

SEMINARIO DE LENGUAJES - OPCION DELPHI - CURSO 2014

Práctica Nº 1

Fecha de Entrega 29/04 (La Plata) – 28/04 (Tres Arroyos)

Ejercicio 1

Se busca implementar una aplicación con la interfaz indicada en la Figura 1. El botón superior incrementa en uno el valor de temperatura del dispositivo, mientras que el botón inferior lo decrementa en una unidad. La temperatura máxima permitida es 40° mientras que la mínima es 0°. Cada botón se deshabilita cuando se alcanza el límite respectivo. Además, el panel cambia de color según el valor de la temperatura actual. Para más detalles ejecute la aplicación de demostración **Temperatura.exe**.

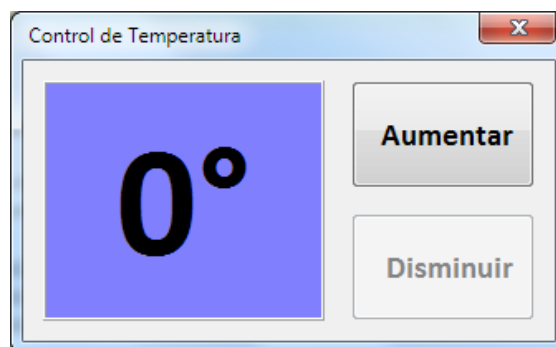


Figura 1

Palabras claves: *TPanel(Caption, Width, Height, Color, Font), TFont(Color, Style,), TButton(Caption, Enabled, Width, Height, OnClick), TForm(Caption, BorderStyle), StrToInt().*

Fecha de resolución sugerida: 18/03 (La Plata) – 20/03 (Tres Arroyos)

Ejercicio 2

Desarrolle una aplicación con la interfaz de la Figura 2, que permita desplazar por el formulario principal un botón "especial". El mismo se encuentra ubicado dentro de un panel y puede ser movido una cierta cantidad de pasos utilizando los botones de la parte inferior del formulario. Cada uno de estos se deshabilita si el botón "especial" toca los bordes del panel y se vuelve habilitar cuando deja de tocar los bordes.

Si ejecuta **BtnEscapa.Exe** verá que el botón central es algo particular ya que no permite que se clickee sobre él a menos que se encuentre fijo. Esto último se indica mediante uno de los dos botones superiores.

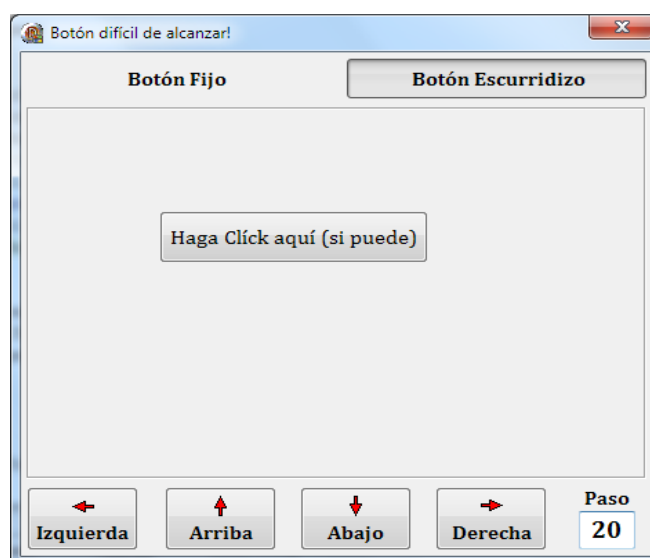


Figura 2

Palabras claves: *TBitBtn(Left, Top, Height, Width, Caption, Glyph, Layout, OnClick), TSpeedButon(Caption, Down, GroupIndex), TPanel (ClientWidth, ClientHeight), TEdit (Text, OnExit), try-except, IntToStr(), StrToInt().*

Fecha de resolución sugerida: 25/03 (La Plata) – 28/03 (Tres Arroyos)

Ejercicio 3

El número de CUIT/CUIT consta de 11 dígitos y dos guiones ubicados de la forma dd-ddddddd-d. La primera parte corresponde a una entidad (persona física o jurídica), la segunda es un número de identificación o documento y la tercera parte es un dígito verificador que permite determinar si hay un error en los dígitos que conforman el CUIL/CUIT.

- a) Escriba una unidad que contenga tres módulos: uno para verificar si la entrada corresponde al formato de un número de CUIT, otro para verificar que la entidad del CUIT es válida y otro para verificar si un número de CUIT dado es válido o no.

```
// Dado un nro.de CUIT representado por un string, retorna TRUE si corresponde
con el formato dd-ddddddd-d y FALSE en caso contrario
Function FormatoCuitValido(numero: String): Boolean;

// Dado un nro. de CUIT con formato valido, retorna TRUE si la entidad es válida
(dos primeros dígitos son 20,23,24,27,30,33 y 34) y FALSE en caso contrario
Function EntidadCuitValido(numero: String): Boolean;

// Dado un nro. de CUIT con formato válido, retorna TRUE si el dígito
verificador es correcto y FALSE si no
Function CuitValido(numero: String): Boolean;
```

Cálculo del dígito verificador

Los 10 primeros dígitos (2 + 8) constituyen el código de identificación y el último el dígito es de verificación. Para obtener esta verificación se procede de la siguiente forma: A cada dígito del código, se lo multiplica por los siguientes números (respectivamente) 5, 4, 3, 2, 7, 6, 5, 4, 3, 2 y cada valor obtenido, se suma para obtener una expresión (que se llamará "valor 1"). Calcular, para este "valor 1", el resto de su división entera por 11. Se obtiene de esta forma un número entre 0 y 10 (que se denominará "valor 2"). Calcular la diferencia entre 11 y el "valor 2"; es decir que se obtendrá un valor comprendido entre 1 y 11 (llamémosle "valor 3"). Si "valor 3" es 11, el código verificador es cero. Si "valor 3" es 10, el código verificador es 9. En cualquier otro caso, el código verificador es "valor 3".

Ejemplo: dado un número de CUIT, que es 20-17254359-7.

2	0	1	7	2	5	4	3	5	9	valor1 = 136	→ 136 módulo 11 = 4	
x	5	4	3	2	7	6	5	4	3	2	valor2 = 4	→ 11 - 4 = 7
-----										valor3 = 7	→ Código de verificación es siete.	
10+00+03+14+14+30+20+12+15+18=136												

- b) Utilice la unidad desarrollada en a) para realizar una aplicación con la interfaz de la Figura 3. Al presionar el botón, este se deshabilita y el número ingresado en el componente Edit es validado utilizando los módulos desarrollados. La aplicación debe reflejar en la etiqueta con texto y con color cuando el CUIT es válido, es inválido, no coincide con el formato o la entidad es inválida. Cuando se modifica alguno de los dígitos del Edit, el botón se vuelve a habilitar y la etiqueta muestra el mensaje "Sin validar". Para más detalle ejecute DemoCUIL.exe.

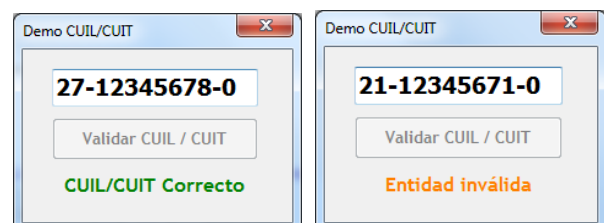


Figura 3

Palabras claves: *unit, interface, implementation, uses, IntToStr(), TMainMenu, TPopupMenu, TMemo(Count, Lines, Lines.LoadFromFile, TListbox(Items, ItemIndex, Clear, Add, SaveToFile), TComboBox(Lines, ItemIndex, Style, OnChange, OnSelected), TOpenDialog(Filename, Execute), TSaveDialog(Filename,Execute), TBitBtn(Caption, Layout, Glyph).*

Fecha de resolución sugerida: 04/04 (La Plata) – 03/04 (Tres Arroyos)

Ejercicio 4

Se desea implementar el clásico juego del AHORCADO de manera que una persona ingrese una palabra secreta y otra(s) trate(n) de adivinarla. Se utilizarán las 27 letras del alfabeto. La Figura 5 muestra la interfaz que debe utilizarse y contiene:

- Menú principal que permite ingresar una nueva palabra o salir del juego.
- Un componente *Image* que muestra el estado del juego.
- Un *Panel* con 27 botones (uno para cada letra) que se deshabilitan al presionarlos.
- Un *Panel* para indicar la cantidad de errores cometidos hasta el momento.
- Otro componente *Panel* donde se va componiendo la palabra a adivinar.

Para comenzar el juego debe seleccionarse Juego/Otra palabra. Esta opción del menú abrirá un nuevo *Form* (Figura 6) donde uno de los participantes ingresará una palabra secreta entre 5 y 16 letras. Es importante "esconder" el texto del componente *Edit* para que los oponentes no lo vean.

El programa deberá colocar de forma automática la letra inicial de la palabra así como todas sus apariciones, deshabilitando el botón correspondiente. Cada letra debe ser deshabilitada luego de ser utilizada. Cada vez que se cometa un error deberá visualizarse la figura correspondiente.

Se dispondrán de siete oportunidades para completar la palabra. Si el oponente no logra adivinar antes de este plazo, debe terminarse el juego visualizando un mensaje que indique cual era la palabra buscada. Por el



Figura 5

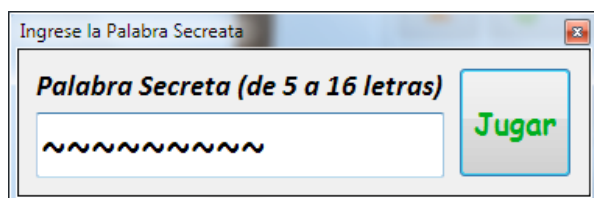


Figura 6

contrario, debe felicitar al participante en caso de que logre su objetivo. Utilice un ícono para la aplicación y el formulario principal. También agregue imágenes en las entradas del menú. Para obtener un mayor detalle del funcionamiento del programa ejecute AHORCADO.EXE

Palabras claves: *Sender, Casting, operador As, TMainMenu, TLabel (Caption), TPanel (Caption, Layout, BevelOuter, Color), TMenuItem (Caption, OnClick), TForm (Width, Height, BorderIcons, ModalResult, ShowModal()), TBitBtn(Caption, Font), TFont(Name, Color, Size, Style), TImage (Name, Transparent, Picture.Bitmap, LoadFromFile()), TEdit(Text, PasswordChar, MaxLength, CharCase, OnChange, OnKeyPress), TImageList(Width,Height), Length(), MessageDlg().*

Fecha de resolución sugerida: 15/04 (La Plata) – 14/04 (Tres Arroyos)

Ejercicio 5

Desarrolle una aplicación que permita realizar el Drag & Drop de figuras sobre un *Panel*. Las mismas deberán arrastrarse desde el formulario hacia el panel y una vez allí podrán moverse en el lugar. Las imágenes que aparecen dentro del panel pueden ser borradas clickeando sobre ellas con el botón derecho del Mouse. El panel permite recibir cuatro figuras como máximo. En la parte inferior de la pantalla debe aparecer la posición actual de la imagen. Note que cada imagen muestra la información de su posición utilizando la propiedad *Hint*. Utilice un ícono especial para el formulario y la aplicación. Para más detalles ejecutar la aplicación **DragDropWalle.exe**.

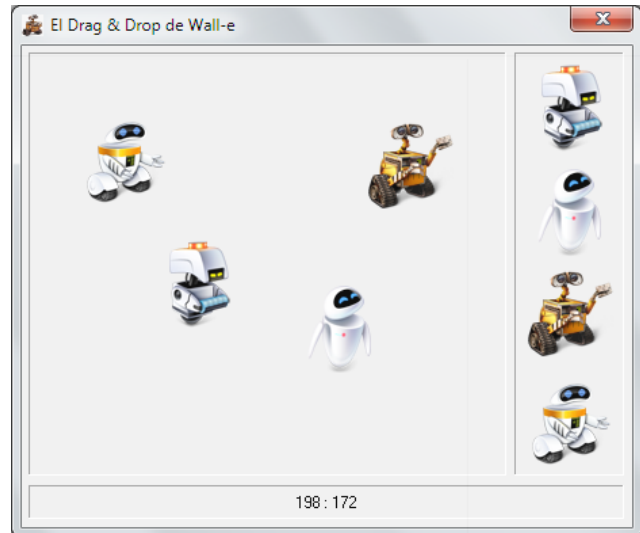


Figura 7

Palabras claves: *Drag & Drop, Sender, Casting, Operador As, TForm (Caption, BorderStyle, Controls, ControlCount), TImage (Name, Hint, ShowHint, Picture.Bitmap, LoadFromFile, OnMouseMove, OnMouseDown, OnMouseUp, OnDragOver, OnDragDrop), TPanel(OnDragOver, OnDragDrop).*

Fecha de resolución sugerida: 22/04 (La Plata) – 21/04 (Tres Arroyos)

Ejercicio 6

Desarrolle una aplicación con la interfaz de la Figura 8. La misma cuenta con un *Memo* (izquierda), dos *Listbox*(derecha), tres botones, un *Mainmenu* y dos *PopupMenu* asociados a las listas. El *Memo* contiene un texto donde cada línea está formada por solo una palabra o un número, y puede ser cargado tanto desde un archivo o como desde el teclado.

Cuando se presiona el botón “Verificar Texto” se debe copiar el contenido del *Memo* en la lista de palabras o en la lista de números, según corresponda. Tenga en cuenta que en todo momento se mantienen actualizadas las cantidades de elementos en el encabezado de cada lista. Los *PopupMenu* de los *Listbox* tienen tres entradas: una para borrar un elemento, otra para borrar todos los elementos y otra para guardar la lista en un archivo. Para más detalle ejecute el programa **Verificador.exe**.

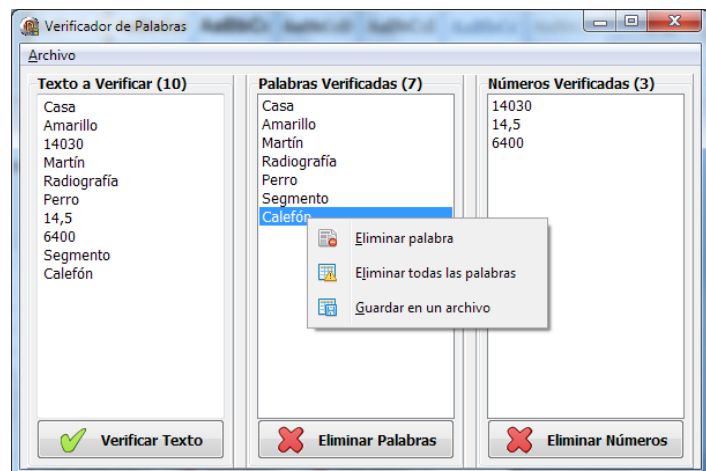


Figura 8

Cuando se presiona el botón “Verificar Texto” se debe copiar el contenido del *Memo* en la lista de palabras o en la lista de números, según corresponda. Tenga en cuenta que en todo momento se mantienen actualizadas las cantidades de elementos en el encabezado de cada lista. Los *PopupMenu* de los *Listbox* tienen tres entradas: una para borrar un elemento, otra para borrar todos los elementos y otra para guardar la lista en un archivo. Para más detalle ejecute el programa **Verificador.exe**.

Palabras claves: *TBitBtn(Align, Caption, Glyph, Layout, OnClick), TPanel (Align, Padding), TGroupBox(Align, Caption), TMainMenu, TPopupMenu, TMemo(Count, Lines, Lines.LoadFromFile, TListBox(Items, ItemIndex, Clear, DeleteSelected, Items.Add, Items.SaveToFile, Items.LoadFromFile, Items.Delete), TOpenDialog(Options, Filter, Filename, Execute), TSaveDialog(Options, Filter, Filename, Execute), DecimalSeparator, StrToFloat(), TImageList(Width, Height).*

Fecha de resolución sugerida: 29/04 (La Plata) – 28/04 (Tres Arroyos)