIN6   WITH cmAltMin6,;
00619 :      ALTMIN7   WITH cmAltMin7,;
00620 :      ALTMIN8   WITH cmAltMin8,;
00621 :      ALTMIN9   WITH cmAltMin9,;
00622 :      ALTMIN10  WITH cmAltMin10,;
00623 :      ALTMIN11  WITH cmAltMin11,;
00624 :      ALTMIN12  WITH cmAltMin12
```

```
00625 :  
00626 : REPLACE AALTMN1 WITH cmAAItMn1,;  
00627 : AALTMN2 WITH cmAAItMn2,;  
00628 : AALTMN3 WITH cmAAItMn3,;  
00629 : AALTMN4 WITH cmAAItMn4,;  
00630 : AALTMN5 WITH cmAAItMn5,;  
00631 : AALTMN6 WITH cmAAItMn6,;  
00632 : AALTMN7 WITH cmAAItMn7,;  
00633 : AALTMN8 WITH cmAAItMn8,;  
00634 : AALTMN9 WITH cmAAItMn9,;  
00635 : AALTMN10 WITH cmAAItMn10,;  
00636 : AALTMN11 WITH cmAAItMn11,;  
00637 : AALTMN12 WITH cmAAItMn12  
00638 :  
00639 : REPLACE CSOMED1 WITH cmCSoMed1,;  
00640 : CSOMED2 WITH cmCSoMed2,;  
00641 : CSOMED3 WITH cmCSoMed3,;  
00642 : CSOMED4 WITH cmCSoMed4,;  
00643 : CSOMED5 WITH cmCSoMed5,;  
00644 : CSOMED6 WITH cmCSoMed6,;  
00645 : CSOMED7 WITH cmCSoMed7,;  
00646 : CSOMED8 WITH cmCSoMed8,;  
00647 : CSOMED9 WITH cmCSoMed9,;  
00648 : CSOMED10 WITH cmCSoMed10,;  
00649 : CSOMED11 WITH cmCSoMed11,;  
00650 : CSOMED12 WITH cmCSoMed12  
00651 :  
00652 : REPLACE CSOMAX1 WITH cmCSoMax1,;  
00653 : CSOMAX2 WITH cmCSoMax2,;  
00654 : CSOMAX3 WITH cmCSoMax3,;  
00655 : CSOMAX4 WITH cmCSoMax4,;  
00656 : CSOMAX5 WITH cmCSoMax5,;  
00657 : CSOMAX6 WITH cmCSoMax6,;  
00658 : CSOMAX7 WITH cmCSoMax7,;  
00659 : CSOMAX8 WITH cmCSoMax8,;  
00660 : CSOMAX9 WITH cmCSoMax9,;  
00661 : CSOMAX10 WITH cmCSoMax10,;  
00662 : CSOMAX11 WITH cmCSoMax11,;  
00663 : CSOMAX12 WITH cmCSoMax12  
00664 :  
00665 : REPLACE ACSOMX1 WITH cmACSoMx1,;  
00666 : ACSOMX2 WITH cmACSoMx2,;  
00667 : ACSOMX3 WITH cmACSoMx3,;  
00668 : ACSOMX4 WITH cmACSoMx4,;  
00669 : ACSOMX5 WITH cmACSoMx5,;  
00670 : ACSOMX6 WITH cmACSoMx6,;  
00671 : ACSOMX7 WITH cmACSoMx7,;  
00672 : ACSOMX8 WITH cmACSoMx8,;
```

```
00673 :      ACSOMX9   WITH cmACSoMx9,;
00674 :      ACSOMX10  WITH cmACSoMx10,;
00675 :      ACSOMX11   WITH cmACSoMx11,;
00676 :      ACSOMX12   WITH cmACSoMx12
00677 :
00678 : REPLACE CSOMIN1   WITH cmCSoMin1,;
00679 :      CSOMIN2   WITH cmCSoMin2,;
00680 :      CSOMIN3   WITH cmCSoMin3,;
00681 :      CSOMIN4   WITH cmCSoMin4,;
00682 :      CSOMIN5   WITH cmCSoMin5,;
00683 :      CSOMIN6   WITH cmCSoMin6,;
00684 :      CSOMIN7   WITH cmCSoMin7,;
00685 :      CSOMIN8   WITH cmCSoMin8,;
00686 :      CSOMIN9   WITH cmCSoMin9,;
00687 :      CSOMIN10  WITH cmCSoMin10,;
00688 :      CSOMIN11  WITH cmCSoMin11,;
00689 :      CSOMIN12  WITH cmCSoMin12
00690 :
00691 : REPLACE ACSOMN1   WITH cmACSoMn1,;
00692 :      ACSOMN2   WITH cmACSoMn2,;
00693 :      ACSOMN3   WITH cmACSoMn3,;
00694 :      ACSOMN4   WITH cmACSoMn4,;
00695 :      ACSOMN5   WITH cmACSoMn5,;
00696 :      ACSOMN6   WITH cmACSoMn6,;
00697 :      ACSOMN7   WITH cmACSoMn7,;
00698 :      ACSOMN8   WITH cmACSoMn8,;
00699 :      ACSOMN9   WITH cmACSoMn9,;
00700 :      ACSOMN10  WITH cmACSoMn10,;
00701 :      ACSOMN11  WITH cmACSoMn11,;
00702 :      ACSOMN12  WITH cmACSoMn12
00703 :
00704 : REPLACE FECHAUACT WITH cmFechaUAct,;
00705 :      ALTELMED  WITH cmAltElMed
00706 :
00707 : Return
00708 :
00709 :
00710 : FUNCTION iniMEHVM()
00711 :
00712 : cmCMoMed1   := 0
00713 : cmCMoMed2   := 0
00714 : cmCMoMed3   := 0
00715 : cmCMoMed4   := 0
00716 : cmCMoMed5   := 0
00717 : cmCMoMed6   := 0
00718 : cmCMoMed7   := 0
00719 : cmCMoMed8   := 0
00720 : cmCMoMed9   := 0
```

```
00721 : cmCMoMed10 := 0
00722 : cmCMoMed11 := 0
00723 : cmCMoMed12 := 0
00724 :
00725 : cmCMoMax1 := 0
00726 : cmCMoMax2 := 0
00727 : cmCMoMax3 := 0
00728 : cmCMoMax4 := 0
00729 : cmCMoMax5 := 0
00730 : cmCMoMax6 := 0
00731 : cmCMoMax7 := 0
00732 : cmCMoMax8 := 0
00733 : cmCMoMax9 := 0
00734 : cmCMoMax10 := 0
00735 : cmCMoMax11 := 0
00736 : cmCMoMax12 := 0
00737 :
00738 : cmACMoMx1 := 0
00739 : cmACMoMx2 := 0
00740 : cmACMoMx3 := 0
00741 : cmACMoMx4 := 0
00742 : cmACMoMx5 := 0
00743 : cmACMoMx6 := 0
00744 : cmACMoMx7 := 0
00745 : cmACMoMx8 := 0
00746 : cmACMoMx9 := 0
00747 : cmACMoMx10 := 0
00748 : cmACMoMx11 := 0
00749 : cmACMoMx12 := 0
00750 :
00751 : cmCMoMin1 := 999.99
00752 : cmCMoMin2 := 999.99
00753 : cmCMoMin3 := 999.99
00754 : cmCMoMin4 := 999.99
00755 : cmCMoMin5 := 999.99
00756 : cmCMoMin6 := 999.99
00757 : cmCMoMin7 := 999.99
00758 : cmCMoMin8 := 999.99
00759 : cmCMoMin9 := 999.99
00760 : cmCMoMin10 := 999.99
00761 : cmCMoMin11 := 999.99
00762 : cmCMoMin12 := 999.99
00763 :
00764 : cmACMoMn1 := 0
00765 : cmACMoMn2 := 0
00766 : cmACMoMn3 := 0
00767 : cmACMoMn4 := 0
00768 : cmACMoMn5 := 0
```



## Programa: ESTHIDVM.PRG - 17

```
00769 : cmACMoMn6      := 0
00770 : cmACMoMn7      := 0
00771 : cmACMoMn8      := 0
00772 : cmACMoMn9      := 0
00773 : cmACMoMn10     := 0
00774 : cmACMoMn11     := 0
00775 : cmACMoMn12     := 0
00776 :
00777 : cmAltMed1       := 0
00778 : cmAltMed2       := 0
00779 : cmAltMed3       := 0
00780 : cmAltMed4       := 0
00781 : cmAltMed5       := 0
00782 : cmAltMed6       := 0
00783 : cmAltMed7       := 0
00784 : cmAltMed8       := 0
00785 : cmAltMed9       := 0
00786 : cmAltMed10      := 0
00787 : cmAltMed11      := 0
00788 : cmAltMed12      := 0
00789 :
00790 : cmAltMax1        := 0
00791 : cmAltMax2        := 0
00792 : cmAltMax3        := 0
00793 : cmAltMax4        := 0
00794 : cmAltMax5        := 0
00795 : cmAltMax6        := 0
00796 : cmAltMax7        := 0
00797 : cmAltMax8        := 0
00798 : cmAltMax9        := 0
00799 : cmAltMax10       := 0
00800 : cmAltMax11       := 0
00801 : cmAltMax12       := 0
00802 :
00803 : cmAA1tMx1        := 0
00804 : cmAA1tMx2        := 0
00805 : cmAA1tMx3        := 0
00806 : cmAA1tMx4        := 0
00807 : cmAA1tMx5        := 0
00808 : cmAA1tMx6        := 0
00809 : cmAA1tMx7        := 0
00810 : cmAA1tMx8        := 0
00811 : cmAA1tMx9        := 0
00812 : cmAA1tMx10       := 0
00813 : cmAA1tMx11       := 0
00814 : cmAA1tMx12       := 0
00815 :
00816 : cmAltMin1         := 999.99
```

## Programa: ESTHIDVM.PRG - 18

```
00817 : cmAltMin2      := 999.99
00818 : cmAltMin3      := 999.99
00819 : cmAltMin4      := 999.99
00820 : cmAltMin5      := 999.99
00821 : cmAltMin6      := 999.99
00822 : cmAltMin7      := 999.99
00823 : cmAltMin8      := 999.99
00824 : cmAltMin9      := 999.99
00825 : cmAltMin10     := 999.99
00826 : cmAltMin11     := 999.99
00827 : cmAltMin12     := 999.99
00828 :
00829 : cmAAltMn1      := 0
00830 : cmAAltMn2      := 0
00831 : cmAAltMn3      := 0
00832 : cmAAltMn4      := 0
00833 : cmAAltMn5      := 0
00834 : cmAAltMn6      := 0
00835 : cmAAltMn7      := 0
00836 : cmAAltMn8      := 0
00837 : cmAAltMn9      := 0
00838 : cmAAltMn10     := 0
00839 : cmAAltMn11     := 0
00840 : cmAAltMn12     := 0
00841 :
00842 : cmCSoMed1      := 0
00843 : cmCSoMed2      := 0
00844 : cmCSoMed3      := 0
00845 : cmCSoMed4      := 0
00846 : cmCSoMed5      := 0
00847 : cmCSoMed6      := 0
00848 : cmCSoMed7      := 0
00849 : cmCSoMed8      := 0
00850 : cmCSoMed9      := 0
00851 : cmCSoMed10     := 0
00852 : cmCSoMed11     := 0
00853 : cmCSoMed12     := 0
00854 :
00855 : cmCSoMax1      := 0
00856 : cmCSoMax2      := 0
00857 : cmCSoMax3      := 0
00858 : cmCSoMax4      := 0
00859 : cmCSoMax5      := 0
00860 : cmCSoMax6      := 0
00861 : cmCSoMax7      := 0
00862 : cmCSoMax8      := 0
00863 : cmCSoMax9      := 0
00864 : cmCSoMax10     := 0
```

```
00865 : cmCSoMax11 := 0
00866 : cmCSoMax12 := 0
00867 :
00868 : cmACSoMx1 := 0
00869 : cmACSoMx2 := 0
00870 : cmACSoMx3 := 0
00871 : cmACSoMx4 := 0
00872 : cmACSoMx5 := 0
00873 : cmACSoMx6 := 0
00874 : cmACSoMx7 := 0
00875 : cmACSoMx8 := 0
00876 : cmACSoMx9 := 0
00877 : cmACSoMx10 := 0
00878 : cmACSoMx11 := 0
00879 : cmACSoMx12 := 0
00880 :
00881 : cmCSoMin1 := 999.99
00882 : cmCSoMin2 := 999.99
00883 : cmCSoMin3 := 999.99
00884 : cmCSoMin4 := 999.99
00885 : cmCSoMin5 := 999.99
00886 : cmCSoMin6 := 999.99
00887 : cmCSoMin7 := 999.99
00888 : cmCSoMin8 := 999.99
00889 : cmCSoMin9 := 999.99
00890 : cmCSoMin10 := 999.99
00891 : cmCSoMin11 := 999.99
00892 : cmCSoMin12 := 999.99
00893 :
00894 : cmACSoMn1 := 0
00895 : cmACSoMn2 := 0
00896 : cmACSoMn3 := 0
00897 : cmACSoMn4 := 0
00898 : cmACSoMn5 := 0
00899 : cmACSoMn6 := 0
00900 : cmACSoMn7 := 0
00901 : cmACSoMn8 := 0
00902 : cmACSoMn9 := 0
00903 : cmACSoMn10 := 0
00904 : cmACSoMn11 := 0
00905 : cmACSoMn12 := 0
00906 :
00907 : cmFechaUAct := CTOD( " / / " )
00908 : cmAltElMed := " "
00909 :
00910 : Return
00911 :
00912 :
```

```
00913 : FUNCTION loadMEHVM()  
00914 :  
00915 : cmCMoMed1      := CMOMED1  
00916 : cmCMoMed2      := CMOMED2  
00917 : cmCMoMed3      := CMOMED3  
00918 : cmCMoMed4      := CMOMED4  
00919 : cmCMoMed5      := CMOMED5  
00920 : cmCMoMed6      := CMOMED6  
00921 : cmCMoMed7      := CMOMED7  
00922 : cmCMoMed8      := CMOMED8  
00923 : cmCMoMed9      := CMOMED9  
00924 : cmCMoMed10     := CMOMED10  
00925 : cmCMoMed11     := CMOMED11  
00926 : cmCMoMed12     := CMOMED12  
00927 :  
00928 : cmCMoMax1      := CMOMAX1  
00929 : cmCMoMax2      := CMOMAX2  
00930 : cmCMoMax3      := CMOMAX3  
00931 : cmCMoMax4      := CMOMAX4  
00932 : cmCMoMax5      := CMOMAX5  
00933 : cmCMoMax6      := CMOMAX6  
00934 : cmCMoMax7      := CMOMAX7  
00935 : cmCMoMax8      := CMOMAX8  
00936 : cmCMoMax9      := CMOMAX9  
00937 : cmCMoMax10     := CMOMAX10  
00938 : cmCMoMax11     := CMOMAX11  
00939 : cmCMoMax12     := CMOMAX12  
00940 :  
00941 : cmACMoMx1      := ACMOMX1  
00942 : cmACMoMx2      := ACMOMX2  
00943 : cmACMoMx3      := ACMOMX3  
00944 : cmACMoMx4      := ACMOMX4  
00945 : cmACMoMx5      := ACMOMX5  
00946 : cmACMoMx6      := ACMOMX6  
00947 : cmACMoMx7      := ACMOMX7  
00948 : cmACMoMx8      := ACMOMX8  
00949 : cmACMoMx9      := ACMOMX9  
00950 : cmACMoMx10     := ACMOMX10  
00951 : cmACMoMx11     := ACMOMX11  
00952 : cmACMoMx12     := ACMOMX12  
00953 :  
00954 : cmCMoMin1      := CMOMIN1  
00955 : cmCMoMin2      := CMOMIN2  
00956 : cmCMoMin3      := CMOMIN3  
00957 : cmCMoMin4      := CMOMIN4  
00958 : cmCMoMin5      := CMOMIN5  
00959 : cmCMoMin6      := CMOMIN6  
00960 : cmCMoMin7      := CMOMIN7
```

```
00961 : cmCMoMin8      := CMOMIN8
00962 : cmCMoMin9      := CMOMIN9
00963 : cmCMoMin10     := CMOMIN10
00964 : cmCMoMin11     := CMOMIN11
00965 : cmCMoMin12     := CMOMIN12
00966 :
00967 : cmACMoMn1       := ACMOMN1
00968 : cmACMoMn2       := ACMOMN2
00969 : cmACMoMn3       := ACMOMN3
00970 : cmACMoMn4       := ACMOMN4
00971 : cmACMoMn5       := ACMOMN5
00972 : cmACMoMn6       := ACMOMN6
00973 : cmACMoMn7       := ACMOMN7
00974 : cmACMoMn8       := ACMOMN8
00975 : cmACMoMn9       := ACMOMN9
00976 : cmACMoMn10      := ACMOMN10
00977 : cmACMoMn11      := ACMOMN11
00978 : cmACMoMn12      := ACMOMN12
00979 :
00980 : cmAltMed1       := ALTMED1
00981 : cmAltMed2       := ALTMED2
00982 : cmAltMed3       := ALTMED3
00983 : cmAltMed4       := ALTMED4
00984 : cmAltMed5       := ALTMED5
00985 : cmAltMed6       := ALTMED6
00986 : cmAltMed7       := ALTMED7
00987 : cmAltMed8       := ALTMED8
00988 : cmAltMed9       := ALTMED9
00989 : cmAltMed10      := ALTMED10
00990 : cmAltMed11      := ALTMED11
00991 : cmAltMed12      := ALTMED12
00992 :
00993 : cmAltMax1       := ALTMAX1
00994 : cmAltMax2       := ALTMAX2
00995 : cmAltMax3       := ALTMAX3
00996 : cmAltMax4       := ALTMAX4
00997 : cmAltMax5       := ALTMAX5
00998 : cmAltMax6       := ALTMAX6
00999 : cmAltMax7       := ALTMAX7
01000 : cmAltMax8       := ALTMAX8
01001 : cmAltMax9       := ALTMAX9
01002 : cmAltMax10      := ALTMAX10
01003 : cmAltMax11      := ALTMAX11
01004 : cmAltMax12      := ALTMAX12
01005 :
01006 : cmAA1tMx1       := AALTMX1
01007 : cmAA1tMx2       := AALTMX2
01008 : cmAA1tMx3       := AALTMX3
```

```
01009 : cmAAItMx4      := AALTMX4
01010 : cmAAItMx5      := AALTMX5
01011 : cmAAItMx6      := AALTMX6
01012 : cmAAItMx7      := AALTMX7
01013 : cmAAItMx8      := AALTMX8
01014 : cmAAItMx9      := AALTMX9
01015 : cmAAItMx10     := AALTMX10
01016 : cmAAItMx11     := AALTMX11
01017 : cmAAItMx12     := AALTMX12
01018 :
01019 : cmAltMin1       := ALTMIN1
01020 : cmAltMin2       := ALTMIN2
01021 : cmAltMin3       := ALTMIN3
01022 : cmAltMin4       := ALTMIN4
01023 : cmAltMin5       := ALTMIN5
01024 : cmAltMin6       := ALTMIN6
01025 : cmAltMin7       := ALTMIN7
01026 : cmAltMin8       := ALTMIN8
01027 : cmAltMin9       := ALTMIN9
01028 : cmAltMin10      := ALTMIN10
01029 : cmAltMin11      := ALTMIN11
01030 : cmAltMin12      := ALTMIN12
01031 :
01032 : cmAAItMn1       := AALTMN1
01033 : cmAAItMn2       := AALTMN2
01034 : cmAAItMn3       := AALTMN3
01035 : cmAAItMn4       := AALTMN4
01036 : cmAAItMn5       := AALTMN5
01037 : cmAAItMn6       := AALTMN6
01038 : cmAAItMn7       := AALTMN7
01039 : cmAAItMn8       := AALTMN8
01040 : cmAAItMn9       := AALTMN9
01041 : cmAAItMn10      := AALTMN10
01042 : cmAAItMn11      := AALTMN11
01043 : cmAAItMn12      := AALTMN12
01044 :
01045 : cmCSoMed1       := CSOMED1
01046 : cmCSoMed2       := CSOMED2
01047 : cmCSoMed3       := CSOMED3
01048 : cmCSoMed4       := CSOMED4
01049 : cmCSoMed5       := CSOMED5
01050 : cmCSoMed6       := CSOMED6
01051 : cmCSoMed7       := CSOMED7
01052 : cmCSoMed8       := CSOMED8
01053 : cmCSoMed9       := CSOMED9
01054 : cmCSoMed10      := CSOMED10
01055 : cmCSoMed11      := CSOMED11
01056 : cmCSoMed12      := CSOMED12
```

```
01057 :  
01058 : cmCSoMax1      := CSOMAX1  
01059 : cmCSoMax2      := CSOMAX2  
01060 : cmCSoMax3      := CSOMAX3  
01061 : cmCSoMax4      := CSOMAX4  
01062 : cmCSoMax5      := CSOMAX5  
01063 : cmCSoMax6      := CSOMAX6  
01064 : cmCSoMax7      := CSOMAX7  
01065 : cmCSoMax8      := CSOMAX8  
01066 : cmCSoMax9      := CSOMAX9  
01067 : cmCSoMax10     := CSOMAX10  
01068 : cmCSoMax11     := CSOMAX11  
01069 : cmCSoMax12     := CSOMAX12  
01070 :  
01071 : cmACSoMx1      := ACSOMX1  
01072 : cmACSoMx2      := ACSOMX2  
01073 : cmACSoMx3      := ACSOMX3  
01074 : cmACSoMx4      := ACSOMX4  
01075 : cmACSoMx5      := ACSOMX5  
01076 : cmACSoMx6      := ACSOMX6  
01077 : cmACSoMx7      := ACSOMX7  
01078 : cmACSoMx8      := ACSOMX8  
01079 : cmACSoMx9      := ACSOMX9  
01080 : cmACSoMx10     := ACSOMX10  
01081 : cmACSoMx11     := ACSOMX11  
01082 : cmACSoMx12     := ACSOMX12  
01083 :  
01084 : cmCSoMin1      := CSOMIN1  
01085 : cmCSoMin2      := CSOMIN2  
01086 : cmCSoMin3      := CSOMIN3  
01087 : cmCSoMin4      := CSOMIN4  
01088 : cmCSoMin5      := CSOMIN5  
01089 : cmCSoMin6      := CSOMIN6  
01090 : cmCSoMin7      := CSOMIN7  
01091 : cmCSoMin8      := CSOMIN8  
01092 : cmCSoMin9      := CSOMIN9  
01093 : cmCSoMin10     := CSOMIN10  
01094 : cmCSoMin11     := CSOMIN11  
01095 : cmCSoMin12     := CSOMIN12  
01096 :  
01097 : cmACSoMn1      := ACSOMN1  
01098 : cmACSoMn2      := ACSOMN2  
01099 : cmACSoMn3      := ACSOMN3  
01100 : cmACSoMn4      := ACSOMN4  
01101 : cmACSoMn5      := ACSOMN5  
01102 : cmACSoMn6      := ACSOMN6  
01103 : cmACSoMn7      := ACSOMN7  
01104 : cmACSoMn8      := ACSOMN8
```

```
01105 : cmACSoMn9      := ACSOMN9
01106 : cmACSoMn10     := ACSOMN10
01107 : cmACSoMn11     := ACSOMN11
01108 : cmACSoMn12     := ACSOMN12
01109 :
01110 : cmFechaUAct     := FECHAUACT
01111 : cmAltElMed      := ALTELMED
01112 :
01113 : Return
01114 :
01115 :
01116 :
```



```
00001 : *****
00002 : *
00003 : *      Archivo : ESTMETVA.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: estmetva
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    01 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo de Valores anuales de
00011 : *      Estaciones Meteorologicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "EstacId.Ch"
00017 : #include "EstMetVA.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aEstacId, aEstMetVA
00022 : PRIVATE cmDepto, cmCodigoEst, cmFechaUAct, cmPANio, cmUANio
00023 : PRIVATE cmPreMed, cmTemMed, cmHumMed, cmEvaMed
00024 :
00025 : SET CENTURY ON
00026 :
00027 : DO Pantalla WITH "Valores Anuales de Est.Meteorológicas", "EstMedVA"
00028 :
00029 : aDepto      := OpenDbf( "DEPTO", aIndDepto, 1 )
00030 : aEstacId    := OpenDbf( "ESTACID", aIndEstId, 1 )
00031 : aEstMVA     := OpenDbf( "ESTMETVA", aIndEMVA, 1 )
00032 :
00033 : *SET KEY -9 TO list_val // Programa Tecla de Función
00034 :
00035 : ABM( "Registro", { || PEMVAO() }, { || PEMVA1() },;
00036 :      { || PEMVA2() }, { || PEMVA2() }, { || bAltaEMVA() },;
00037 :      { || dbDelete() }, { || bModiEMVA() } )
00038 :
00039 : SET KEY -9 TO // Programa Tecla de Función
00040 : SET CENTURY OFF
00041 : RELEASE ALL LIKE cm*
00042 :
00043 : CloseDbf( "ESTMETVA" )
00044 : CloseDbf( "ESTACID" )
00045 : CloseDbf( "DEPTO" )
00046 :
00047 : RETURN
00048 :
```

```
00049 :
00050 : FUNCTION baItaEMVA
00051 :
00052 : REPLACE DEPTO      WITH cmDepto,;
00053 :      CODIGOEST WITH cmCodigoEst
00054 :
00055 : RETURN
00056 :
00057 :
00058 : FUNCTION PEMVA0()
00059 :
00060 : @ 4, 1 SAY "Codigo Departamento:"
00061 : @ 5, 1 SAY "Tipo de Estación..."
00062 : @ 6, 1 SAY "Codigo de Estación:"
00063 : @ 7, 1 SAY "Primer Año Serie..."
00064 : @ 8, 1 SAY "Ultimo Año Serie..."
00065 : @ 9, 1 SAY "Fecha Ult.Actualiz.:"
00066 : @ 10, 1 SAY "Precip.Media.[mm/a]:"
00067 : @ 11, 1 SAY "Temperat.Media.[°C]:"
00068 : @ 12, 1 SAY "Humedad Media[%Rel]:"
00069 : @ 13, 1 SAY "Evapor.Media.[mm/m]:"
00070 :
00071 : RETURN
00072 :
00073 :
00074 : FUNCTION PEMVA1()
00075 :
00076 : PRIVATE KN1, KN2, KN3
00077 :
00078 : KN1 := " "
00079 : KN2 := " "
00080 : KN3 := 0
00081 : iniMEMVA()
00082 :
00083 : @ 4, 23 CLEAR TO 19, 79
00084 :
00085 : * 4, 22 SAY CHR(18)
00086 : @ 4, 23 GET KN1 PICTURE "99";
00087 :      VALID vaExist( "DEPTO", KN1,;
00088 :      "Departamento Inexistente", { || showDepto( 4 ) } )
00089 : * 5, 22 SAY CHR(18)
00090 : @ 5, 23 GET KN2 PICTURE "!" VALID KN2$"AM"
00091 : * 6, 22 SAY CHR(18)
00092 : @ 6, 23 GET KN3 PICTURE "99999.99";
00093 :      VALID vaExist( "ESTACID", RIGHT( "00"+ALLTRIM(KN1),2)+KN2+;
00094 :      STR( KN3, 8, 2 ),;
00095 :      "Estación Meteorológica Inexistente", NIL )
00096 :
```

```
00097 : READ
00098 : CLEAR GETS
00099 :
00100 : cmDepto      := RIGHT( "00"+ALLTRIM(KN1),2)
00101 : cmTipoEst   := KN2
00102 : cmCodigoEst := STR( KN3, 8, 2 )
00103 :
00104 : RETURN( cmDepto+cmCodigoEst )
00105 :
00106 :
00107 : FUNCTION PEMVA2()
00108 :
00109 : @ 7, 23 GET cmPAnio      PICTURE "9999"
00110 : @ 8, 23 GET cmUANio     PICTURE "9999"
00111 : @ 9, 23 GET cmFechaUAct
00112 : @ 10, 23 GET cmPreMed   PICTURE "9999.9"
00113 : @ 11, 23 GET cmTemMed   PICTURE "99.9"
00114 : @ 12, 23 GET cmHumMed   PICTURE "999.9"
00115 : @ 13, 23 GET cmEvaMed   PICTURE "9999.9"
00116 :
00117 : RETURN
00118 :
00119 :
00120 : Func bModiEMVA()
00121 :
00122 : REPLACE PANIO      WITH cmPAnio,;
00123 :      UANIO        WITH cmUANio,;
00124 :      PREMED       WITH cmPreMed,;
00125 :      TEMMED       WITH cmTemMed,;
00126 :      HUMMED       WITH cmHumMed,;
00127 :      EVAMED       WITH cmEvaMed,;
00128 :      FECHAUACT    WITH cmFechaUact
00129 :
00130 : Return
00131 :
00132 :
00133 : FUNCTION iniMEMVA()
00134 :
00135 : cmPAnio      := 0
00136 : cmUANio      := 0
00137 : cmPreMed     := 0
00138 : cmTemMed     := 0
00139 : cmHumMed     := 0
00140 : cmEvaMed     := 0
00141 : cmFechaUAct := CTOD( " / / " )
00142 :
00143 : Return
00144 :
```

```
00145 :  
00146 : FUNCTION loadMEMVA()  
00147 :  
00148 : cmPAnio      := PANIO  
00149 : cmUAnio      := UANIO  
00150 : cmPreMed     := PREMED  
00151 : cmTemMed     := TEMMED  
00152 : cmHumMed     := HUMMED  
00153 : cmEvaMed     := EVAMED  
00154 : cmFechaUAct := FECHAUACT  
00155 :  
00156 : Return  
00157 :  
00158 :  
00159 :
```

```
00001 : *****
00002 : *
00003 : *      Archivo : ESTMETVE.PRG
00004 : *      Sistema : Sistema Banco de Datos Hidrogeológicos
00005 : *      Programa: estmetve
00006 : *      Autor:    Ing. Daniel Fabián GAZZOLA
00007 : *      Fecha:    01 de Setiembre de 1.993
00008 : *
00009 : *      Descripción:
00010 : *                  Actualiza Archivo de Valores extremos de
00011 : *      Estaciones Meteorologicas.
00012 : *
00013 : *****
00014 :
00015 : #include "Depto.Ch"
00016 : #include "EstacId.Ch"
00017 : #include "EstMetVE.Ch"
00018 :
00019 : bBlankField := { |aA, cField| SPACE(aA[2][ASCAN(aA[1],cField)]) }
00020 :
00021 : PRIVATE aDepto, aEstacId, aEstMetVE
00022 : PRIVATE cmDepto, cmCodigoEst, cmFechaUAct, cmAnio
00023 : PRIVATE cmPreMax, cmAPreMx, cmPreMin, cmAPreMn
00024 : PRIVATE cmTemMax, cmATemMx, cmTemMin, cmATemMn
00025 : PRIVATE cmHumMax, cmAHumMx, cmHumMin, cmAHumMn
00026 : PRIVATE cmEvaMax, cmAEvaMx, cmEvaMin, cmAEvaMn
00027 :
00028 : SET CENTURY ON
00029 :
00030 : DO Pantalla WITH "Valores Extremos de Est.Meteorológicas", "EstMedVE"
00031 :
00032 : aDepto    := OpenDbf( "DEPTO", aIndDepto, 1 )
00033 : aEstacId  := OpenDbf( "ESTACID", aIndEstId, 1 )
00034 : aEstMVE   := OpenDbf( "ESTMETVE", aIndEMVE, 1 )
00035 :
00036 : *SET KEY -9 TO list_val // Programa Tecla de Función
00037 :
00038 : ABM( "Registro", { || PEMVE0() }, { || PEMVE1() },;
00039 :      { || PEMVE2() }, { || PEMVE2() }, { || bAltaEMVE() },;
00040 :      { || dbDelete() }, { || bModiEMVE() } )
00041 :
00042 : SET KEY -9 TO // Programa Tecla de Función
00043 : SET CENTURY OFF
00044 : RELEASE ALL LIKE cm*
00045 :
00046 : CloseDbf( "ESTMETVE" )
00047 : CloseDbf( "ESTACID" )
00048 : CloseDbf( "DEPTO" )
```

```
00049 :  
00050 : RETURN  
00051 :  
00052 :  
00053 : FUNCTION bAltaEMVE  
00054 :  
00055 : REPLACE DEPTO      WITH cmDepto,;  
00056 :      CODIGOEST WITH cmCodigoEst  
00057 :  
00058 : RETURN  
00059 :  
00060 :  
00061 : FUNCTION PEMVEO()  
00062 :  
00063 : @ 4, 1 SAY "Codigo Departamento:"  
00064 : @ 5, 1 SAY "Tipo de Estación...:"  
00065 : @ 6, 1 SAY "Codigo de Estación:"  
00066 : @ 7, 1 SAY "Fecha Ult.Actualiz:"  
00067 : @ 8, 1 SAY "Precip.Máxima[mm/a]:"  
00068 : @ 8, 41 SAY "Año de Ocurrencia..."  
00069 : @ 9, 1 SAY "Precip.Mínima[mm/a]:"  
00070 : @ 9, 41 SAY "Año de Ocurrencia..."  
00071 : @ 10, 1 SAY "Temperat.Máxima[°C]:"  
00072 : @ 10, 41 SAY "Año de Ocurrencia..."  
00073 : @ 11, 1 SAY "Temperat.Mínima[°C]:"  
00074 : @ 11, 41 SAY "Año de Ocurrencia..."  
00075 : @ 12, 1 SAY "Humedad Máxima [%R]:"  
00076 : @ 12, 41 SAY "Año de Ocurrencia..."  
00077 : @ 13, 1 SAY "Humedad Mínima [%R]:"  
00078 : @ 13, 41 SAY "Año de Ocurrencia..."  
00079 : @ 14, 1 SAY "Evapor.Máxima[mm/m]:"  
00080 : @ 14, 41 SAY "Año de Ocurrencia..."  
00081 : @ 15, 1 SAY "Evapor.Máxima[mm/m]:"  
00082 : @ 15, 41 SAY "Año de Ocurrencia..."  
00083 :  
00084 : RETURN  
00085 :  
00086 :  
00087 : FUNCTION PEMVE1()  
00088 :  
00089 : PRIVATE KN1, KN2, KN3  
00090 :  
00091 : KN1 := " "  
00092 : KN2 := " "  
00093 : KN3 := 0  
00094 : iniMEMVE()  
00095 :  
00096 : @ 4, 23 CLEAR TO 7, 79
```

```
00097 : @ 8, 23 CLEAR TO 19, 40
00098 : @ 8, 63 CLEAR TO 19, 79
00099 :
00100 : * 4, 22 SAY CHR(18)
00101 : @ 4, 23 GET KN1 PICTURE "99";
00102 :         VALID vaexist( "DEPTO", KN1,;
00103 :         "Departamento Inexistente", { || showDepto( 4 ) } )
00104 : * 5, 22 SAY CHR(18)
00105 : @ 5, 23 GET KN2 PICTURE "!" VALID KN2$"AM"
00106 : * 6, 22 SAY CHR(18)
00107 : @ 6, 23 GET KN3 PICTURE "99999.99";
00108 :         VALID vaexist( "ESTACID", RIGHT( "00"+ALLTRIM(KN1),2)+KN2+;
00109 :         STR( KN3, 8, 2 ),;
00110 :         "Estación Meteorológica Inexistente", NIL )
00111 :
00112 : READ
00113 : CLEAR GETS
00114 :
00115 : cmDepto      := RIGHT( "00"+ALLTRIM(KN1),2)
00116 : cmTipoEst   := KN2
00117 : cmCodigoEst := STR( KN3, 8, 2 )
00118 :
00119 : RETURN( cmDepto+cmCodigoEst )
00120 :
00121 :
00122 : FUNCTION PEMVE2()
00123 :
00124 : @ 7, 23 GET cmFechaUAct
00125 : @ 8, 23 GET cmPreMax      PICTURE "9999.9"
00126 : @ 8, 63 GET cmAPreMx     PICTURE "9999"
00127 : @ 9, 23 GET cmPreMin     PICTURE "9999.9"
00128 : @ 9, 63 GET cmAPreMn     PICTURE "9999"
00129 : @ 10, 23 GET cmTemMax    PICTURE "99.9"
00130 : @ 10, 63 GET cmATemMx   PICTURE "9999"
00131 : @ 11, 23 GET cmTemMin    PICTURE "999.9"
00132 : @ 11, 63 GET cmATemMn   PICTURE "9999"
00133 : @ 12, 23 GET cmHumMax   PICTURE "999.9"
00134 : @ 12, 63 GET cmAHumMx  PICTURE "9999"
00135 : @ 13, 23 GET cmHumMin   PICTURE "999.9"
00136 : @ 13, 63 GET cmAHumMn  PICTURE "9999"
00137 : @ 14, 23 GET cmEvaMax   PICTURE "9999.9"
00138 : @ 14, 63 GET cmAEvaMx  PICTURE "9999"
00139 : @ 15, 23 GET cmEvaMin   PICTURE "9999.9"
00140 : @ 15, 63 GET cmAEvaMn  PICTURE "9999"
00141 :
00142 : RETURN
00143 :
00144 :
```

```
00145 : Func bModiEMVE()
00146 :
00147 : REPLACE PREMAX      WITH cmPreMax,;
00148 :      APREMX      WITH cmAPreMx,;
00149 :      PREMIN      WITH cmPreMin,;
00150 :      APREMN      WITH cmAPreMn,;
00151 :      TEMMAX      WITH cmTemMax,;
00152 :      ATEMMX      WITH cmATemMx,;
00153 :      TEMMIN      WITH cmTemMin,;
00154 :      ATEMMN      WITH cmATemMn,;
00155 :      HUMMAX      WITH cmHumMax,;
00156 :      AHUMMX      WITH cmAHumMx,;
00157 :      HUMMIN      WITH cmHumMin,;
00158 :      AHUMMN      WITH cmAHumMn,;
00159 :      EVAMAX      WITH cmEvaMax,;
00160 :      AEVAMX      WITH cmAEvaMx,;
00161 :      EVAMIN      WITH cmEvaMin,;
00162 :      AEVAMN      WITH cmAEvaMn,;
00163 :      FECHAUACT WITH cmFechaUact
00164 :
00165 : Return
00166 :
00167 :
00168 : FUNCTION iniMEMVE()
00169 :
00170 : cmPreMax      := 0
00171 : cmAPreMx      := 0
00172 : cmPreMin      := 0
00173 : cmAPreMn      := 0
00174 : cmTemMax      := 0
00175 : cmATemMx      := 0
00176 : cmTemMin      := 0
00177 : cmATemMn      := 0
00178 : cmHumMax      := 0
00179 : cmAHumMx      := 0
00180 : cmHumMin      := 0
00181 : cmAHumMn      := 0
00182 : cmEvaMax      := 0
00183 : cmAEvaMx      := 0
00184 : cmEvaMin      := 0
00185 : cmAEvaMn      := 0
00186 : cmFechaUAct := CTOD( " / / " )
00187 :
00188 : Return
00189 :
00190 :
00191 : FUNCTION loadMEMVE()
00192 :
```



```
00193 : cmPreMax      := PREMAX
00194 : cmAPreMx      := APREMX
00195 : cmPreMin      := PREMIN
00196 : cmAPreMn      := APREMn
00197 : cmTemMax      := TEMMAX
00198 : cmATemMx      := ATEMMX
00199 : cmTemMin      := TEMMIN
00200 : cmATemMn      := ATEMMN
00201 : cmHumMax      := HUMMAX
00202 : cmAHumMx      := AHUMMX
00203 : cmHumMin      := HUMMIN
00204 : cmAHumMn      := AHUMMN
00205 : cmEvaMax      := EVAMAX
00206 : cmAEvaMx      := AEVAMX
00207 : cmEvaMin      := EVAMIN
00208 : cmAEvaMn      := AEVAMN
00209 : cmFechaUAct   := FECHAUACT
00210 :
00211 : Return
00212 :
00213 :
00214 :
```