

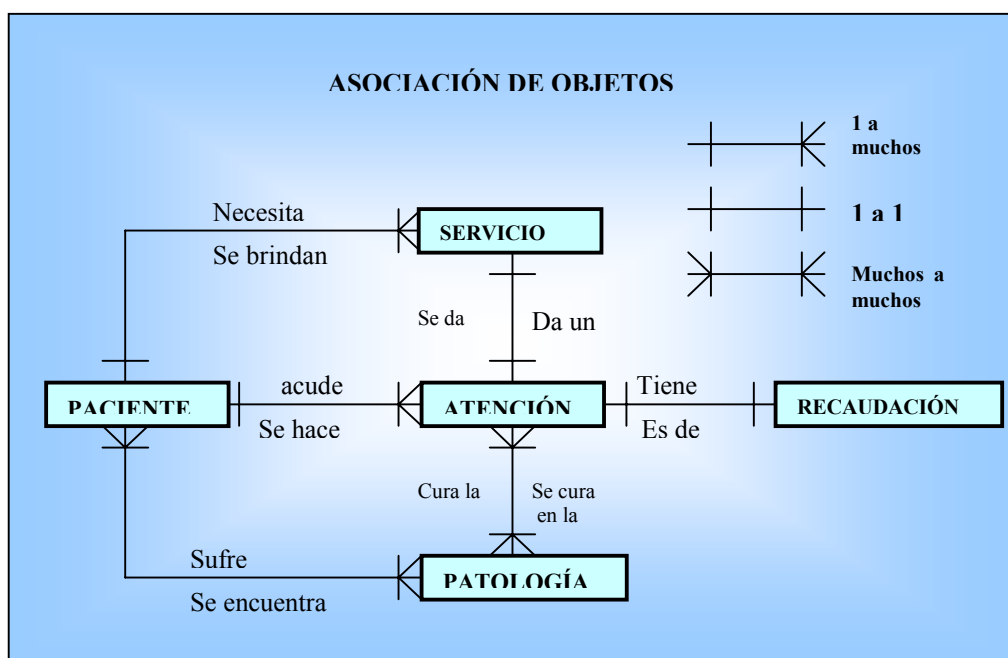
MANUAL TÉCNICO DEL SISTEMA DE INFORMACIÓN GERENCIAL DE LOS CENTROS DE REHABILITACIÓN MEDICA DEL INNFA

DISEÑO LÓGICO

Construcción del depósito de datos del SIG y determinación de los módulos del sistema

Tomando en cuenta los servicios con los que cuenta los CRM y los datos almacenados en la base de datos mediante el sistema de procesamiento de esta institución podemos distinguir los siguientes objetos útiles para el sistema gerencial: Pacientes, Atenciones, Recaudaciones, Servicios, Patologías.

A estos objetos se los podría relacionar de la siguiente manera:



Diseño de la estructura de objetos

Tablas de la base de datos

Las tablas que contiene la base de datos BCRM4 son: Paciente, Atención, Recaudación, Patología, Servicios que se detallarán mas adelante en la sección Metadata.

Diagrama de Flujo de Datos

Simbología

Para la representación de los diagramas de contexto se ha utilizado los siguientes símbolos:




SÍMBOLO	DESCRIPCIÓN
 Datos almacenados:	Representa el deposito donde se encuentran almacenados los datos.
 Proceso:	Representa los procesos de transformación.
 Pantalla:	Representa los reportes presentados por pantalla.
 Documento:	Representa los reportes enviados a la impresora.
	Representa el destino de los datos. Su punta señala el destino de los datos.

Diagrama de Contexto

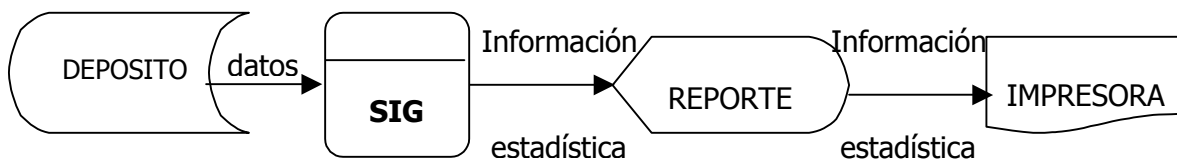


Diagrama de nivel 0

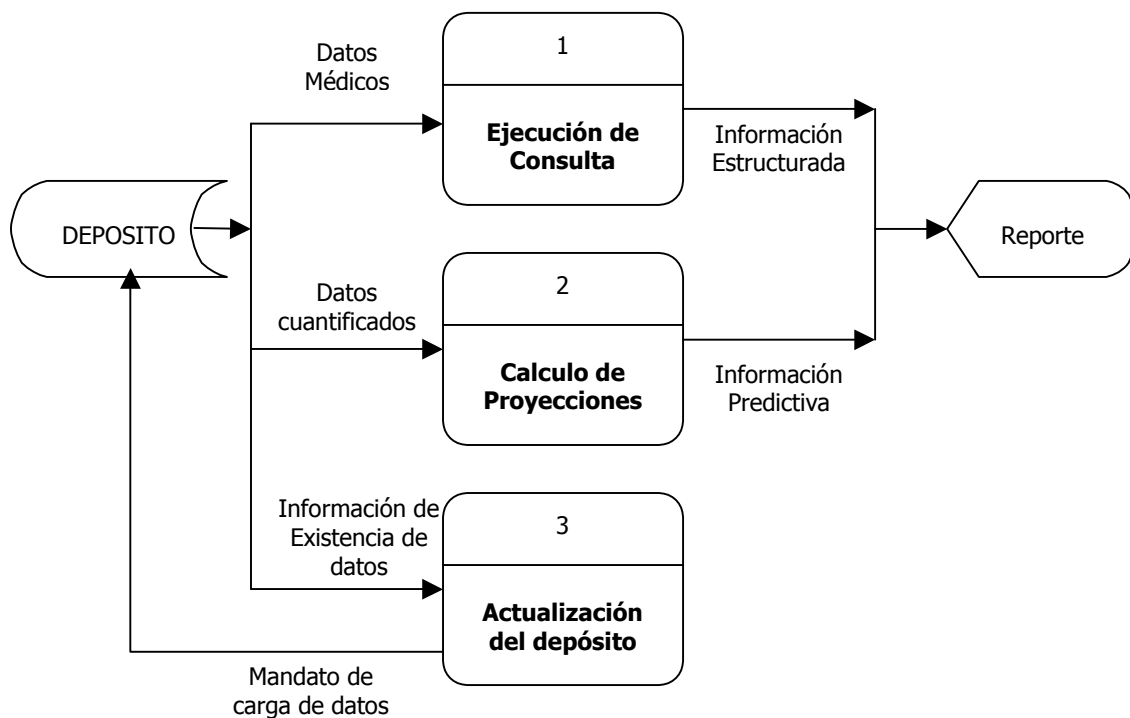


Diagrama de nivel 1

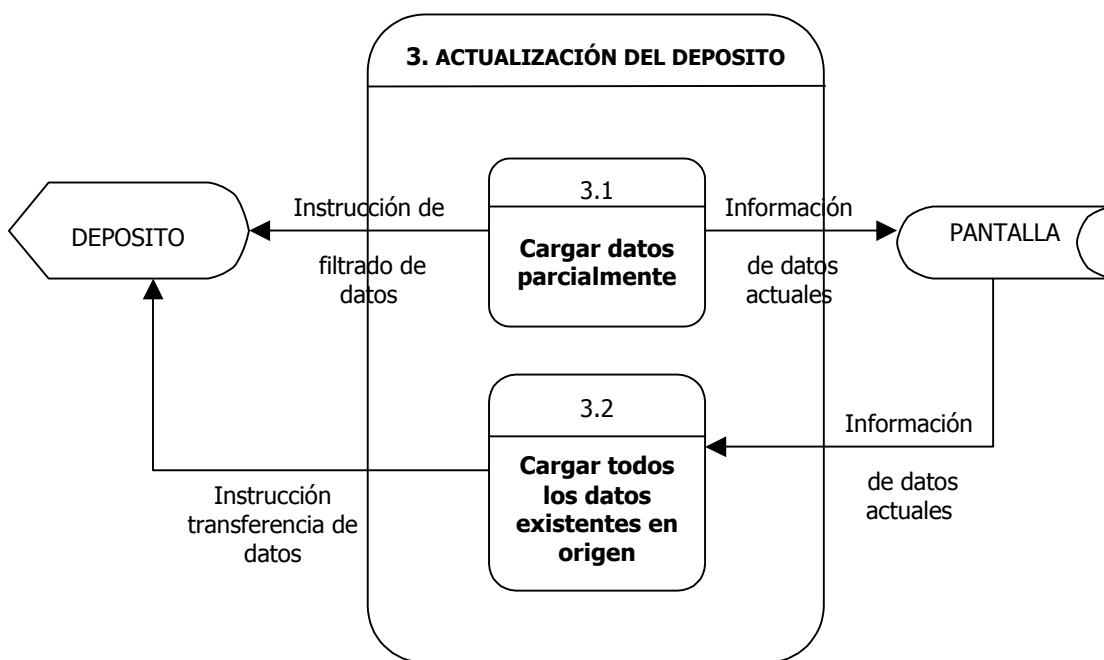


Diagrama de nivel 1

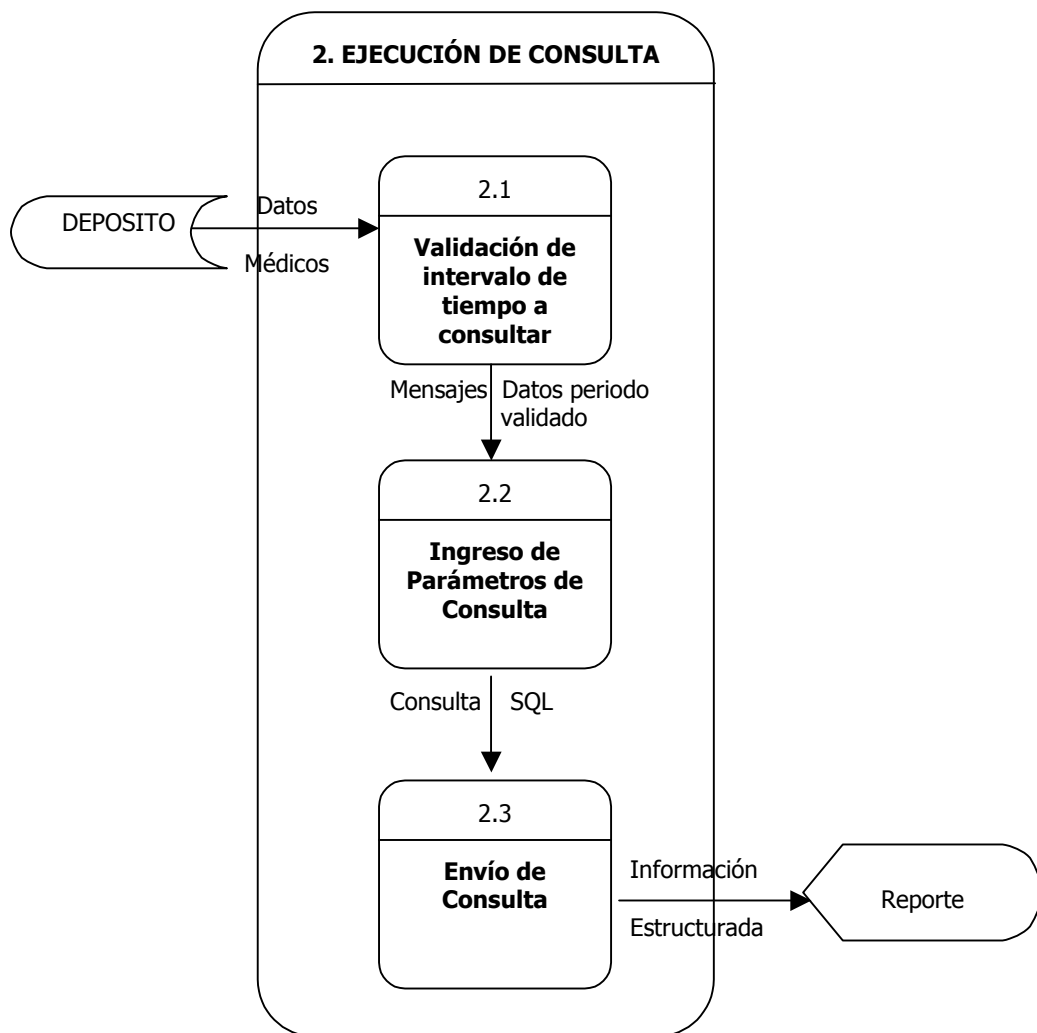
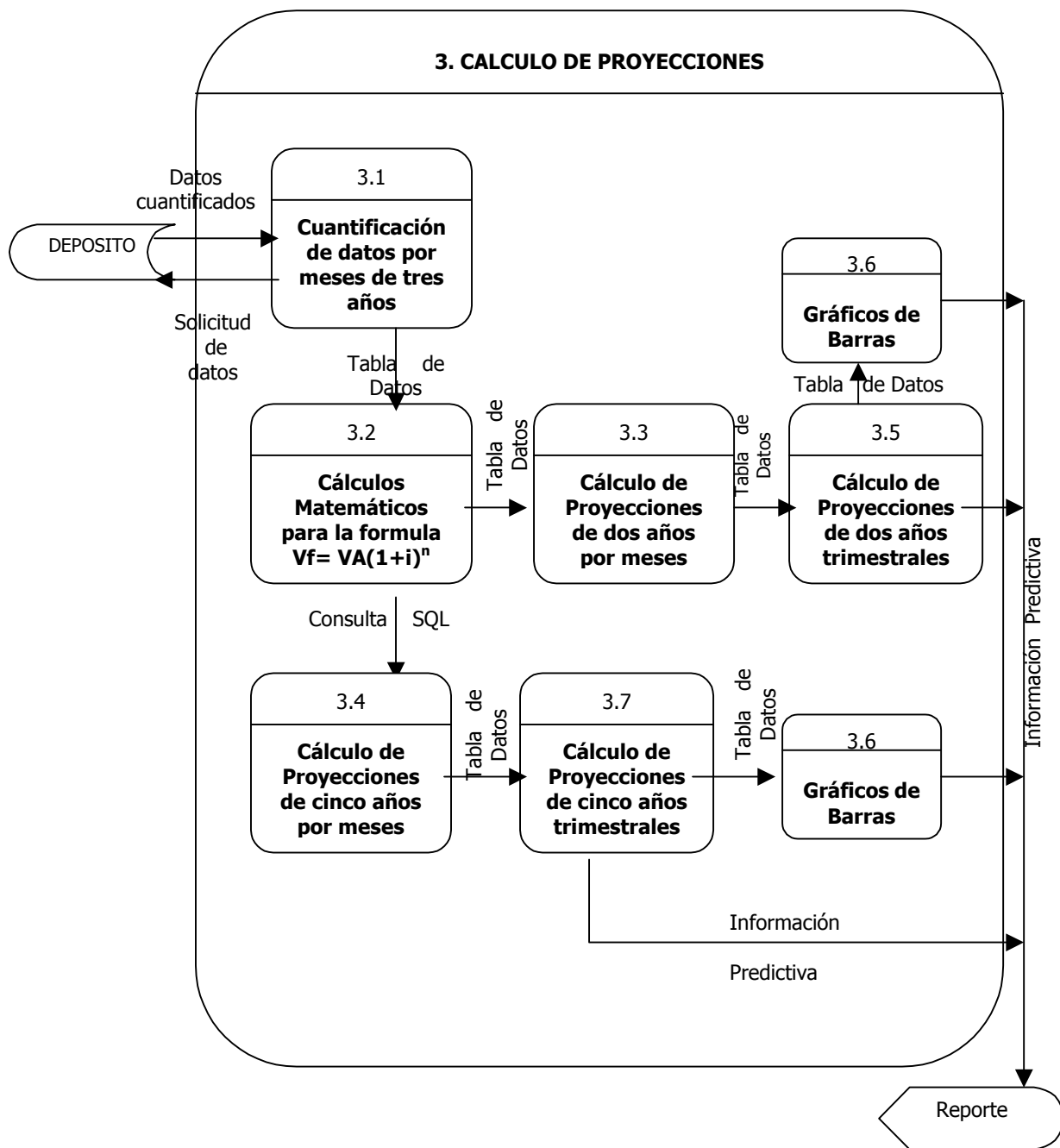


Diagrama de nivel 1



Detalle de procesos de intercambio de datos

a. Medio Físico de transmisión de datos

El medio físico para la transmisión de datos es la red, basada en TCP/IP.

b. Modelo de comunicación

El modelo de comunicación utilizado es una conexión Client Access basada en ODBC:

c. Procesos de extracción de datos

La extracción de datos se realiza mediante consultas SQL que se automatizaron mediante el motor de base de datos con paquetes de SQL Server.

Operación de los Datos de la base del SIG

- **Limpieza de los datos:** En algunas columnas de las tablas a transferir se encontraron inconsistencias tales como: fechas de nacimiento con un valor numérico igual a 0, campos de caracteres específicos contenían blancos, columnas con datos fecha que eran de tipo numérico. Estas inconsistencias fueron resueltas con la utilización de filtros implementados con código SQL que permitieron que los datos se transfirieran adecuadamente para su uso en el SIG.
- **Clasificación de los datos:** La mayoría de consultas SQL multidimensionales necesarias para la obtención de resultados visuales del sistema no se podían llevar a cabo con las columnas obtenidas en la transferencia. Para lograr buenos resultados fue necesario la creación de columnas calculadas basándose en más de una columna original. La manera técnica de lograr este requisito para el funcionamiento del sistema se denomina **vistas** de SQL.
- **Procesos de mantenimiento y automatización:** Para la actualización permanente de la base de datos se automatizó los procesos de transferencia de datos mediante herramientas del manejador de base de datos con tareas programadas llamadas paquetes y, además, se implementó en el SIG la ejecución de las tareas en el momento que se requieran.

Métodos utilizados en la Implementación

Método utilizado para las proyecciones del sistema

El método elegido para utilizarse en el cálculo de datos futuros a partir de datos tomados de la experiencia del CRM4 se fundamenta en la evaluación e interpretación de los mismos mediante métodos cronológicos y de promedios móviles.

Al hacer el análisis de los datos primeramente como cambian en el tiempo, es decir, si las atenciones del centro de rehabilitación aumentan sistemáticamente conforme con el paso de los meses de cada año o cambiaban en otras formas se determina que el mejor método cuantitativo para las proyecciones es un método usado en la evaluación de proyectos económicos denominado Método de Entrada-Salida basado en la fórmula de cálculo siguiente:

$$VF = VA (1 + i)^n$$

Que se define como:

- VF Valor futuro o final
- VA Valor Actual
- i tasa de rentabilidad exigida
- n es el número de períodos

En esta fórmula todos estos parámetros se los toma como entes financieros, pero para el cálculo de las proyecciones de atenciones del centro son necesarias las siguientes adecuaciones de las variables de esta fórmula:

- VF Valor de la proyección de número de atenciones futuras
- VA Valor de atenciones ya efectuadas.
- i tasa de aumento de atenciones en los últimos tres años.
- n número de años a proyectar

La aplicación de la fórmula en este caso necesita los siguientes valores, esto, para la obtención del valor futuro de cada mes, de cada año que se desee proyectar:

- **VA** representa la media de los valores de las atenciones de cada mes de los dos últimos años.
- **i** es la tasa de aumento de crecimiento de atenciones se tomo como 0.07 de acuerdo a la observación de los datos pasados.

Ejemplo:

Si las atenciones del mes de enero de los años 2002,2003 fueran: 3939, 4274 entonces la proyección en el mismo mes en el año siguiente tendrá el siguiente valor:

$$VF = VA (1 + i)^n$$

$$VF = \text{media } (3939+4274) \cdot (1+0.07)^1$$

VF = 4701

El valor VF = 4701 es el valor aproximado de atenciones en Enero del 2004.

METADATA

Para el funcionamiento del Sistema Gerencial es necesario contar con una base de datos que lleve por nombre BDDCRM4 implementada en SQL Server 2000 Edición Personal.

La base de datos BDDCRM4 debe contener las siguientes tablas y vistas de datos:

TABLAS

CRPACIP: Datos del Paciente

CRPACIP	
CAMPO	DESCRIPCIÓN
CRPNHC	Número de historia clínica del paciente
CRPNOM	Nombre del paciente
CRPSEX	Sexo del paciente
CRPCAT	Categoría de exoneración del paciente
CRPSCA	Subcategoría de exoneración del paciente
CRPPRR	Provincia de residencia del paciente
CRPAF1	Afiliación al IESS
CRPFAP	Fecha de apertura de la historia clínica
CRPPAR	Parroquia de Procedencia(urbano/rural)

En esta tabla los datos se obtendrán de las bases de datos del sistema médico que se encuentra en el AS/400 mediante una subconsulta utilizando una instrucción SELECT :

```
SELECT  
CASE  
WHEN CRPAFI<>'S' AND CRPAFI<>'N' THEN 'NO APLICABLE'  
ELSE CRPAFI  
END AS CRPAF1,  
CRPNHC, CRPNOM, CRPSEX,CRPCAT,CRPSCA,CRPPRR,CRPFAP,CRPPAR  
FROM CRPACIP
```

Este SELECT se ejecuta mediante un paquete creado en el manejador de base de datos automáticamente desde el Sistema Gerencial al usar el menú Archivo/ Cargar Datos. Por facilidad de diseño se ha tomado como nombre de los campos los mismos nombres de la

tabla CRPACIP del sistema médico, pero haciendo una modificación en el campo CRPAFI para eliminar la inconsistencia encontrada como es algunos registros en blanco en este campo, sustituyendo estos registros mediante la X, en los demás casos se copiará la información del mismo modo como se encuentra en la tabla origen CRPACIP.

CRTRANP: Información de atenciones a los pacientes.

CRTRANP	
CAMPO	DESCRIPCIÓN
CRTNHC	Número de historia clínica del paciente
CRTFEA	Fecha de atención al paciente
CRTESP	Especialidad que atendió
CRTTKT	Número de ticket
CRTPA1	Patología del paciente
CRTEDA	Edad del paciente

En esta tabla los datos se obtendrán de la base de datos del sistema médico que se encuentra en el AS/400 mediante una subconsulta utilizando una instrucción SELECT :

```
SELECT  
CASE  
WHEN CRPFNA<>0 THEN (CRTFEA- CRPFNA)/10000  
ELSE 200  
END AS CRTEDA,  
CRTNHC,CRTFEA, CRTESP, CRTTKT,CRTPA1  
FROM CRPACIP, CRTRANP  
WHERE CRTNHC=CRPNHC
```

Este SELECT igualmente se ejecuta mediante un paquete creado en el manejador de base de datos automáticamente desde el Sistema Gerencial al usar el menú Archivo/ Cargar Datos. Por facilidad de diseño se ha tomado como nombre de los campos los mismos nombres de la tabla CRTRAP del sistema médico, a excepción del campo CRTEDA que es un campo calculado de acuerdo a la instrucción (CRTFEA- CRPFNA)/10000 donde CRTFEA es la fecha de atención al paciente, CRPFNA es la fecha de nacimiento del paciente(se ha dividido para 10000 ya que los dos campos son numéricos), en los demás casos se copiará la información del mismo modo como se encuentra en la tabla origen CRTRANP.

La instrucción SELECT puede tener una instrucción WHERE como la siguiente:

```
WHERE CRTNHC = CRPNHC AND CRTFEA >= ?
```

La cual indica que la subconsulta obtendrá solamente los registros donde CRTFEA (fecha de atención) sean iguales o mayores a la fecha que ingresará como parámetro y sustituirá al signo de interrogación. El parámetro se obtendrá mediante el Sistema Gerencial en la opción Cargar Datos/ Parte de los Datos donde se pide al usuario la fecha inicial desde la cual se desea cargar los datos.

Nota: Ya que la operación que se realiza para obtener la edad retorna un campo decimal, luego de transferir la tabla se procede a redondear al inmediato inferior el campo CRTEDA, mediante un paquete que se ejecuta mediante el SIG en la forma Cargar Datos automáticamente con la siguiente instrucción:

```
USE BDDCRM4  
UPDATE CRTRANP  
SET CRTEDA=ROUND(CRTEDA,0,1)
```

CRECAUP: Información de las recaudaciones de las atenciones a los pacientes.

CRTRANP	
CAMPO	DESCRIPCIÓN
CRENHC	Número de historia clínica del paciente
CRETKT	Número de Ticket
CREFET	Fecha de emisión de ticket
CREESP	Especialidad que se requiere
CREVLT	Costo de la atención

```
SELECT CRENHC, CRETKT, CREFET, CREESP, CREVLT  
FROM CRECAUP  
WHERE CREFET >= ?
```

Este SELECT igualmente se ejecuta mediante un paquete creado en el manejador de base de datos automáticamente desde el Sistema Gerencial al usar el menú Archivo/ Cargar Datos. Por facilidad de diseño se ha tomado como nombre de los campos los mismos nombres de la tabla CRECAUP del sistema médico.

La instrucción WHERE puede llevar igualmente un parámetro como en el caso de la tabla CRTRANP donde la subconsulta retornará solo los registros que el usuario desee. Se utilizará la misma información obtenida en la carga de datos del sistema.

CRTPATP: Información de las patologías existentes.

CRTPATP	
CAMPO	DESCRIPCIÓN
CRTPCO	Código de la patología
CRTPDE	Descripción de la patología

Esta tabla igualmente se transfiere del sistema AS/400 completamente.

CRSERVICIOS: Información de los servicios existentes en el centro médico.

CRSERVICIOS	
CAMPO	DESCRIPCIÓN
CRCOD	Código de la especialidad
CRDESC	Descripción de la especialidad
CRCATA	Valor de la atención con categoría A
CRCOD1	Agrupación por servicio

La tabla se transfiere del sistema AS/400, y se adiciona un campo nuevo(CRCOD1) calculado a partir de la división de CRCOD para 10000. Esto se realiza para obtener un código de agrupación de servicio, luego de transferir los datos con el campo calculado se debe redondear el valor ya que la división entre CRCOD para 10000 retorna un real y así poder obtener un número de código aceptable. O sea, el campo **CRCOD1 contendrá los tres primeros dígitos desde la izquierda del numero que se encuentre en el campo CRCOD.**

CRSERVICIOSG: Información de los códigos de agrupación de servicios.

CRSERVICIOSG	
CAMPO	DESCRIPCIÓN
CRCODSER	Código del servicio
CRDESSER	Descripción del servicio

Esta tabla ha sido creada, no corresponde a ninguna del sistema AS/400. Los campos que contiene es denominación que se ha hecho a cada servicio existente en el centro. El código que se ha dado a cada servicio corresponde a los primeros tres dígitos que se encuentran a la izquierda de los códigos de las especialidades que incluya el servicio, por ejemplo si todas las especialidades de medicina general contienen un código que empiece en 402... entonces al servicio(CRDESSER) medicina general se le ingresará un CRCODSER igual a 402. Los códigos de especialidad se pueden mirar en la tabla CRSERVICIOS en el campo CRCOD1.

PROVINCIAS: Información de Provincias.

CRSERVICIOSG	
CAMPO	DESCRIPCIÓN
CRTOPR	Código de provincia.
CRTODE	Descripción de provincia

Esta tabla se transfiere desde el sistema AS/400 y corresponde a la denominación de las provincias del país en el sistema médico.

FECHA

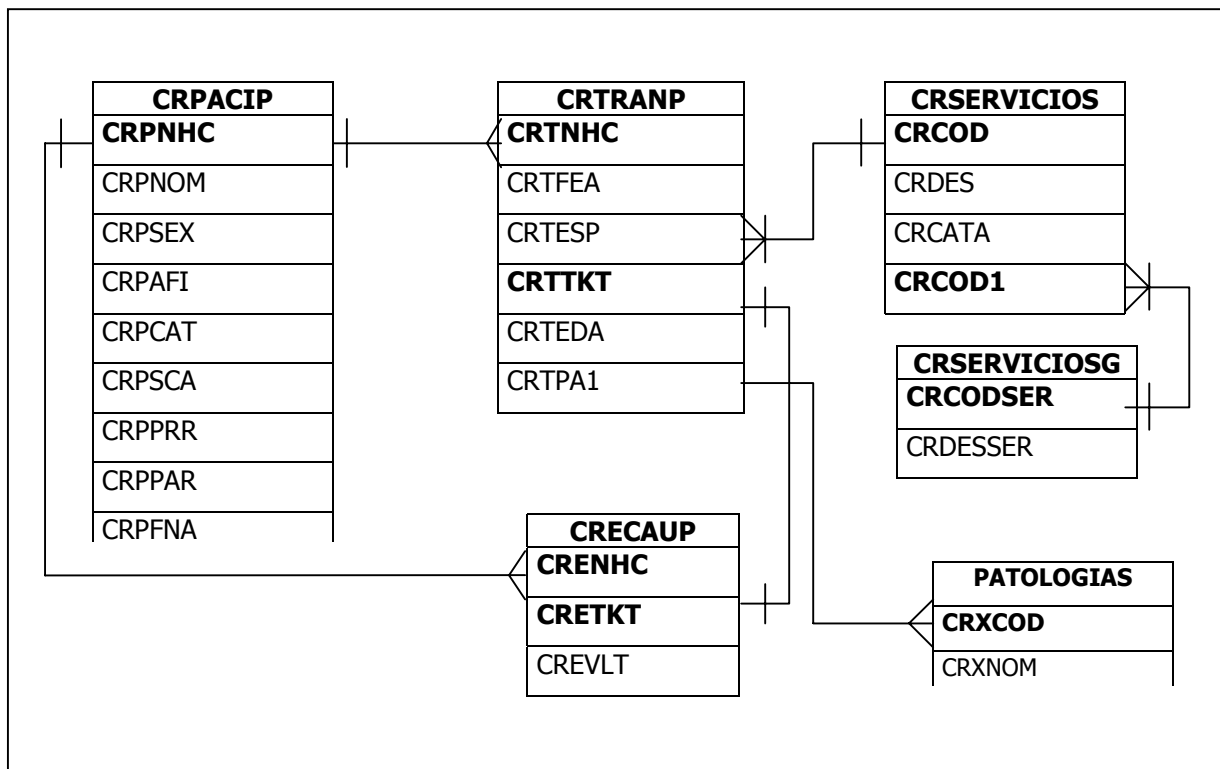
Almacena fecha inicial de subconsulta de datos de CRTRANP Y CRECAUP.

CRSERVICIOSG	
CAMPO	DESCRIPCIÓN
FECHAI	FECHA INICIAL DE CONSULTA
NAMECENTRO	NOMBRE DEL CENTRO MEDICO

Tabla donde se almacenan la fecha inicial que sirve como parámetro en las subconsultas de las tablas CRTRANP y CRECAUP. La información se obtiene del SIG en la Opción Cargar datos/Fecha Inicial.

Además, en esta tabla se ha almacenado un campo que contiene el nombre del centro de rehabilitación donde funcionará el Sistema Gerencial.

Diagrama Entidad- Relación



NOTA: Para el correcto funcionamiento del sistema, las tablas deben tener los nombres que se mencionan y contener los campos indicados, caso contrario dará error.

VISTAS DE BASE DE DATOS QUE UTILIZA EL SISTEMA

A parte de las tablas mencionadas anteriormente, se deben tener dentro de la base de datos las siguientes Vistas:

VISTA VISTACRTRANP

Es una vista de suma importancia para el sistema, es la vista que mayormente se accesa por el sistema, su utilidad radica en que aquí se obtiene información acomodada para obtener los resultados transformados para el sistema, principalmente es donde se obtiene el campo CRTRAN donde se almacena el rango de edad a la cual pertenece el paciente el momento que se ha efectuado la atención, otro campo es CRTESP1 donde se calcula el servicio al que pertenece cada especialidad que atiende, el sexo del paciente también de almacena en el campo CRTSEX. El código de la vista es el siguiente:

```
CREATE VIEW VISTACRTRANP(CRTNHC,
CRTFEA,CRTESP,CRTSEX,CRPSCA,CRTESP1,CRTRAN)AS
SELECT
CRTNHC,CRTFEA,CRTESP,CRPSEX,CRPSCA,ROUND(CRTESP/10000,1,0),
```

CASE

```
WHEN CRTEDA>=0 AND CRTEDA<=6 THEN 'R1:0-6'  
WHEN CRTEDA>6 AND CRTEDA<=12 THEN 'R2:6-12'  
WHEN CRTEDA>12 AND CRTEDA<=18 THEN 'R3:12-18'  
WHEN CRTEDA>18 THEN 'R4:>18'  
ELSE 'R5:SE'  
END AS CRTRAN  
FROM CRTRANP, CRPACIP  
WHERE CRTNHC=CRPNHC
```

VISTA EXONERADOS

Es una vista que sirve para obtener todos los paciente exonerados del centro, para poder obtener la información en el sistema referente a este grupo de pacientes del centro. Su código de creación es el siguiente:

```
CREATE VIEW VEXONERADOS(CRPNHC, CRPNOM)AS  
SELECT CRPNHC, CRPNOM  
FROM CRPACIP  
WHERE CRPCAT ='E'
```

Las siguientes tres vistas son necesarias para el calculo de las proyecciones dentro del sistema, su función radica principalmente en que el campo fecha de atención dentro de la tabla CRTRANP es de tipo numérico, y para hacer la clasificación del número de atenciones por meses es necesario que se tenga el campo de la fecha de atención de tipo DATE(fecha) para obtener los resultados requeridos.

VISTA PROY

Es necesario al campo fecha de atención CRTFEA transformarla a tipo carácter.

```
CREATE VIEW VISTAPROY (CRTKTV, CRTFEAV)AS  
SELECT CRTTKT,CONVERT(CHAR(8),CRTFEA)  
FROM CRTRANP
```

VISTA PROY1

Una vez transformado este campo a carácter s e debe obtener solo los seis caracteres de la derecha de campo CRTFEAV ya que la función para convertir a tipo DATE sufría un desbordamiento con los ocho caracteres de la fecha original.

```
CREATE VIEW VISTAPROY1 (CRTKTV, CRTFEAV)AS  
SELECT CRTKTV,SUBSTRING(CRTFEAV,3,6)  
FROM VISTAPROY
```

VISTAPROYECCION

Se utiliza la función CONVERT del SQL-Transact para transformar la fecha de atención a tipo DATE.

```
CREATE VIEW VISTAPROYECCION (CRTKTV, CRTFEAV)  
AS SELECT CRTKTV, CONVERT(DATETIME,CRTFEAV)  
FROM VISTAPROY1
```

CÓDIGO FUENTE DEL SISTEMA GERENCIAL

La herramienta que se utilizó para la implementación del Sistema Gerencial es Borland Delphi 3.1 Cliente/ Servidor.

FORMA ACTUALIZAR BASE DE DATOS

```
var
  FActualizarbdd: TActualizarbdd;
  anio, mes, dia: word;
  anioc, mesc, diac, fechac: string;
implementation

uses UImpresion, fmxutils, unit1;

{$R *.DFM}

procedure TActualizarbdd.BitBtn1Click(Sender: TObject);
begin
  if op1.Checked = True then
  begin
    if DateTimePicker1.Date <= Date then
    begin
      Decodedate(DateTimePicker1.Date,anio,mes,dia);
      anioc :=inttostr(anio);
      diac:=transfecha(dia);
      mesc:=transfecha(mes);
      fechac:= anioc+mesc+diac;
      Table1.Active:=True;
      with Table1 do
      begin
        Table1.Edit;
        DBEdit1.text:=fechac;
        Table1.Post;
        end;//del with
        Table1.Close;
        Showmessage('Se ha actualizado la fecha, pulse Aceptar para proceder a la Transferencia');
        ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NBTABLAS
        /MSA','C:\SISTEMAGERENCIALCRM\SQL\','SW_SHOW');
        ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRTRANP
        /MSA','C:\SISTEMAGERENCIALCRM\SQL\','SW_SHOW');
        Showmessage('Tabla CRTRANP Transferida');
        ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRECAUP
        /MSA','C:\SISTEMAGERENCIALCRM\SQL\','SW_SHOW');
        Showmessage('Tabla CRECAUP Transferida');
        ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRPACIP
        /MSA','C:\SISTEMAGERENCIALCRM\SQL\','SW_SHOW');
        Showmessage('Tabla CRPACIP Transferida');
        ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRTSERP
        /MSA','C:\SISTEMAGERENCIALCRM\SQL\','SW_SHOW');
        Showmessage('Tabla CRTSERP Transferida');
        ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NPRED
        /MSA','C:\SISTEMAGERENCIALCRM\SQL\','SW_SHOW');
        ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NPREDSER
        /MSA','C:\SISTEMAGERENCIALCRM\SQL\','SW_SHOW');
```



```
Showmessage('Transferencia Ejecutada');
end
else
Showmessage('Verifique la Fecha');
end;
if op2.checked = True then
begin
ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NBTABLAS
/MSA','C:\SISTEMAGERENCIALCRM\SQL\,SW_SHOW);
ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRTRANPT
/MSA','C:\SISTEMAGERENCIALCRM\SQL\,SW_SHOW);
Showmessage('Tabla CRTRANP Transferida');
ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRECAUPT
/MSA','C:\SISTEMAGERENCIALCRM\SQL\,SW_SHOW);
Showmessage('Tabla CRECAUP Transferida');
ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRPACIP
/MSA','C:\SISTEMAGERENCIALCRM\SQL\,SW_SHOW);
Showmessage('Tabla CRPACIP Transferida');
ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NTCRTSERP
/MSA','C:\SISTEMAGERENCIALCRM\SQL\,SW_SHOW);
Showmessage('Tabla CRTSERP Transferida');
ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NPRED
/MSA','C:\SISTEMAGERENCIALCRM\SQL\,SW_SHOW);
ExecuteFile('dtsrun.exe','/S'+DBEdit2.EditText+ ' /USA /PSA /NPREDSER
/MSA','C:\SISTEMAGERENCIALCRM\SQL\,SW_SHOW);
Showmessage('Transferencia Ejecutada');
end;
end;
```

```
procedure TFActualizarbdd.BitBtn2Click(Sender: TObject);
begin
Query2.Close;
close;
end;
```

```
procedure TFActualizarbdd.FormActivate(Sender: TObject);
begin
FActualizarbdd.Height:= 400;
FActualizarbdd.Left:= 177;
FActualizarbdd.Top:= 67;
FActualizarbdd.Width:=520;
```

```
DateTimePicker1.Enabled:=True;
Decodedate(Date,anio,mes,dia);
if mes<>1 then
DatettimePicker1.Date:=strtodate('01/'+inttostr(mes-1)+'/'+inttostr(anio))
else
DatettimePicker1.Date:=strtodate('01/12/'+inttostr(anio-1));
```

```
Query2.Active:= True;
if FPrincipal.DBEdit1.EditText = '' then
begin
label1.visible:= False;
label2.visible:= False;
Label4.visible:=True;
Label4.caption:= 'Base de Datos Vacía ';
end
else
```

```
begin
label6.caption:= fechamin;
label5.caption:= fechamax;
end;
end;

procedure TFActualizarbdd.op2Click(Sender: TObject);
begin
DateTimePicker1.Enabled:= False;
end;

procedure TFActualizarbdd.op1Click(Sender: TObject);
begin
DateTimePicker1.Enabled:= True;
end;
```

FORMA FASexoEdad

```
var
  FASexoEdad: TFASexoEdad;
  fechai, fechaf :string;
  dia, mes, anio,dia1,mes1,anio1:word;
  diac, mesc, anioc,dia1c,mes1c,anio1c:string;
  condicion:string;
  num:integer;

implementation

uses UImpresion, Unit23;
{$R *.DFM}

procedure TFASexoEdad.FormActivate(Sender: TObject);
begin
Decodedate(Date,anio,mes,dia);
DatetimePicker3.Date:=strtodate('01/'+inttostr(mes)+'/'+inttostr(anio-1));
DatetimePicker4.Date:=Date;
label1.caption:="";
end;

procedure TFASexoEdad.BitBtn2Click(Sender: TObject);
begin
if DateTimePicker3.Date <= DateTimePicker4.Date then
begin
Animate1.Active:=True;
StatusBar1.SimpleText:='Buscando Información';
Decodedate(DateTimePicker3.Date,anio,mes,dia);
Decodedate(DateTimePicker4.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
condicion:='WHERE CRTFEA>= '+fechai+' AND CRTFEA<=' + fechaf ;
DecisionCube1.DataSet:=nil;
DecisionQuery1.Close;
```

```
DecisionQuery1.SQL.Clear;
DecisionQuery1.SQL.Add('SELECT CRTRAN,CRTSEX, COUNT(CRTRAN),COUNT(CRTSEX) FROM
VISTACRTRANP ' +
condicion + ' GROUP BY CRTRAN, CRTSEX ');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>0 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
label1.caption:='En el Periodo: '+datetostr(DateTimePicker3.Date)+' a '
+datetostr(DateTimePicker4.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales por Sexo y Edades';
BitBtn3.Enabled:=True;
BitBtn4.Enabled:=True;
end
else
begin
label1.caption:='';
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales por Sexo y Edades';
showmessage('No Hay Datos Que Mostrar');
end;
end
else//if de arriba
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;

procedure TFASexoEdad.BitBtn3Click(Sender: TObject);
begin
Impresion(Decisiongraph1,PrinterSetupDialog1);
end;

procedure TFASexoEdad.BitBtn4Click(Sender: TObject);
begin
RASexoEdad.QuickRep1.Preview;
end;
```

FORMA FPServicios

```
var
FPServicios: TFPservicios;
fechai, fechaf :string;
dia, mes, anio,dia1,mes1,anio1:word;
diac, mesc, anioc,dia1c,mes1c,anio1c:string;
condicion:string;
num:integer;
seleccion:string;
implementation

uses UImpresion, Unit5;
{$R *.DFM}

procedure TFPservicios.BitBtn2Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
begin
Animate1.Active:=True;
```

```
label3.caption:= 'SERVICIOS BRINDADOS';
StatusBar1.SimpleText:='Buscando Información';
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
seleccion:= InttoStr(SpinEdit1.Value);
DecisionCube1.DataSet:=nil;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
condicion:='WHERE CRTNHC = ' + seleccion + ' AND CRTFEA>= '+fechai+' AND CRTFEA<='+
fechaf ;
DecisionQuery1.SQL.Add('SELECT CRTESP1,COUNT(CRTESP1) FROM VISTACRTRANP ' +
condicion + ' GROUP BY CRTESP1 ORDER BY CRTESP1');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>1 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
label1.caption:='En el Período: '+datetostr(DateTimePicker1.Date)+' a '
+datetostr(DateTimePicker2.Date)+ ' Al Paciente '+ seleccion;
Animate1.Active:=False;
StatusBar1.SimpleText:='Terapias y Servicios Brindados a los Pacientes';
Query1.Close;
Query1.SQL.Clear;
Query1.Sql.Add('SELECT DISTINCT CRTESP1 AS CODIGO,CRDESSER AS DESCRIPCION FROM
VISTACRTRANP, CRSERVICIOSG '+ condicion+' AND CRTESP1= CRCODSER');
Query1.Open;
DBGrid1.DataSource:=DataSource1;
BitBtn3.Enabled:=True;
BitBtn4.Enabled:=True;
end
else
begin
label1.caption:="";
Query1.close;
Animate1.Active:=False;
StatusBar1.SimpleText:='Terapias y Servicios Brindados a los Pacientes';
showmessage('No Hay Datos Que Mostrar');
end;
end
else//if de arriba
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;
```

FORMA FATerapias2

```
var
  FATerapias2: TFATerapias2;
  fechai, fechaf :string;
  dia, mes, anio,dia1,mes1,anio1:word;
  diac, mesc, anioc,dia1c,mes1c,anio1c:string;
```

```
condicion:string;  
num:integer;  
codser, codser1, codesp:string;
```

```
implementation  
uses UImpresion;  
{ $R *.DFM }
```

```
procedure TFATerapias2.DBGrid1CellClick(Column: TColumn);  
var  
servicio:string;  
servicioi,servicioi1:integer;  
begin  
servicio:=DBGrid1.SelectedField.Text;  
Table1.Active:=True;  
with Table1 do  
begin  
IndexFieldNames:='crdesser';  
SetKey;  
FieldByName('crdesser').AsString:=servicio;  
Gotokey;  
codser:=Table1.FieldByName('crcodser').AsString;  
end;  
Table1.Close;  
servicioi:=strtoint(codser)*10000;  
servicioi1:=(strtoint(codser)+1)*10000;  
codser:=inttostr(servicioi);  
codser1:=inttostr(servicioi1);  
Query2.Close;  
Query2.SQL.Clear;  
Query2.Sql.Add('SELECT CRDESC AS ESPECIALIDAD FROM CRSERVICIOS WHERE  
CRCOD>'+codser+' AND CRCOD<'+codser1);  
Query2.Open;  
DBGrid2.DataSource:=DataSource2;  
end;
```

```
procedure TFATerapias2.BitBtn2Click(Sender: TObject);  
begin  
if DateTimePicker1.Date <= DateTimePicker2.Date then  
begin  
Animate1.Active:=True;  
StatusBar1.SimpleText:='Buscando Información ';  
Decodedate(DateTimePicker1.Date,anio,mes,dia);  
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);  
anioc :=inttostr(anio);  
anio1c:=inttostr(anio1);  
diac:=transfecha(dia);  
mesc:=transfecha(mes);  
dia1c:=transfecha(dia1);  
mes1c:=transfecha(mes1);  
fechai:= anioc+mesc+diac;  
fechaf:= anio1c+mes1c+dia1c;  
DecisionCube1.DataSet:=nil;  
DecisionQuery1.Close;  
DecisionQuery1.SQL.Clear;  
condicion:='WHERE CRTFEA>= '+fechai+' AND CRTFEA<= '+fechaf+' AND CRTESP = '+ codesp;  
DecisionQuery1.SQL.Add('SELECT CRTRAN, COUNT(CRTRAN) FROM VISTACRTRANP ' +  
condicion + ' GROUP BY CRTRAN ORDER BY CRTRAN');
```

```
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>0 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
DecisionGrid1.DecisionSource:=DecisionSource1;
DecisionGraph1.DecisionSource:=DecisionSource1;
label3.Caption:= 'ATENCIONES POR RANGOS DE EDAD DE '+ DBGrid2.SelectedField.Text;
Label1.caption:='En el Período: '+datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones por Rangos de Edad y Servicios';
Bitbtn3.Enabled:=True;
end
else
begin
label1.caption:="";
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones por Rangos de Edad y Servicios';
showmessage('No Hay Datos Que Mostrar');
end;
end
else//if de arriba
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;

procedure TFATerapias2.DBGrid2CellClick(Column: TColumn);
var
servicio:string;
begin
servicio:=DBGrid2.SelectedField.Text;
Table2.Active:=True;
with Table2 do
begin
IndexFieldNames:='crdesc';
SetKey;
FieldByName('crdesc').AsString:=servicio;
Gotokey;
codesp:=Table2.FieldByName('crcod').AsString;
end;
Table2.Close;
if codesp<>" then
Bitbtn2.Enabled:=True;
end;
end;
```

FORMA FRecaudaciones

```
var
FRecaudaciones: TFRRecaudaciones;
fechai, fechaf :string;
dia, mes, anio,dia1,mes1,anio1:word;
diac, mesc, anioc,dia1c,mes1c,anio1c:string;
condicion, condicion1:string;
num:integer;
a,b :real;

implementation
uses UImpresion;
```

{ \$R *.DFM }

```
procedure TFRecaudaciones.BitBtn1Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
begin
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
DecisionQuery2.Close;
DecisionQuery2.SQL.Clear;
condicion:='WHERE CRETKT IN (SELECT CRTTKT FROM CRTRANP WHERE CRTFEA>= '+ fechai+'
AND CRTFEA<= '+ fechaf +)';
DecisionQuery1.SQL.Add('SELECT SUM(CREVL) AS RECAUDACION_TOTAL_ATENCIONES FROM
CRECAUP ' +
condicion );
DecisionQuery1.Open;
DBGrid1.DataSource:= DataSource1;
condicion1:='WHERE (CREESP = 4010101 OR CREESP = 4010102 ) AND CREFET>= '+ fechai+'
AND CREFET<= '+ fechaf ;
DecisionQuery2.SQL.Add('SELECT SUM(CREVL) AS RECAUDACION_ADMISION FROM CRECAUP '
+
condicion1 );
DecisionQuery2.Open;
DBGrid2.DataSource:=DataSource2;
Label1.Caption:='En el Periodo '+datetostr(DateTimePicker1.Date)+' a '
+datetostr(DateTimePicker2.Date);
a:=strtofloat(DBGrid1.Columns.Items[0].Field.Text);
b:=strtofloat(DBGrid2.Columns.Items[0].Field.Text);
Label4.Caption:=floattostrF(a+b,ffGeneral ,8,1);
end
else
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;
```

FORMA FProyecciones

```
var
FProyecciones: TFProyecciones;
diahoy, meshoy, aniohoy: word;
V: array[1..12] of integer; //vector para guardar la media
T: array[1..6] of integer; //totales
C: array[1..6] of Tcolor; //colores
L: array[1..6] of TLabel; //labels
M: array[1..24] of TLabel;
E: array[1..4] of TLabel;
LE: array[1..6] of TLabel;
implementation
```

{ \$R *.DFM }

```
procedure TFProyecciones.FormActivate(Sender: TObject);
var
af, bc: word;
begin
FProyecciones.WindowState:=wsMaximized;
BDatosR.Enabled:=True;
BAnio.Enabled:=False;
BTrim.Enabled:=False;
BProy2.Enabled:=False;
BProy5.Enabled:=False;
PageControl1.ActivePage:=ProyMensuales;
//asigna el numero de columnas y filas a la cuadrícula pagina Mensuales
for af:=1 to StringGrid1.RowCount-1 do
for bc:=1 to StringGrid1.ColCount-1 do
StringGrid1.Cells[bc,af]="";
for af:=1 to StringGrid2.RowCount-1 do
for bc:=1 to StringGrid2.ColCount-1 do
StringGrid2.Cells[bc,af]="";
StringGrid1.ColCount:=6;
StringGrid2.ColCount:=6;
StringGrid1.RowCount:=14;
StringGrid2.RowCount:=6;
Decodedate(Date,aniohoy, meshoy, diahoy);
StringGrid1.Cells[0,0]='MESES/AÑOS';
StringGrid1.Cells[0,1]='ENERO';
StringGrid1.Cells[0,2]='FEBRERO';
StringGrid1.Cells[0,3]='MARZO';
StringGrid1.Cells[0,4]='ABRIL';
StringGrid1.Cells[0,5]='MAYO';
StringGrid1.Cells[0,6]='JUNIO';
StringGrid1.Cells[0,7]='JULIO';
StringGrid1.Cells[0,8]='AGOSTO';
StringGrid1.Cells[0,9]='SEPTIEMBRE';
StringGrid1.Cells[0,10]='OCTUBRE';
StringGrid1.Cells[0,11]='NOVIEMBRE';
StringGrid1.Cells[0,12]='DICIEMBRE';
StringGrid1.Cells[0,13]='TOTALES';
//poner los titulos a cada columna de la cuadrícula de datos
StringGrid1.Cells[1,0]:=inttostr(aniohoy-2);
StringGrid1.Cells[2,0]:=inttostr(aniohoy-1);
StringGrid1.Cells[3,0]:=inttostr(aniohoy);
StringGrid1.Cells[4,0]:=inttostr(aniohoy+1);
StringGrid1.Cells[5,0]:=inttostr(aniohoy+2);
//poner los titulos a cada fila la cuadrícula de datos en pagina Trimestres
StringGrid2.Cells[0,1]='I TRIMESTRE';
StringGrid2.Cells[0,2]='II TRIMESTRE';
StringGrid2.Cells[0,3]='III TRIMESTRE';
StringGrid2.Cells[0,4]='IV TRIMESTRE';
StringGrid2.Cells[0,5]='TOTALES';
//poner los titulos a cada columna la cuadrícula de datos en pagina Trimestres
StringGrid2.Cells[1,0]:=inttostr(aniohoy-2);
StringGrid2.Cells[2,0]:=inttostr(aniohoy-1);
StringGrid2.Cells[3,0]:=inttostr(aniohoy);
StringGrid2.Cells[4,0]:=inttostr(aniohoy+1);
StringGrid2.Cells[5,0]:=inttostr(aniohoy+2);
Edit1.Text:='Hola Señor Director';
end;
```



```
procedure TFProyecciones.BDatosRClick(Sender: TObject);
var
//cargar los datos
num, num1, cont,i,j,k,conty,col: word;
Condicion:String;
anio1,anio2:string;
begin
Animate1.Active:=True;
Edit1.Text:='Espere un momento por favor, hasta ver los datos';
cont:=0; conty:=3;num1:=0;
try
Repeat
DecisionCube1.DataSet:=nil;
anio1:=inttostr(aniohoy-(conty-2));
anio2:=inttostr(aniohoy-(conty-1));
Condicion:='WHERE (CRTFEAV < "01/01/'+anio1+'" ) AND (CRTFEAV > " 01/01/'+anio2+'"');
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
DecisionQuery1.SQL.Add('SELECT CRTFEAV, COUNT(CRTFEAV) FROM VPROYECCION '+
Condicion+ ' GROUP BY CRTFEAV');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num >0 then
begin
DecisionCube1.DataSet:=DecisionQuery1;
if num1<3 then
if DecisionGrid1.ColCount= 14 then
begin
i:=0; j:=0;
for k:=1 to DecisionGrid1.ColCount-2 do
begin
StringGrid1.Cells[cont+1,j+1]:=DecisionGrid1.Cells[j,i];
j:=j+1;
end; //for
cont:=cont+1; conty:=conty-1;num1:=num1+1;
end //if 12
else conty:=0;//if 3

if num1 = 3 then
begin
DecisionCube1.DataSet:=DecisionQuery1;
i:=0; j:=0;
for k:=1 to DecisionGrid1.ColCount-2 do
begin
StringGrid1.Cells[cont+1,j+1]:=DecisionGrid1.Cells[j,i];
j:=j+1;
end;//for
cont:=cont+1; conty:=conty-1;num1:=num1+1;
end;//num1=3
end //if num>0
else
conty:=0;
until conty=0;
if cont>1 then
begin
BDatosR.Enabled:=False;
BProy2.Enabled:=True;
```

```
Animate1.Active:=False;
Edit1.Text:='Datos Cargados de los dos últimos años hasta la fecha actual';
end
else
begin
Animate1.Active:=False;
showmessage('No existen suficientes datos para hacer las proyecciones, por lo menos cargue
datos de dos años anteriores');
end;
except
MessageBeep(0);
MessageDlg('No se puede acceder a la base de Datos', mtError,[mbok],0);
end;// del try
end;
```

```
procedure TFProyecciones.media(Sender: TObject; z:word);
var
a,b,med:integer;
aum:word;
begin
for aum:=1 to 12 do
V[aum]:=0;
for aum:=1 to 12 do
begin
a:=strtoint(StringGrid1.Cells[z,aum]);
b:=strtoint(StringGrid1.Cells[z+1,aum]);
med:=round((a+b)/2);
V[aum]:=med;
end;
end;
```

```
function potencia(base:real; n:word):real;
var
cont:word;
salida:real;
begin
salida:=1;
for cont:=1 to n do
salida:=salida*base;
potencia:=salida;
end;
```

```
procedure TFProyecciones.Proyeccion(Sender: TObject;c:word ; n:word);
var
f: word;
tasa: real;
p:integer;
begin
tasa:=0.07;
for f:=1 to 12 do
if(StringGrid1.Cells[c,f]= "") then
begin
p:=round(V[f]*potencia((1+tasa),n));
StringGrid1.Cells[c,f]:=floattostr(p);
end;
end;
```

```
function TFProyecciones.Total(StringG:TStringGrid; c:word; fi:word;ff:word):string;
```

```
var
suma: integer;
t:word;
begin
suma:=0;
for t:=fi to ff do
suma:=suma+strtoint(StringG.Cells[c,t]);
Total:=inttostr(suma);
end;

procedure TFProyecciones.BProy5Click(Sender: TObject);
var
numpy,tot:word;
c,fi,ff,trim:word;//para proy trimestrales
begin
StringGrid1.ColCount:=9;
StringGrid2.ColCount:=9;
StringGrid1.Cells[6,0]:=inttostr(aniohoy+3);
StringGrid1.Cells[7,0]:=inttostr(aniohoy+4);
StringGrid1.Cells[8,0]:=inttostr(aniohoy+5);
StringGrid2.Cells[6,0]:=inttostr(aniohoy+3);
StringGrid2.Cells[7,0]:=inttostr(aniohoy+4);
StringGrid2.Cells[8,0]:=inttostr(aniohoy+5);
for numpy:=4 to 7 do
begin
media(StringGrid1,numpy);
Proyeccion(StringGrid1,numpy+2,numpy);
end;
for tot:=6 to 8 do
StringGrid1.Cells[tot,13]:=Total(StringGrid1,tot,1,12);
Edit1.text:='Se ha calculado las proyecciones de cinco años futuros';
for c:=1 to StringGrid1.ColCount-1 do
begin
fi:=1; ff:=3;
for trim:= 1 to 4 do
begin
StringGrid2.cells[c,trim]:= total(StringGrid1,c,fi,ff);
fi:=ff+1;
ff:= ff+3;
end;
StringGrid2.cells[c,5]:= total(StringGrid2,c,1,4);
end;
end;

procedure TFProyecciones.BProy2Click(Sender: TObject);
var
numpy,tot:word;
c,fi,ff,trim:word;//para proy trimestrales
begin
for numpy:=1 to 3 do
begin
media(StringGrid1,numpy);
Proyeccion(StringGrid1,numpy+2,numpy);
end;
for tot:=1 to 5 do
StringGrid1.Cells[tot,13]:=Total(StringGrid1,tot,1,12);
BProy5.Enabled:=True;
for c:=1 to StringGrid1.ColCount-1 do
```

```
begin
  fi:=1; ff:=3;
  for trim:= 1 to 4 do
    begin
      StringGrid2.cells[c,trim]:= total(StringGrid1,c,fi,ff);
      fi:=ff+1;
      ff:= ff+3;
    end;
  StringGrid2.cells[c,5]:= total(StringGrid2,c,1,4);
end;
BAnio.Enabled:=true;
BTrim.Enabled:=true;
Edit1.text:='Se ha calculado las proyecciones de dos años futuros';
end;
```

FORMA FAExonerados1

```
var
  FAExonerados1: TFAExonerados1;
  fechai, fechaf :string;
  dia, mes, anio,dia1,mes1,anio1:word;
  diac, mesc, anioc,dia1c,mes1c,anio1c:string;
  condicion:string;
  num:integer;
  seleccion: string;

implementation
Uses UImpresion;
{$R *.DFM}

procedure TFAExonerados1.BitBtn3Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
begin
Animate1.Active:=True;
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
DecisionCube1.DataSet:=nil;
DecisionCube2.DataSet:=nil;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
DecisionQuery2.Close;
DecisionQuery2.SQL.Clear;
if op1.Checked=True then
begin
condicion:=' WHERE CRTNHC= ' + seleccion + ' AND CRTFEA >= '+fechai+ ' AND CRTFEA<= '+
fechaf;
DecisionQuery1.SQL.Add('SELECT CRTESP, COUNT(CRTESP) FROM CRTRANP ' +
condicion + ' GROUP BY CRTESP ORDER BY CRTESP');
DecisionQuery1.Active:=True;
```

```
num:=DecisionQuery1.RecordCount;
if num>1 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
DecisionGrid1.DecisionSource:=DecisionSource1;
DecisionGraph1.DecisionSource:=DecisionSource1;
Label1.caption:= 'En el Período: '+ datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date)+
' (No de Atenciones) al Paciente con NHC= '+seleccion;
Animate1.Active:=False ;
Query2.Close;
Query2.SQL.Clear;
Query2.Sql.Add('SELECT DISTINCT CRTESP AS CODIGO,CRDESC AS ESPECIALIDAD FROM
CRTRANP,CRSERVICIOS'+
' WHERE CRTNHC= '+seleccion+ ' AND CRTESP=CRCOD AND CRTFEA>= '+fechai+ ' AND CRTFEA
<= '+fechaf+ 'ORDER BY CRTESP');
Query2.Open;
DBGrid2.DataSource:=DataSource3;
end
else
begin
Animate1.Active:=False;
Query2.Close;
label1.caption:="";
showmessage('No Hay Datos Que Mostrar');
end;
end;//del if select
if op2.Checked=True then
begin
condicion:= ' WHERE CRTNHC= ' + seleccion + ' AND CRTFEA >= '+fechai+ ' AND CRTFEA<= '+
fechaf+
'AND CRTESP=CRCOD';
DecisionQuery2.SQL.Add(' SELECT CRTESP, SUM(CRCATA) FROM CRTRANP, CRSERVICIOS'
+condicion + ' GROUP BY CRTESP ORDER BY CRTESP ');
DecisionQuery2.Open;
num:=DecisionQuery2.RecordCount;
if num>1 then
begin
DecisionCube2.DataSet:=DecisionQuery2 ;
DecisionGrid1.DecisionSource:=DecisionSource2;
DecisionGraph1.DecisionSource:=DecisionSource2;
Label1.caption:='En el Período: '+ datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date)+
' (Recaudación si fuera Clase A) al Paciente con NHC= '+seleccion;
Animate1.Active:=False ;
Query2.Close;
Query2.SQL.Clear;
Query2.Sql.Add('SELECT DISTINCT CRTESP AS CODIGO,CRDESC AS ESPECIALIDAD FROM
CRTRANP,CRSERVICIOS'+
' WHERE CRTNHC= '+seleccion+ ' AND CRTESP=CRCOD AND CRTFEA>= '+fechai+ ' AND CRTFEA
<= '+fechaf+ 'ORDER BY CRTESP');
Query2.Open;
DBGrid2.DataSource:=DataSource3;
end
else
begin
Animate1.Active:=False ;
Query2.Close;
```

```
label1.caption:="";
showmessage('No Hay Datos Que Mostrar');
end;
end; //if select
end
else//if de todo el bloque
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;
```

```
procedure TFAExonerados1.BitBtn2Click(Sender: TObject);
begin
Impresion(DecisionGraph1,PrinterSetupDialog1);
end;
```

```
procedure TFAExonerados1.DBGrid1CellClick(Column: TColumn);
var
nombre:string;
begin
if DBGrid1.SelectedIndex=0 then
seleccion:=DBGrid1.SelectedField.Text;
if DBGrid1.SelectedIndex=1 then
begin
nombre:=DBGrid1.SelectedField.Text;
Table1.Active:=True;
with Table1 do
begin
IndexFieldNames:='crpnom';
SetKey;
FieldByName('crpnom').AsString:=nombre;
Gotokey;
seleccion:=Table1.FieldByName('crpnhc').AsString;
end;
end;
end;
```

FORMA FAPatologia

```
var
  FAPatologia: TFAPatologia;
  fechai, fechaf :string;
  dia, mes, anio,dia1,mes1,anio1:word;
  diac, mesc, anioc,dia1c,mes1c,anio1c:string;
  condicion:string;
  num:integer;
```

```
implementation
uses UImpresion, Unit42, Unit23;
{$R *.DFM}
```

```
procedure TFAPatologia.FormActivate(Sender: TObject);
begin
Decodedate(Date,anio,mes,dia);
DatetimePicker1.Date:=strtodate('01/'+inttostr(mes)+'/'+inttostr(anio-1));
DatetimePicker2.Date:=Date;
label1.caption:="";
end;
```

```
procedure TFAPatologia.BitBtn2Click(Sender: TObject);
```

```
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then //if arriba
begin
Animate1.Active:=True;
StatusBar1.SimpleText:= 'Buscando Información' ;
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
condicion:='WHERE CRTPA1 IN(SELECT TOP 10 CRTPA1 FROM CRTRANP WHERE CRTFEA>= '+
fechai+ ' AND CRTFEA<= '+ fechaf+
' AND CRTPA1<> 0 GROUP BY CRTPA1 ORDER BY COUNT(CRTPA1) DESC) ';
DecisionCube1.DataSet:=nil;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
DecisionQuery1.SQL.Add('SELECT CRTPA1,CRPSEX, COUNT(CRPSEX) FROM CRTRANP, CRPACIP '
+
condicion + ' AND CRTNHC=CRPNHC AND CRTFEA>= '+fechai+ ' AND CRTFEA<= '+ fechaf+
'GROUP BY CRTPA1,CRPSEX ORDER BY CRTPA1 ');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>0 then
begin
DecisionCube1.DataSet:=DecisionQuery1;
Label1.caption:='En el Periodo: '+datetostr(DateTimePicker1.Date)+' a '+
datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:= 'Atenciones de las patologias más Frecuentes' ;
BitBtn3.Enabled:=True;
BitBtn4.Enabled:=True;
Query1.Close;
Query1.SQL.Clear;
Query1.Sql.Add('SELECT CRTPA1 AS CODIGO, CRTPDE AS DESCRIPCION FROM CRTRANP,
CRPACIP, CRTPATP'+ ' WHERE CRTPA1 IN(SELECT TOP 10 CRTPA1 FROM CRTRANP'+ ' WHERE
CRTFEA>= '+ fechaf+ ' AND CRTFEA<=' +fechai+ ' AND CRTPA1<> 0 GROUP BY CRTPA1'+
' ORDER BY COUNT(CRTPA1) DESC) AND CRTNHC=CRPNHC AND CRTFEA>= '+
fechai+
' AND CRTFEA<=' +fechai+ ' AND CRTPCO = CRTPA1'+
' GROUP BY CRTPA1,CRTPDE ORDER BY CRTPA1');
Query1.Open;
DBGrid1.DataSource:=DataSource1;
end
else
begin
label1.caption:="";
Animate1.Active:=False;
StatusBar1.SimpleText:= 'Atenciones de las patologias más Frecuentes' ;
showmessage('No Hay datos que mostrar');
Query1.close;
end;
end
else//if de arriba
```

```
showmessage('El periodo para la consulta es inadecuado');  
end;
```

FORMA FAExonerados2

```
var
```

```
FAExonerados2: TFAExonerados2;  
fechai, fechaf :string;  
dia, mes, anio,dia1,mes1,anio1:word;  
diac, mesc, anioc,dia1c,mes1c,anio1c:string;  
condicion:string;  
num:integer;
```

```
implementation
```

```
{ $R *.DFM }
```

```
Uses UImpresion, Unit36;
```

```
procedure TFAExonerados2.BitBtn1Click(Sender: TObject);  
begin  
if DateTimePicker1.Date <= DateTimePicker2.Date then  
begin  
Animate1.Active:=True;  
StatusBar1.SimpleText:='Buscando Informacion ';  
Decodedate(DateTimePicker1.Date,anio,mes,dia);  
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);  
anioc :=inttostr(anio);  
anio1c:=inttostr(anio1);  
diac:=transfecha(dia);  
mesc:=transfecha(mes);  
dia1c:=transfecha(dia1);  
mes1c:=transfecha(mes1);  
fechai:= anioc+mesc+diac;  
fechaf:= anio1c+mes1c+dia1c;  
DecisionCube1.DataSet:=nil;  
DecisionCube2.DataSet:=nil;  
DecisionQuery1.Close;  
DecisionQuery1.SQL.Clear;  
DecisionQuery2.Close;  
DecisionQuery2.SQL.Clear;  
if op1.Checked=True then  
begin  
condicion:='WHERE CRTNHC IN (SELECT DISTINCT CRTNHC FROM VISTACRTRANP,  
VEXONERADOS ' +  
'WHERE CRTNHC =CRPNHC AND CRTFEA >= '+fechai+ ' AND CRTFEA<= '+ fechaf +  
) AND CRTFEA >= '+fechai+ ' AND CRTFEA<= '+ fechaf ;  
DecisionQuery1.SQL.Add('SELECT CRTESP1, COUNT(CRTESP1) FROM VISTACRTRANP ' +  
condicion + ' GROUP BY CRTESP1 ORDER BY CRTESP1');  
DecisionQuery1.Open;  
num:=DecisionQuery1.RecordCount;  
if num>0 then  
begin  
DecisionCube1.DataSet:=DecisionQuery1 ;  
DecisionGrid1.DecisionSource:=DecisionSource1;  
DecisionGraph1.DecisionSource:=DecisionSource1;  
DecisionGraph1.Title.Text.Text:='No de Atenciones de Exonerados';  
Label1.caption:='En el Período: '+datetostr(DateTimePicker1.Date)+' a  
' +datetostr(DateTimePicker2.Date);
```



```
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales a Exonerados ';
BitBtn2.Enabled:=True;
BitBtn4.Enabled:=True;
Query1.Close;
Query1.SQL.Clear;
Query1.Sql.Add('SELECT DISTINCT CRTESP1 AS CODIGO,CRDESSER AS SERVICIO FROM
VISTACRTRANP,CRSERVICIOSG'+
' WHERE CRTESP1=CRCODSER AND CRTFEA>= '+fechai+ ' AND CRTFEA <= '+fechaf+ 'ORDER
BY CRTESP1');
Query1.Open;
DBGrid1.DataSource:=DataSource1;
end
else
begin
Animate1.Active:=False;
label1.caption:="";
Query1.Close;
StatusBar1.SimpleText:='Atenciones Generales a Exonerados ';
showmessage('No Hay Datos Que Mostrar');
end;
end;//del if select
if op2.Checked=True then
begin
condicion:='WHERE CRTNHC IN (SELECT DISTINCT CRTNHC FROM VISTACRTRANP,
VEXONERADOS ' +
'WHERE CRTNHC =CRPNHC AND CRTFEA >= '+fechai+ ' AND CRTFEA<= '+ fechaf +
') AND CRTFEA >= '+fechai+ ' AND CRTFEA<= '+ fechaf + ' AND CRTESP=CRCOD ' ;
DecisionQuery2.SQL.Add('SELECT CRTESP1, SUM(CRCATA) FROM VISTACRTRANP, CRSERVICIOS '
+
condicion + ' GROUP BY CRTESP1 ORDER BY CRTESP1');
DecisionQuery2.Open;
num:=DecisionQuery2.RecordCount;
if num>0 then
begin
DecisionCube2.DataSet:=DecisionQuery2 ;
DecisionGrid1.DecisionSource:=DecisionSource2;
DecisionGraph1.DecisionSource:=DecisionSource2;
DecisionGraph1.Title.Text.Text:='Representacion de Recaudación de Exonerados con costo A';
Label1.caption:='En el Período: '+datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales a Exonerados ';
BitBtn2.Enabled:=True;
BitBtn4.Enabled:=False;
Query1.Close;
Query1.SQL.Clear;
Query1.Sql.Add('SELECT DISTINCT CRTESP1 AS CODIGO,CRDESSER AS SERVICIO FROM
VISTACRTRANP,CRSERVICIOSG'+
' WHERE CRTESP1=CRCODSER AND CRTFEA>= '+fechai+ ' AND CRTFEA <= '+fechaf+ 'ORDER
BY CRTESP1');
Query1.Open;
DBGrid1.DataSource:=DataSource1;
end
else
begin
label1.caption:="";
Query1.Close;
```

```
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales a Exonerados ';
showmessage('No Hay Datos Que Mostrar');
end;
end; //if select
end
else//if de arriba
showmessage('El periodo para la consulta es inadecuado');
end;
```

FORMA FAEspecialidadSexo

```
var
  FAEspecialidadSexo: TFAEspecialidadSexo;
  fechai, fechaf :string;
  dia, mes, anio,dia1,mes1,anio1:word;
  diac, mesc, anioc,dia1c,mes1c,anio1c:string;
  condicion:string;
  num:integer;
  implementation
Uses UImpresion, Unit39, Unit3;
{$R *.DFM}

procedure TFAEspecialidadSexo.BitBtn2Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
begin
Animate1.Active:=True;
StatusBar1.SimpleText:='Buscando Información';
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
DecisionCube1.DataSet:=nil;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
condicion:='WHERE CRTFEA>= '+fechai+' AND CRTFEA<='+ fechaf ;
DecisionQuery1.SQL.Add('SELECT CRTESP1,COUNT(CRTESP1),CRTSEX FROM VISTACRTRANP '
+condicion + ' GROUP BY CRTESP1, CRTSEX ORDER BY CRTESP1, CRTSEX ');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>0 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
label1.caption:='En el Periodo: '+datetostr(DateTimePicker1.Date)+' a '
+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales por Especialidad y Sexo ';
BitBtn3.Enabled:=True;
BitBtn4.Enabled:=True;
Query1.Close;
Query1.SQL.Clear;
```

```
Query1.Sql.Add('SELECT DISTINCT CRTESP1 AS CODIGO,CRDESSER AS SERVICIO FROM
VISTACRTRANP,CRSERVICIOSG'+
' WHERE CRTESP1=CRCODSER AND CRTFEA>= '+fechai+ ' AND CRTFEA <= '+fechaf+ 'ORDER
BY CRTESP1');
Query1.Open;
DBGrid1.DataSource:=DataSource1;
end
else
begin
label1.caption:="";
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales por Especialidad y Sexo ';
Query1.Close;
showmessage('No Hay Datos Que Mostrar');
end;
end
else//if de arriba
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;
```

FORMA FAEspecialidadEdades

```
var
FAEspecialidadEdades: TFAEspecialidadEdades;
fechai, fechaf :string;
dia, mes, anio,dia1,mes1,anio1:word;
diac, mesc, anioc,dia1c,mes1c,anio1c:string;
condicion:string;
num:integer;

implementation

uses Unit41, UImpresion, Unit3;

{$R *.DFM}
procedure TFAEspecialidadEdades.FormActivate(Sender: TObject);
begin
Decodedate(Date,anio,mes,dia);
DateTimePicker1.Date:=strtodate('01/'+inttostr(mes)+'/'+inttostr(anio-1));
DateTimePicker2.Date:=Date;
label1.caption:="";
FAEspecialidadEdades.Left:= 61;
FAEspecialidadEdades.Top:= 86;
FAEspecialidadEdades.Height:= 470;
FAEspecialidadEdades.Width:= 696;
end;

procedure TFAEspecialidadEdades.BitBtn2Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
begin
Animate1.Active:=True;
StatusBar1.SimpleText:='Buscando Información';
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
```

```
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
DecisionCube1.DataSet:=nil;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
condicion:='WHERE CRTFEA>= '+fechai+' AND CRTFEA<='+ fechaf ;
DecisionQuery1.SQL.Add('SELECT CRTESP1,COUNT(CRTESP1),CRTRAN FROM VISTACRTRANP '
+condicion + ' GROUP BY CRTESP1, CRTRAN ORDER BY CRTESP1, CRTRAN ');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>0 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
label1.caption:='En el Periodo: '+datetostr(DateTimePicker1.Date)+' a '
+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones por Especialidad y Rangos de Edad';
BitBtn4.Enabled:=True;
Query1.Close;
Query1.SQL.Clear;
Query1.Sql.Add('SELECT DISTINCT CRTESP1 AS CODIGO,CRDESSER AS SERVICIO FROM
VISTACRTRANP,CRSERVICIOSG'+
' WHERE CRTESP1=CRCODSER AND CRTFEA>= '+fechai+' AND CRTFEA <= '+fechaf+' ORDER
BY CRTESP1');
Query1.Open;
DBGrid1.DataSource:=DataSource1;
end
else
begin
label1.caption:="";
Query1.Close;
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones por Especialidad y Rangos de Edad';
showmessage('No Hay Datos Que Mostrar');
end;
end
else//if de arriba
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;
```

FORMA FAMensuales

```
var
  FAMensuales: TFAMensuales;
  fechai, fechaf :string;
  dia, mes, anio:word;
  dia1,mes1,anio1:word;
  condicion:string;
  num:integer;

implementation
uses UImpresion, Unit44;
{$R *.DFM}

procedure TFAMensuales.FormActivate(Sender: TObject);
begin
Decodedate(Date,anio,mes,dia);
```

```
DatetimePicker1.Date:=strtodate('01/'+inttostr(mes)+'/'+inttostr(ano-1));
DatetimePicker2.Date:=Date;
end;

procedure TFAMensuales.BitBtn1Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
begin
Animate1.Active:=True;
StatusBar1.SimpleText:='Buscando Información';
fechai:=Datetostr(DateTimePicker1.Date);
fechaf:=Datetostr(DateTimePicker2.Date);
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
condicion:=' WHERE (CRTFEAV <= "'+ fechaf + '" ) AND (CRTFEAV >= "' + fechai + '" )';
DecisionQuery1.SQL.Add('SELECT CRTFEAV,COUNT(CRTFEAV) FROM VPROYECCION ' +
condicion + ' GROUP BY CRTFEAV ');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>0 then
begin
label1.caption:='En el Periodo: '+datetostr(DateTimePicker1.Date)+' a '
+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales por Meses';
BitBtn2.Enabled:=True;
BitBtn4.Enabled:=True;
end
else
begin
label1.caption:='';
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones Generales por Meses ';
showmessage('No Hay Datos Que Mostrar');
end;
end
else//if de arriba
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;
```

FORMA FAExonerados

```
var
FAExonerados: TFAExonerados;
fechai, fechaf :string;
dia, mes, anio,dia1,mes1,anio1:word;
diac, mesc, anioc,dia1c,mes1c,anio1c:string;
condicion:string;
num:integer;

implementation

{$R *.DFM}
Uses UImpresion,Unit4;

procedure TFAExonerados.BitBtn3Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
```

```
begin
Animate1.Active:=True;
StatusBar1.SimpleText:='Buscando Informacion ';
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
DecisionCube1.DataSet:=nil;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
condicion:=' WHERE CRTNHC IN (SELECT DISTINCT CRTNHC FROM VISTACRTRANP,
VEXONERADOS ' +
'WHERE CRTNHC = CRPNHC AND CRTFEA >= '+ fechai + ' AND CRTFEA<= '+ fechaf +')AND
CRTFEA >= '+ fechai + ' AND CRTFEA<= '+ fechaf ;
DecisionQuery1.SQL.Add('SELECT CRPSCA, COUNT(CRPSCA) FROM VISTACRTRANP ' +
condicion + ' GROUP BY CRPSCA ORDER BY CRPSCA');
DecisionQuery1.Active:=True;
num:=DecisionQuery1.RecordCount;
if num>1 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
DecisionGrid1.DecisionSource:=DecisionSource1;
DecisionGraph1.DecisionSource:=DecisionSource1;
Label1.caption:= 'En el Período: '+ datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date);
Animate1.Active:=False ;
StatusBar1.SimpleText:='Atenciones por Exonerados ';
BitBtn2.Enabled:=True;
BitBtn4.Enabled:=True;
ListBox1.Visible:=True;
end
else
begin
Animate1.Active:=False;
StatusBar1.SimpleText:='Numero de Atenciones a Pacientes Exonerados Agrupados por Categorías
';
label1.caption:="";
ListBox1.Visible:=False;
showmessage('No Hay Datos Que Mostrar');
end;
end;
end;
```

FORMA FATERAPIAS1

```
var
  FATerapias1: TFATerapias1;
  fechai, fechaf :string;
  dia, mes, anio,dia1,mes1,anio1:word;
  diac, mesc, anioc,dia1c,mes1c,anio1c:string;
  condicion, codigoser, codigoser1:string;
  num:integer;
```

implementation

```
uses UImpresion, Unit3;  
{ $R *.DFM }
```

```
procedure TFATerapias1.FormActivate(Sender: TObject);  
begin  
Decodedate(Date,anio,mes,dia);  
DateTimePicker1.Date:=strtodate('01/'+inttostr(mes)+'/'+inttostr(anio-1));  
DateTimePicker2.Date:=Date;  
label1.caption:="";  
label3.caption:="";  
Query1.Active:=True;  
end;
```

```
procedure TFATerapias1.BitBtn1Click(Sender: TObject);  
begin  
DecisionQuery1.Close;  
DecisionQuery2.Close;  
Query1.Close;  
Query2.Close;  
Close;  
end;
```

```
procedure TFATerapias1.BitBtn2Click(Sender: TObject);  
begin  
if DateTimePicker1.Date <= DateTimePicker2.Date then  
begin  
Animate1.Active:=True;  
StatusBar1.SimpleText:='Buscando Información ';  
Decodedate(DateTimePicker1.Date,anio,mes,dia);  
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);  
anioc :=inttostr(anio);  
anio1c:=inttostr(anio1);  
diac:=transfecha(dia);  
mesc:=transfecha(mes);  
dia1c:=transfecha(dia1);  
mes1c:=transfecha(mes1);  
fechai:= anioc+mesc+diac;  
fechaf:= anio1c+mes1c+dia1c;  
DecisionCube1.DataSet:=nil;  
DecisionCube2.DataSet:=nil;  
DecisionQuery1.Close;  
DecisionQuery1.SQL.Clear;  
DecisionQuery2.Close;  
DecisionQuery2.SQL.Clear;  
if op1.Checked=True then  
begin  
condicion:=' WHERE CRTFEA >= '+fechai+ ' AND CRTFEA<= '+ fechaf +  
' AND CRTESP>= '+codigoser+ ' AND CRTESP< '+codigoser1;  
DecisionQuery1.SQL.Add('SELECT CRTESP, COUNT(CRTESP) FROM CRTRANP ' +  
condicion + ' GROUP BY CRTESP ORDER BY CRTESP');  
DecisionQuery1.Open;  
num:=DecisionQuery1.RecordCount;  
if num>1 then  
begin  
DecisionCube1.DataSet:=DecisionQuery1 ;  
DecisionGrid1.DecisionSource:=DecisionSource1;
```

```
DecisionGraph1.DecisionSource:=DecisionSource1;
DecisionGraph1.Title.Text.Text:='Numero de Atenciones de '+ DBGrid1.SelectedField.Text;;
Label3.caption:='Numero de Atenciones de '+ DBGrid1.SelectedField.Text;
Label1.caption:='En el Período: '+datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
BitBtn3.Enabled:=True;
BitBtn4.Enabled:=True;
StatusBar1.SimpleText:='Atenciones y Recaudación '+DBGrid1.SelectedField.Text;
Query2.Close;
Query2.SQL.Clear;
Query2.Sql.Add('SELECT CRCOD AS CODIGO,CRDESC AS ESPECIALIDAD FROM CRSERVICIOS
WHERE CRCOD>='+codigoser+' AND CRCOD<'+codigoser1);
Query2.Open;
DBGrid2.DataSource:=DataSource3;
end
else
begin
Query2.Close;
label1.caption:='';
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones y Recaudación '+DBGrid1.SelectedField.Text;
showmessage('No Hay Datos Que Mostrar');
end;
end;//del if select
if op2.Checked=True then
begin
condicion:=' WHERE CRETKT IN(SELECT CRTTKT FROM CRTRANP WHERE CRTFEA >= '+fechai+
' AND CRTFEA<= '+ fechaf + ' AND CRTESP >= '+codigoser+ ' AND CRTESP < '+codigoser1+')';
DecisionQuery2.SQL.Add(' SELECT CREESP, SUM(CREVLТ) FROM CRECAUP ' +
condicion + ' GROUP BY CREESP ORDER BY CREESP ');
DecisionQuery2.Open;
num:=DecisionQuery2.RecordCount;
if num>1 then
begin
DecisionCube2.DataSet:=DecisionQuery2;
DecisionGrid1.DecisionSource:=DecisionSource2;
DecisionGraph1.DecisionSource:=DecisionSource2;
DecisionGraph1.Title.Text.Text:='Recaudación Monetaria de '+ DBGrid1.SelectedField.Text;;
Label3.caption:='Recaudación Monetaria de'+ DBGrid1.SelectedField.Text;
Label1.caption:='En el Período: '+datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Atenciones y Recaudación '+ DBGrid1.SelectedField.Text;
Query2.Close;
Query2.SQL.Clear;
Query2.Sql.Add('SELECT CRCOD AS CODIGO,CRDESC AS ESPECIALIDAD FROM CRSERVICIOS
WHERE CRCOD>='+codigoser+' AND CRCOD<'+codigoser1);
Query2.Open;
DBGrid2.DataSource:=DataSource3;
BitBtn3.Enabled:=True;
BitBtn4.Enabled:=True;
end
else
begin
Query2.Close;
label1.caption:='';
Animate1.Active:=False;
```



```
StatusBar1.SimpleText:='Atenciones y Recaudación de '+DBGrid1.SelectedField.Text;
showmessage('No Hay Datos Que Mostrar');
end;
end; //if select
end
else//if de arriba
showmessage('El Periodo Para La Consulta Es Inadecuado');
end;
```

```
procedure TFATerapias1.BitBtn3Click(Sender: TObject);
begin
Impresion(DecisionGraph1,PrinterSetupDialog1);
end;
```

```
procedure TFATerapias1.DBGrid1CellClick(Column: TColumn);
var
servicio, codigo:string;
servicioi,servicioi1:integer;
begin
servicio:=DBGrid1.SelectedField.Text;
Table1.Active:=True;
with Table1 do
begin
IndexFieldNames:='crdesser';
SetKey;
FieldByName('crdesser').AsString:=servicio;
Gotokey;
codigo:=Table1.FieldByName('crodser').AsString;
end;
Table1.Close;
servicioi:=strtoint(codigo)*10000;
servicioi1:=(strtoint(codigo)+1)*10000;
codigoser:=inttostr(servicioi);
codigoser1:=inttostr(servicioi1);
BitBtn2.Enabled:=True;
end;
```

FORMA FActualizarbdd

```
var
FActualizarbdd: TActualizarbdd;
anio, mes, dia: word;
anioc, mesc, diac, fechac: string;
```

```
implementation
uses UImpresion, fmxutils;
{$R *.DFM}
```

```
procedure TActualizarbdd.BitBtn1Click(Sender: TObject);
begin
if op1.Checked = True then
begin
if DateTimePicker1.Date <= Date then
begin
Decodedate(DateTimePicker1.Date,anio,mes,dia);
anioc :=inttostr(anio);
diac:=transfecha(dia);
mesc:=transfecha(mes);
```

```
fechac:= anioc+mesc+diac;
Table1.Active:=True;
with Table1 do
begin
Table1.Edit;
DBEdit1.text:=fechac;
Table1.Post;
end;//del with
Table1.Close;
Showmessage('Se ha actualizado la fecha, pulse Aceptar para proceder a la Transferencia');
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NBTABLAS
/MSA','C:\MSSQL\BINN\',SW_SHOW);
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NTCRTRANP
/MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Tabla CRTRANP TTransferida');
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NTCRECAUP
/MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Tabla CRECAUP TTransferida');
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NTCRPACIP
/MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Tabla CRPACIP TTransferida');
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NPRED /MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Transferencia Ejecutada');
end
else
Showmessage('Verifique la Fecha');
end;
if op2.checked = True then
begin
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NBTABLAS
/MSA','C:\MSSQL\BINN\',SW_SHOW);
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NTCRTRANPT
/MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Tabla CRTRANP TTransferida');
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NTCRECAUPT
/MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Tabla CRECAUP TTransferida');
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NTCRPACIP
/MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Tabla CRPACIP TTransferida');
ExecuteFile('dtsrun.exe','/SDIRECCION /USA /PSA /NPRED /MSA','C:\MSSQL\BINN\',SW_SHOW);
Showmessage('Transferencia Ejecutada');
end;
end;

procedure TFActualizarbdd.op2Click(Sender: TObject);
begin
DateTimePicker1.Enabled:= False;
end;

procedure TFActualizarbdd.op1Click(Sender: TObject);
begin
DateTimePicker1.Enabled:= True;
end;
```

FORMA FPSEXOEDAD

```
var
    FPSexoEdad: TFPSexoEdad;
```

```
fechai, fechaf :string;
dia, mes, anio,dia1,mes1,anio1:word;
diac, mesc, anioc,dia1c,mes1c,anio1c:string;
condicion:string;
num:integer;
```

implementation

```
{ $R *.DFM }
```

Uses UImpresion, Unit37;

```
procedure TFPSexoEdad.BitBtn2Click(Sender: TObject);
begin
if DateTimePicker1.Date <= DateTimePicker2.Date then
begin
Animate1.Active:=True;
StatusBar1.SimpleText:='Buscando Información ';
Decodedate(DateTimePicker1.Date,anio,mes,dia);
Decodedate(DateTimePicker2.Date,anio1,mes1,dia1);
anioc :=inttostr(anio);
anio1c:=inttostr(anio1);
diac:=transfecha(dia);
mesc:=transfecha(mes);
dia1c:=transfecha(dia1);
mes1c:=transfecha(mes1);
fechai:= anioc+mesc+diac;
fechaf:= anio1c+mes1c+dia1c;
DecisionCube1.DataSet:=nil;
DecisionCube2.DataSet:=nil;
DecisionQuery1.Close;
DecisionQuery1.SQL.Clear;
DecisionQuery2.Close;
DecisionQuery2.SQL.Clear;
condicion:='(SELECT DISTINCT CRTNHC, CRTRAN, CRTSEX FROM VISTACRTRANP WHERE
CRTFEA>= ' + fechai + ' AND CRTFEA<= ' + fechaf + ')AS P ';
DecisionQuery1.SQL.Add('SELECT CRTRAN, COUNT(CRTRAN),CRTSEX, COUNT(CRTSEX) FROM ' +
condicion + ' GROUP BY P.CRTRAN, P.CRTSEX ORDER BY P.CRTRAN, P.CRTSEX ');
DecisionQuery1.Open;
num:=DecisionQuery1.RecordCount;
if num>1 then
begin
DecisionCube1.DataSet:=DecisionQuery1 ;
DecisionGrid1.DecisionSource:=DecisionSource1;
DecisionGraph1.DecisionSource:=DecisionSource1;
Label1.caption:='En el Período: '+datetostr(DateTimePicker1.Date)+' a
'+datetostr(DateTimePicker2.Date);
Animate1.Active:=False;
StatusBar1.SimpleText:='Pacientes Clasificados por Sexo Edad ';
BitBtn3.Enabled:=True;
BitBtn4.Enabled:=True;
end
else
begin
label1.caption:='';
Animate1.Active:=False;
StatusBar1.SimpleText:='Pacientes Clasificados por Sexo Edad ';
showmessage('No Hay Datos Que Mostrar');
end;
end
```

```
else//if de arriba  
showmessage('El Periodo Para La Consulta Es Inadecuado');  
end;
```