

1	2	3	4	5	6	7	8	TOTAL

## SEMINARIO DE LENGUAJES – OPCION DELHI – 07/07/08

Tema I

Nº Alumno ..... Nombre y Apellido .....

1. ¿Cuáles de las siguientes componentes son contenedoras? (Las respuestas incorrectas restan puntaje) (0.90 pts)

A – Panel      B – ImageList      C – StringGrid      D – Form      E – PageControl      F – MainMenu

Rta : A, D y E

2. Se dispone de una base de datos con información de empleados de una empresa organizada de la siguiente forma:

Categorías : Tabla			
	Nombre del campo	Tipo de datos	Descripción
	idCategoria	Autonumérico	
	Descripción	Texto	Nombre de la Categoría

Empleados : Tabla			
	Nombre del campo	Tipo de datos	Descripción
	DNI	Número	Nº de Documento
	Ap_y_Nom	Texto	Apellido y Nombre
	Categoria	Número	Código de Categoría (e/ 1 y 50)
	Sueldo	Número	Sueldo actual

Escriba el segmento de programa que permita incrementar en un 12% el sueldo de los empleados de las categorías “Técnico Mecánico”, “Ayudante de 1ra” y “Auxiliar de Torno”. (2 pts)

a) Utilizando una relación Maestro-Detalle entre las tablas

```
TEmpleados.IndexFieldName := 'Categoria';
TEmpleados.MasterSource := DSCategorias;
TEmpleados.MasterField := 'idcategoria';
```

```
TCategorias.Filter := '(Descripcion = ''Técnico Mecánico'') or
(Descripcion = ''Ayudante de 1ra'') or (Descripcion = ''Auxiliar de
Torno'')';
TCategorias.Filtered := true;
```

```
TEmpleados.first;
While not TEmpleados.Eof do
Begin
    Sueldo:= TEmpleados.Fieldbyname('Sueldo').asfloat;
    Sueldo := Sueldo + (Sueldo * 12 )/100;
    TEmpleados.Edit;
    TEmpleados.Fieldbyname('Sueldo').asfloat := Sueldo;
    TEmpleados.Post;
    TEmpleados.Next;
End;
```

b) Sin relacionar las tablas y utilizando el método Locate de la tabla CATEGORIAS.

```
Var Descr : String;
    Sueldo : Real;
Begin
    TEmpleados.first;
    While not TEmpleados.Eof do
    Begin
        TCategorias.Locate('idCategoria',
                           TEmpleados.fieldbyname('Categoria').asInteger,[]);
        Descr := TCategorias.fieldByName('Descripcion').asString;
        If (Descr = 'Técnico Mecánico') or
            (Descr = 'Ayudante de 1ra') or
            (Descr = 'Auxiliar de Torno') then
            begin
                Sueldo:= TEmpleados.Fieldbyname('Sueldo').asfloat;
                Sueldo := Sueldo + (Sueldo * 12 )/100;
                TEmpleados.Edit;
                TEmpleados.Fieldbyname('Sueldo').asfloat := Sueldo;
                TEmpleados.Post;
            end;
        TEmpleados.Next;
    End;
```

3. Dada la siguiente consulta que utiliza las tablas del ejercicio 2

(1pto)

**Select \* from Categorías C, Empleados E where C.IdCategoría=E.Categoría**

a) Indique la cantidad de campos (columnas) que poseen sus registros.

6 Campos

b) Indique la cantidad de registros que se obtendrán al realizar la consulta sabiendo que:

1) La tabla CATEGORIAS posee el código y la descripción de las 50 posibles categorías y la tabla EMPLEADOS tiene información de las 150 personas que trabajan en la empresa. Esta segunda tabla tiene la particularidad de poseer 3 empleados de cada categoría.

150 Registros

2) La tabla CATEGORIAS está vacía y la tabla EMPLEADOS es como en b.1)

0 Registros

3) La tabla CATEGORIAS sólo contiene información de las primeras 25 categorías y la tabla EMPLEADOS es como en b.1)

75 Registros

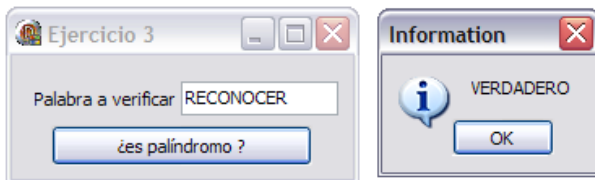
4. Utilizando las tablas del ejercicio 2, visualice en un componente Memo el nombre y apellido y el número de DNI de las personas que pertenecen a una categoría cuyo nombre (descripción) se indica a través de un TEdit. (1 pto)

Se utilizará un ADOQUERY con la siguiente consulta SQL

```
Select Ap_y_Nom,DNI
from Empleados E, Categorías C
Where E.Categoría=C.IdCategoría and C.Descripción =:CAT
```

```
ADOQUERY.Close;
ADOQUERY.Parameters.ParamByName('CAT').AsString := Edit1.text;
ADOQUERY.Open;
Memo1.Lines.Clear;
While not ADOQUERY.Eof do
begin
    memo1.Lines.Add(
        ADOQUERY.fielbyname(Ap_y_Nom).AsString + ' DNI '
        ADOQUERY.fielbyname(DNI).AsString );
    ADOQUERY.Next;
end;
```

5. Dada la siguiente aplicación se desea disponer de una componente que responda al siguiente código del OnClick del botón del formulario principal. (1 pto)



```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Palindromol.palabra := Edit1.text;
    if Palindromol.EsCapicua then
        MessageDlg('VERDADERO',mtInformation,[mbok],0)
    else MessageDlg('FALSO',mtInformation,[mbok],0)
end;
```

```

unit Componente_Tema1;
interface
uses SysUtils, Classes;
type
  TPalindromo = class(TComponent)
  private { Private declarations }
    FPalabra : string;
    FEsCapicua : Boolean;
    procedure Verificar(pal : string);
  public { Public declarations }
    property EsCapicua : Boolean read FEsCapicua;
  published { Published declarations }
    property Palabra:String read FPalabra write Verificar;
  end;
procedure Register;
implementation
procedure Register;
begin
  RegisterComponents('Samples', [TPalindromo]);
end;
procedure TPalindromo.Verificar(pal : string);
var lon, i : integer;
begin
  i:= 1;
  lon := Length(pal);
  while (i<=(lon div 2)) and (pal[i]=pal[lon-i+1]) do
    i := i + 1;
  FPalabra := pal;
  FEsCapicua := (i>(lon div 2));
end;
end.

```

6. Sobre el formulario principal se encuentran pegados un Memo y un ListBox. Se desea habilitar el arraste desde el ListBox al Memo. El proceso consiste en insertar al final de las líneas del Memo todas las líneas del ListBox dejándolo vacío. Indique las propiedades que deben setearse en cada componente y complete los eventos, cuyos encabezados se presentan a continuación, para que la aplicación muestre el comportamiento deseado. **(1 pto)**

```

procedure TForm1.Memo1DragOver(Sender, Source: TObject; X, Y: Integer;
                               State: TDragState; var Accept: Boolean);
begin
  Accept := Source is TListBox;
end;

procedure TForm1.Memo1DragDrop(Sender, Source: TObject; X, Y: Integer);
var I :Integer;
begin
  for I := 0 to ListBox1.Count - 1 do
    Memo1.Lines.Add(ListBox1.Items[I]);
  ListBox1.Clear;
end;

```

7. Implemente la función BuscaMaximo que recibe un arreglo dinámico de reales y retorna el máximo valor. **(1 pto)**

```

function BuscaMaximo( const A : array of Real) : Real;
var i: integer;
    Max :Real;
begin
  Max := 0;
  for I := Low(A) to High(A) do { puede reemplazar Low(A) por el valor 0 }
    if Max < A[i] then Max := A[i];
  BuscarMaximo = Max;
end;

```

8. Indique el valor de verdad de las siguientes afirmaciones **(2.10 pto)**

- |  |          |
|--|----------|
| a) Sólo los strings dinámicos utilizan el proceso de <i>contador de referencias</i> para almacenar la información. | <b>F</b> |
| b) En una aplicación de consola pueden utilizarse los métodos Writeln y Readln de Pascal.                          | <b>V</b> |
| c) Puede utilizarse la función SetLength para definir el tamaño de un TStringGrid.                                 | <b>F</b> |
| d) El índice del primer elemento de un string es 0.  | <b>F</b> |

- e) La función InputBox tiene la posibilidad de devolver un valor por defecto. V
- f) Puede utilizar la propiedad ModalResult para conocer si el usuario ha cerrado un componente OpenFileDialog utilizando el botón Aceptar. F
- g) El método Locate produce una excepción cuando no encuentra el registro buscado. F

1	2	3	4	5	6	7	8	TOTAL

## SEMINARIO DE LENGUAJES – OPCION DELHI – 07/07/08

Tema II

Nº Alumno ..... Nombre y Apellido .....

1. ¿Cuáles de las siguientes componentes son contenedoras? (Las respuestas incorrectas restan puntaje) (0.90 pts)

A - GroupBox    B - ImageList    C - StringGrid    D - Panel    E - MainMenu    F - PageControl

Rta : A, D y F

2. Se dispone de una base de datos con información de autos usados organizada de la siguiente forma:

Marcas : Tabla		
Nombre del campo	Tipo de datos	Descripción
IdMarca	Autonumérico	
Nombre	Texto	Ej: 'Chevrolet', 'Fiat', etc.

Autos : Tabla		
Nombre del campo	Tipo de datos	Descripción
Patente	Texto	Nº de patente o Dominio del auto
Propietario	Texto	Apellido y Nombre del propietario
Marca	Número	Código de marca (valor entre 1 y 100)
Precio	Número	Valor del vehículo

Escriba el segmento de programa que permita incrementar en un 25% el precio de los autos correspondientes a las marcas 'Peugeot', 'Fiat' y 'Volkswagen' (2 pts)

**a) Utilizando una relación Maestro-Detalle entre las tablas**

```
TAutos.IndexFieldName := 'Marca';
TAutos.MasterSource := DSMarcas;
TAutos.MasterField := 'idMarca';
TMarcas.Filter := '(Nombre = 'Peugeot') or (Nombre = 'Fiat') or
                  (Nombre = 'Volkswagen')';
TMarcas.Filtered := true;
TMarcas.First;
While not TAutos.Eof do
Begin
    Precio:= TAutos.Fieldbyname('Precio').asfloat;
    Precio:= Precio + (Precio * 25 )/100;
    TAutos.Edit;
    TAutos.Fieldbyname(Precio).asfloat := Precio;
    TAutos.Post;
    TAutos.Next;
End;
```

**b) Sin relacionar las tablas y utilizando el método Locate de la tabla AUTOS.**

```
Var Nom : String;
    Precio : Real;
Begin
    TAutos.first;
    While not TAutos.Eof do
    Begin
        TMarcas.Locate('idMarca',TAutos.fieldbyname('Marca').asInteger,[]);
        Nom := TMarcas.fieldByName('Nombre').asString;

        if (Nom = 'Peugeot') or (Nom='Fiat') or (Nom='Volkswagen') then
        begin
            Precio:= TAutos.Fieldbyname('Precio').asfloat;
            Precio:= Precio + (Precio * 25 )/100;
            TAutos.Edit;
            TAutos.Fieldbyname(Precio).asfloat := Precio;
            TAutos.Post;
            end;
        TAutos.Next;
    End;
```

3. Dada la siguiente consulta que utiliza las tablas del ejercicio 2 (1pto)

**Select \* from Marcas M, Autos A where M.IdMarca = A.Marca**

- a) Indique la cantidad de campos (columnas) que poseen sus registros.  
6 Campos
- b) Indique la cantidad de registros que se obtendrán al realizar la consulta sabiendo que:
- b.1) La tabla MARCAS posee el código y el nombre de las 100 posibles marcas de automóviles que existen en el mercado mundial y la tabla AUTOS tiene información de 1000 autos usados. Esta segunda tabla tiene la particularidad de poseer 10 autos de cada marca.  
1000 Registros
- b.2) La tabla MARCAS está vacía y la tabla AUTOS es como en b.1)  
0 Registros
- b.3) La tabla MARCAS sólo contiene información de las primeras 50 marcas y la tabla AUTOS es como en b.1)  
500 Registros

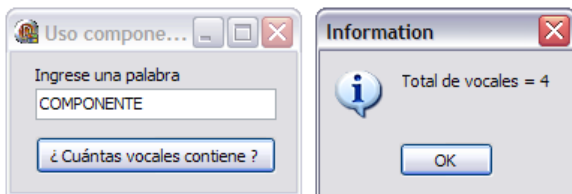
4. Utilizando las tablas del ejercicio 2, visualice en un componente ListBox el nombre del propietario y el n° de patente de los autos que pertenecen a una determinada marca cuyo nombre se indica a través de un TEdit. (1 pto)

Se utilizará un ADOQUERY con la siguiente consulta SQL

```
Select Propietario,Patente
from Autos A, Marcas M
Where (A.Marca=M.IdMarca) and (M.Nombre =:Marca)
```

```
ADOQUERY.Close;
ADOQUERY.Parameters.ParamByName('Marca').AsString := Edit1.text;
ADOQUERY.Open;
ListBox1.Items.Clear;
While not ADOQUERY.Eof do
begin
    ListBox1.Items.Add(
        ADOQUERY.fielbyname(Propietario).AsString + ' Nro Patente '
        ADOQUERY.fielbyname(Patente).AsString );
    ADOQUERY.Next;
end;
```

5. Dada la siguiente aplicación se desea disponer de una componente que responda al siguiente código del OnClick del botón del formulario principal. (1 pto)



```
procedure TForm1.Button1Click(Sender: TObject);
begin
    Vocales1.palabra := Edit1.text;
    MessageDlg('Total de vocales = ' +
        IntToStr(Vocales1.CantVocales),
        mtInformation, [mbok], 0);
end;
```

```

unit Componente_Tema1;
interface
uses SysUtils, Classes;
type
  TVocales = class(TComponent)
  private { Private declarations }
    FPalabra : string;
    FCantVoc : integer;
    procedure Verificar(pal : string);
  public { Public declarations }
    property CantVocales : Boolean read FCantVoc;
  published { Published declarations }
    property Palabra:String read FPalabra write Verificar;
  end;

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('Samples', [TVocales]);
end;

procedure TVocales.Verificar(pal : string);
var cant, i : integer;
begin
  cant := 0;
  for i:= 1 to lenght(pal) do
    if pal[i] in ['a','e','i','o','u'] then
      cant := cant + 1;
  FPalabra := pal;
  FCantVoc := cant;
end;
end.

```

6. Sobre el formulario principal se encuentran pegados un Memo y un ListBox. Se desea habilitar el arraste desde el Memo al ListBox. El proceso consiste en reemplazar las líneas del ListBox por las del Memo dejando a este último vacío. Indique las propiedades que deben setearse en cada componente y complete los eventos, cuyos encabezados se presentan a continuación, para que la aplicación muestre el comportamiento deseado. **(1 pto)**

```

procedure TForm1.ListBox1DragOver(Sender, Source: TObject; X, Y: Integer;
                                State: TDragState; var Accept: Boolean);

begin
  Accept := Source is TMemo;
end;

procedure TForm1. ListBox1DragDrop(Sender, Source: TObject; X, Y: Integer);
var I :Integer;
begin
  for I := 0 to Memo1.Lines.Count - 1 do
    ListBox1.Items.Add(Memo1.Lines[I]);
  Memo1.Lines.Clear;
end;

```

7. Implemente la función BuscaMinimo que recibe un arreglo dinámico de reales y retorna su valor mínimo. **(1 pto)**

```

function BuscaMinimo( const A : array of Real) : Real;
function BuscaMinimo( const A : array of Real) : Real;
var i: integer;
    Min :Real;
begin
  if High(A)>=0 then Min := A[0];
  for I := Low(A) to High(A) do
    if Min > A[i] then Min := A[i];
  BuscarMin = Min;
end;

```

8. Indique el valor de verdad de las siguientes afirmaciones (2.10 pts)
- a) El proceso de *contador de referencias* permite administrar el espacio de memoria ocupado por los arreglos dinámicos. V
  - b) Una aplicación de consola es una aplicación que sólo recibe datos por teclado. F
  - c) El índice del primer elemento de un arreglo dinámico es 1. F
  - d) La función `InputDialog` siempre devuelve un valor de tipo `String`. V
  - e) Puede utilizar la propiedad `ModalResult` para conocer si el usuario ha cerrado un componente `SaveDialog` utilizando el botón Guardar. F
  - f) Si una tabla queda vacía al ser filtrada a través de la propiedad `Filter`, se produce una excepción. F
  - g) Puede utilizarse la función `SetLength` para definir el tamaño de un `TStringGrid`. F