SEMINARIO DE LENGUAJES – OPCION DELHI – 11/06/07 **Tema I**

Nombre y Apellido .....

1. El Servicio Meteorológico Nacional lleva el registro de las precipitaciones (en mm de agua caída) y las temperaturas (en ºC) ocurridas en las distintas localidades del país. La información se encuentra organizada de la siguiente forma: **(3,50 ptos)**

PROVINCIAS

IdProvincia	Numérico
Nombre	Alfanumérico(40)

LOCALIDADES

IdLocalidad	Autoincremental
Nombre	Alfanumérico(40)
CantHabitantes	Numérico
IdProvincia	Numérico

PRECIPITACIONES

IdPrecipitacion	Numérico
AguaCaida	Numérico
Temperatura	Numérico
Fecha	Date
IdLocalidad	Numérico

Indique la manera de obtener:

a) Los nombres de las localidades y de las provincias ordenados por el nombre de la provincia en forma descendente y dentro de la misma provincia por el nombre de la localidad en forma ascendente. Visualice los nombres de las localidades y provincias en un único ListBox.

```
Select P.nombre as NomProv, L.nombre as NomLoc
from  Localidades L, Provincias P
where (L.idProvincia = P.IdProvincia)
order by P.nombre Desc, L.nombre
```

```
-------------------------------------------------------
Var Loc, Prov : string;
begin
   Query1.close;
   Query1.open;
   ListBox1.items.clear;
   While not Query1.eof do
   Begin
     Loc := Query1.fieldbyname('NomLoc').asString;
     Prov := Query1.fieldbyname('NomProv').asString;

     ListBox1.items.add(Prov + ' - ' + Loc);
     Query1.next;
   End;
End;
```

---

b) El nombre de la provincia que tiene la mayor cantidad de habitantes. Visualice el nombre de la provincia y la cantidad de habitantes que tiene utilizando ShowMessage.

```
Select P.nombre, sum(CantHabitantes) as Cant
from  Localidades L, Provincias P
where (L.idProvincia = P.IdProvincia)
group by P.nombre
order by 2 desc
```

```
-------------------------------------------------------
Var Prov : string;
    TotHab : integer;
Begin
   Query1.close;
```

```
   Query1.open;
   Prov := Query1.fieldbyname('nombre').asString;
   TotHab := Query1.fieldbyname('Cant').asInteger;
   ShowMessage('La provincial '+Prov+' tiene '+intToStr(TotHab)+' hab.');
End;
```

c) El nombre de la localidad donde se registró la temperatura más alta en el mes de mayo de 2007. A igual temperatura, elegir la que tuvo la menor cantidad de agua caída en la fecha más reciente. Visualice el resultado en un componente Label. Además, si pertenece a la provincia con nombre 'Buenos Aires' debe mostrar el mensaje 'Es de Buenos Aires'.

```
Select idProvincia, nombre, temperatura, AguaCaida, fecha
from  Localidades L, Precipitaciones T
where (L.idLocalidad = T.idLocalidad) and
      (fecha>='05/01/07') and (fecha<='05/31/07')
order by temperatura desc, fecha desc, AguaCaida

-------------------------------------------------------

Var Loc : string;
    idProv : integer;
Begin
  Query1.close;
  Query1.open;
  Label1.caption := Query1.fieldbyname('nombre').asString;
  idProv := Query1.fieldbyname('idProvincia').asInteger;

  { TPROVINCIAS es una componente Table que señala la tabla PROVINCIAS }
  TProvincias.Locate('idProvincia',idProv,[]);
  if TProvincias.fieldbyname('nombre').asString = 'Buenos Aires' then
     ShowMessage('Es de Buenos Aires');

  { En lugar del LOCATE puede agregar el nombre de la provincial al
    Query y usarlo para preguntar }
End;
```

d) Los nombres de las provincias (sin repetición) donde se registraron nevadas alguna vez (se registra Agua caída con temperaturas inferiores a 0ºC). Visualice el resultado en un Memo.

```
Select distinct P.nombre
from  Localidades L, Precipitaciones T, Provincias P
where (L.idProvincia = P.IdProvincia) and
      (L.idLocalidad = T.idLocalidad) and
      (AgusCaida > 0) and (temperatura < 0)

   Query1.close;
   Query1.open;
   Memo1.lines.clear;
   While not Query1.eof do
   Begin
     Memo1.Lines.add(Query1.fieldbyname('Nombre').asString);
     Query1.next;
   End;
```

e) Los nombres de las provincias cuya temperatura promedio durante los dos primeros meses de 2007 haya sido superior a los 22 ºC. Utilice un DBLookupListBox para mostrar los nombres de las provincias junto con la temperatura promedio correspondiente.

```
Select P.nombre, avg(Temperatura) as promedio
from  Localidades L, Precipitaciones T, Provincias P
where (L.idLocalidad = T.idLocalidad) and (L.idProvincia = P.idProvincia)
      (fecha>='01/01/07') and (fecha<='02/28/07')
Group by P.nombre
Having avg(temperatura)>22

Query1.close;
Query1.open;
{ DSQuery es el datasource del Query1 }
DBLookipListBox1.LisSource := DSQuery1;
DBLookupLitBox1.KeyField := nombre;
DBLookupListBox1.ListField := 'nombre;promedio'
```

2. La tabla PRECIPITACIONES contiene la temperatura almacenada en °C (grados centígrados o Celsius). Escriba el segmento de código necesario para modificar el contenido de la tabla de manera que este campo quede almacenado en grados Fahrenheit. La fórmula de conversión a utilizar es la siguiente:

°F = °C * 1,8 + 32. **(1,50 ptos)**

```
TPRECIPITACIONES.First;
While not TPRECIPITACIONES.eof do
Begin
    TPRECIPITACIONES.edit;
    Grados := TPRECIPITACIONES.fieldbyname('temperatura').asFloat;
    TPRECIPITACIONES.fieldbyname('temperatura').asFloat:=Grados * 1.8 +32;
    TPRECIPITACIONES.post;
    TPRECIPITACIONES.next;
End;
```

3. Indique la manera de resolver los siguientes problemas. En cada caso mencione las componentes y propiedades utilizadas e implemente el código Delphi correspondiente: **(2 ptos)**
   a. El usuario puede elegir 0,1 o varias opciones de un grupo de 50. Visualizar el texto 'Falta elegir' si ninguna ha sido seleccionada.

```
Se utilizará un CheckListBox con 50 líneas cargadas en su propiedad
ítems. Para saber si no hay ninguna elegida debe utilizarse la
propiedad Checked de cada elemento.

Var i : integer;
Begin
   i:= 0;
   while (i<CheckListBox1.count) and
            not CheckListBox1.checked[i] do
         i:= i+1;

   if i=CheckListBox1.count then
            ShowMessage('Falta elegir');
```

```
end;
```

b. Seleccionar uno de los meses del año. El nombre elegido debe escribirse en un componente Label.

Se utilizará un ComboBox con 12 líneas cargadas en su propiedad ítems (cada una con el nombre de un mes del año). Como respuesta al evento OnClick debe ejecutarse:

```
Label1.caption := ComboBox1.text
```

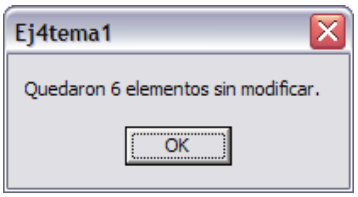
c. Implementar un botón que al ser clickeado se corre 20 pixels (puntos) a la derecha.

En el OnClick del Botón poner: `Button1.left := Button1.Left+20;`

4. Un formulario contiene 4 componentes Edit y un panel. Dentro del panel hay una cierta cantidad de componentes Edit y Labels.

Los 4 componentes Edit, ubicados en la parte superior del formulario, comparten el mismo comportamiento (método). Cualquiera de ellos, al ser clickeado, escribe su contenido en cada uno de los componentes del Panel siempre y cuando sea anterior alfabéticamente. Al final informa la cantidad de elementos del Panel que quedaron sin modificar.

Por ejemplo, la imagen de la derecha muestra el resultado de clickear sobre el Edit que contiene la palabra 'examen'. Dentro del panel quedaron 6 elementos sin modificar porque la palabra 'examen' no es anterior alfabéticamente a sus respectivos contenidos. Escriba el método compartido por los 4 edit del formulario. **(1,50 ptos)**




```
procedure TForm1.Edit1Click(Sender: TObject);
var i,noPudo : integer;
begin
  noPudo := 0;
  for i:=0 to Panel1.controlCount-1 do
  begin
    if Panel1.controls[i] is TEdit then
         begin
          if TEdit(Panel1.controls[i]).text > Tedit(Sender).Text then
                TEdit(Panel1.controls[i]).text := Tedit(Sender).Text
          else  noPudo := noPudo+1;
         end
    else begin
          if TLabel(Panel1.controls[i]).caption > Tedit(Sender).Text then
                TLabel(Panel1.controls[i]).caption := Tedit(Sender).Text
          else  noPudo := noPudo + 1;
         end;
  end;
  ShowMessage('Quedaron '+IntToStr(NoPudo)+' elementos sin modificar.');
end;
```

5. Indicar Verdadero o Falso **(1,50 ptos.)**

(F)	a) En un proceso que define el comportamiento de un evento, el parámetro Source representa a la componente que recibió el evento.
(F)	b) Todos los controles son no visuales.
(F)	c) Para establecer una relación Maestro-Detalle entre dos tablas, ambas deben estar indexadas.
(F)	d) La única forma de que un proceso Drag & Drop se lleve a cabo es que la componente donde se inicia el arrastre tenga en su propiedad DragMode el valor dmAutomatic.
(V)	e) El producto cartesiano ente varias tablas de las cuales al menos una está vacía, es vacío.
(V)	f) La componente ListBox posee un método que permite copiar el contenido de su lista en un RadioGroup.
(V)	g) Al asignar un valor en la propiedad ModalResult de un Formulario, éste se cierra automáticamente.
(F)	h) La invocación de la función StrToFloat con el parámetro '72' produce una excepción.
(F)	i) El método SetKey-GotoKey puede aplicarse sobre una consulta sólo si ésta está ordenada.
(V)	j) Si se asigna un valor en la propiedad KeyValue de un DBLookupListBox, éste puede llegar a posicionarse en otro registro.

SEMINARIO DE LENGUAJES – OPCION DELHI – 11/06/07 **Tema II**

Nombre y Apellido .....

1. El Servicio Meteorológico Nacional lleva el registro de las precipitaciones (en mm de agua caída) y las temperaturas (en ºC) ocurridas en las distintas localidades del país. La información se encuentra organizada de la siguiente forma: **(3,50 ptos)**

PROVINCIAS

IdProvincia	Numérico
Nombre	Alfanumérico(40)

LOCALIDADES

IdLocalidad	Autoincremental
Nombre	Alfanumérico(40)
CantHabitantes	Numérico
IdProvincia	Numérico

PRECIPITACIONES

IdPrecipitacion	Numérico
AguaCaida	Numérico
Temperatura	Numérico
Fecha	Date
IdLocalidad	Numérico

Indique la manera de obtener:

a) Los nombres de las localidades y de las provincias ordenados por el nombre de la provincia en forma descendente y dentro de la misma provincia por cantidad de habitantes de la localidad en forma ascendente. Visualice los nombres de las localidades y provincias en un Memo.

```
Select P.nombre as NomProv, L.nombre as NomLoc
from  Localidades L, Provincias P
where (L.idProvincia = P.IdProvincia)
order by P.nombre Desc, CantHabitantes
```

```
-------------------------------------------------------
Var Loc, Prov : string;
begin
   Query1.close;
   Query1.open;
   Memo1.clear;
   While not Query1.eof do
   Begin
     Loc := Query1.fieldbyname('NomLoc').asString;
     Prov := Query1.fieldbyname('NomProv').asString;

     Memo1.Lines.add(Prov + ' - ' + Loc);
     Query1.next;
   End;
End;
```

---

b) El nombre de la provincia que tiene la temperatura promedio más baja. Visualice el nombre de la localidad y la temperatura promedio utilizando ShowMessage.

```
Select nombre, avg(temperature) as AVGTemp
from  Localidades L, Precipitaciones P
where (L.idLocalidad = T.idLocalidad)
group by nombre
order by 2
```

```
-------------------------------------------------------
Var Loc : string;
    Temp : integer;
Begin
   Query1.close;
   Query1.open;
```

```
   Loc := Query1.fieldbyname('nombre').asString;
   Temp := Query1.fieldbyname('AVGTemp').asInteger;
   ShowMessage('La localidad '+Loc+' tiene una temperatura promedio de
'+FloatToStr(AVGTemp)+' °C');
End;
```

---

c) Los nombres de las provincias que registran al menos 300 mm de agua caída durante el 2006. Utilice un DBLookupListBox para mostrar los nombres de las provincias junto con la cantidad de agua caída durante el 2006.

```
Select P.nombre, sum(AguaCaida) as cant
from  Localidades L, Precipitaciones T
where (L.idLocalidad = T.idLocalidad) and
      (fecha>='01/01/06') and (fecha<='12/31/06')
Group by P.nombre
Having sum(aguaCaida)>=300

Query1.close;
Query1.open;
{ DSQuery es el datasource del Query1 }
DBLookipListBox1.LisSource := DSQuery1;
DBLookupLitBox1.KeyField := nombre;
DBLookupListBox1.ListField := 'nombre;cant'
```

---

d) Los nombres de las localidades (sin repetición) donde no se produjeron lluvias durante el mes de marzo de 2006. Visualice el resultado en un ListBox

```
Select distinct nombre
from  Localidades L, Precipitaciones T
where (L.idLocalidad = T.idLocalidad) and (AguaCaida=0)
       and (fecha>='03/01/06') and (fecha<='03/31/06')

   Query1.close;
   Query1.open;
   ListBox1.clear;
   While not Query1.eof do
   Begin
     ListBox1.items.add(Query1.fieldbyname('Nombre').asString);
     Query1.next;
   End;
```

---

e) El nombre de la localidad donde se registró la mayor cantidad de agua caída en el mes de abril de 2007. A igualdad de mm de agua caída elegir la que tuvo la menor temperatura en la fecha más reciente. Visualice el resultado en un componente Edit. Además, si pertenece a la provincia con nombre 'Neuquén' debe mostrar el mensaje 'Zona Oeste'.

```
Select P.nombre as NomProv, L.nombre as NomProv, temperatura, AguaCaida,
fecha
from  Localidades L, Precipitaciones T, Provincias P
where (P.idProvincia=l.idProvincia) and (L.idLocalidad = T.idLocalidad)
       and (fecha>='04/01/07') and (fecha<='04/30/07')
order by AguaCaida desc, fecha desc, Temperatura
```

```
Var Loc : string;
    idProv : integer;
Begin
  Query1.close;
  Query1.open;
  Edit1.text := Query1.fieldbyname('NomProv').asString;
  if Query1.fieldbyname('NomProv').asString = 'Neuquén' then
     ShowMessage('Zona Oeste');

  { En lugar agregar el nombre de la provincial al Query y usarlo para
preguntar, podría agregar el idProvincia y hacer un LOCATE de la tabla de
Provincias }
End;
```

2. La tabla LOCALIDADES posee un índice llamado NomLoc que permite ordenarla alfabéticamente por el nombre de la localidad. Utilizando el índice anterior: **(1,50 ptos)**
   a) Visualice en pantalla la cantidad de habitantes de La Plata sin utilizar una consulta SQL. Si no dispone de la información debe indicarlo conbel texto 'No encontrada'.

```
{ TLOCALIDADES es la tabla de localidades}
TLOCALIDADES.itemindex := 'NomLoc';
TLOCALIDADES.setKey;
TLOCALIDADES.fieldbyname('Nombre').asstring := 'La Plata';
If TLOCALIDADES..GotoKey then
             ShowMessage('La Plata tiene'+
           IntToSTr(TLOCALIDADES.fieldbyname('CantHabitantes').asInteger)+
                   'habitantes')
     Else showmessage('no encontrada');
```

   b) Modifique la cantidad de habitantes de la localidad de Junín para que quede registrado en la tabla LOCALIDADES el valor 85320.

```
TLOCALIDADES.Locate('Nombre','Junín',[]);
TLOCALIDADES.edit;
TLOCALIDADES.FieldByname('CantHabitantes').asInteger := 85320;
TLOCALIDADES.post;
```

3. Indique la manera de resolver los siguientes problemas. En cada caso mencione las componentes y propiedades utilizadas e implemente el código Delphi correspondiente: **(2 ptos)**
   a) Seleccionar uno de los días de la semana. El nombre elegido debe aparece en un componente Edit.

Se utilizará un ComboBox con 7 líneas cargadas en su propiedad ítems (cada una con el nombre de un día de la semana). Como respuesta al evento OnClick debe ejecutarse:

```
Edit1.text := ComboBox1.text
```

   b) Implementar un botón que al ser clickeado se corre 40 pixels (puntos) a la izquierda.

En el OnClick del Botón poner:

```
Button1.left := Button1.Left - 40;
```

c) El usuario puede elegir 0,1 o varias opciones de un grupo de 50. Visualizar el texto 'Elección Completa' si todas se encuentran seleccionadas.

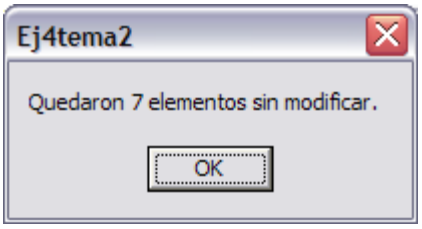
Se utilizará un CheckListBox con 50 líneas cargadas en su propiedad ítems. Para saber si están seleccionadas debe utilizarse la propiedad Checked de cada elemento.

```
Var i : integer;
Begin
   i:= 0;
   while (i<CheckListBox1.count) and
          CheckListBox1.checked[i] do
         i:= i+1;
   if i=CheckListBox1.count then
            ShowMessage('Elección completa');
end;
```

4. Un formulario contiene 4 componentes Edit y un panel. Dentro del panel hay una cierta cantidad de componentes Edit y Buttons.

Los 4 componentes Edit, ubicados en la parte superior del formulario, comparten el mismo comportamiento. Cualquiera de ellos, al ser clickeado, escribe su contenido en cada uno de los componentes del Panel siempre y cuando sea posterior alfabéticamente. Al final informa la cantidad de elementos del Panel que quedaron sin modificar.

Por ejemplo, la imagen de la derecha muestra el resultado de clickear sobre el Edit que contiene la palabra 'despacio'. Dentro del panel quedaron 7 elementos sin modificar porque la palabra 'despacio' no es posterior alfabéticamente a sus respectivos contenidos. Escriba el método compartido por los 4 edit del formulario. **(1,50 ptos)**




```
procedure TForm1.Edit1Click(Sender: TObject);
var i,noPudo : integer;
begin
  noPudo := 0;
  for i:=0 to Panel1.controlCount-1 do
  begin
    if Panel1.controls[i] is TEdit then
         begin
          if TEdit(Panel1.controls[i]).text < Tedit(Sender).Text then
                TEdit(Panel1.controls[i]).text := Tedit(Sender).Text
          else  noPudo := noPudo+1;
         end
    else begin
          if TButton(Panel1.controls[i]).caption < Tedit(Sender).Text then
                TButton(Panel1.controls[i]).caption := Tedit(Sender).Text
          else  noPudo := noPudo + 1;
         end;
  end;
```

```
  ShowMessage('Quedaron '+IntToStr(NoPudo)+' elementos sin modificar.');
end;
```

5. Indicar Verdadero o Falso **(1,50 ptos.)**

(F)	a) En un proceso que define el comportamiento de un evento, el parámetro Source representa a la componente que recibió el evento.
(F)	b) Para establecer una relación Maestro-Detalle entre dos tablas, no se necesitan índices.
(V)	c) En un proceso Drag & Drop, el parámetro del método BeginDrag indica el momento en que comienza el arrastre.
(V)	d) La cantidad máxima de registros que puede devolver una consulta es menor o igual que el producto de la cantidad de registros que contiene cada una de las tablas utilizadas en dicha consulta.
(V)	e) La componente ListBox posee un método que permite copiar el contenido de su lista en un CheckListBox.
(F)	f) Puede utilizarse el método Show para conocer la manera en que se cerró el formulario invocado.
(F)	g) Todos los controles son componentes no visuales.
(F)	h) La invocación de la función StrToFloat con el parámetro '72' produce una excepción.
(F)	i) El método FindKey puede aplicarse sobre una consulta sólo si esta está ordenada.
(F)	j) El ListBox y el RadioButton tienen la propiedad itemindex.