UNIVERSIDAD NACIONAL DE RÍO CUARTO
FACULTAD DE CS. EXACTAS, FCO-QCAS Y NATURALES - DEPARTAMENTO DE COMPUTACIÓN
ASIGNATURA: **INTRODUCCIÓN A LA ALGORÍTMICA Y PROGRAMACIÓN**
Año: 2016

Proyecto Final

1 - Juego: “Pasapalabra”

El juego “Pasapalabra” consiste de 26 letras ordenadas (de la A a la Z, sin la Ñ), donde cada letra se corresponde a una palabra que contiene dicha letra. Al jugador (usuario) se le muestra cada palabra (una por vez). Hay 2 niveles, el fácil y el díficil. En el nivel fácil se tapan dos letras de cada palabra, y en el difícil 3 o más (si la palabra tiene menos de 4 letras, se tapa solamente una letra). Pero nunca se tapa la letra que corresponde a la palabra, es decir, si estoy en la letra A y la palabra es ***marcos***, se muestra ***ma_c_s***).
El juego comienza por la letra A y se recorre dos veces el abecedario. En el primer recorrido el usuario tiene 2 opciones:
a) intentar acertar la palabra (si lo hace correctamente suma 2 puntos, si falla suma 0 puntos)
b) o bien decir ***pasapalabra***. En esta segunda opción el usuario continúa jugando con la siguiente palabra que le corresponde. La opción de pasapalabra puede ser usada cuantas veces como quiera en el juego, pero sólo 3 seguidas.
En el segundo recorrido del abecedario (en orden de A a Z nuevamente), el usuario solo puede jugar con las palabras que salteó eligiendo ***pasapalabra***, es decir, si intentó adivinar pero falló, no juega de nuevo con esa palabra. Si acierta la palabra en el segundo recorrido suma solo 1 punto.

2 - Implementación

Se desea implementar el juego descripto anteriormente, para esto tiene que tener en cuenta las siguientes observaciones:

- ❑ Las palabras deben ser leídas del archivo de registros “palabras.dat”. Cada registro del archivo tiene la siguiente forma: **TPalabra = <palabra: TLista, letra: Caracter>**
- ❑ Como se ve en el registro anterior, cada palabra es representada por una lista de caracteres, dicha lista, la han implementado con una Unit en el laboratorio de la materia. Como se juega con 26 palabras diferentes, y cada palabra se representa por una lista, debe tener un total de 26 listas.
- ❑ A la Unit debe modificar el campo TInfo para que el registro ahora no solo contenga el carácter sino también si dicho carácter se muestra o se oculta al usuario (ver anexo) y modificar la Unit para soportar dicho cambio.
- ❑ Para representar el abecedario con las palabras puede utilizar la estructura que desee. La estructura que elija es la encargada de encapsular en una secuencia las 26 listas.
- ❑ Luego de cada juego, se debe guardar, en un archivo, el puntaje obtenido y el nombre de usuario del jugador.
- ❑ Para interactuar con el usuario se debe realizar un menú que posibilite al usuario jugar todas las veces que lo desee. También debe posibilitar guardar las partidas con distintos datos de usuarios (**ver:** 3 - Menú). Sobre la representación del usuario: puede usar la estructura que quiera (por ejemplo, crear un registro con varios datos o simplemente guardar su nombre, es a elección). También es de libre decisión el tipo de archivo que va a utilizar para guardar los datos del usaurio.

- ❑ En el menú debe dar la opción de mostrar todos los puntajes con los datos de los usuarios, así también los 10 mejores puntajes ordenados del 1° al 10° (ordenarlos utilizando alguno de los algoritmos de ordenamiento vistos en clase).
- ❑ Además, cada usuario puede preguntar por su puntaje promedio. Dicho promedio debe ser calculado **recursivamente**.

3 - Menú

Al iniciar el juego debe solicitar al jugador (usuario) que cargue su nombre de usuario.
MENÚ PRINCIPAL:
Las opciones mínimas que debe tener el menú principal son las siguientes:

- ➢ Iniciar partida.
- ➢ Ver mi promedio. (muestra el promedio de puntajes del usuario actual).
- ➢ Cambiar de usuario (si no está creado el usuario por el que quiere cambiar debe solicitarle que lo cree).
- ➢ Crear nuevo usuario.
- ➢ Ver los 10 mejores puntajes.
- ➢ Salir del juego

DURANTE EL JUEGO:
Por cada palabra debe mostrarse: la palabra (tapadas las letras correspondientes), la letra a la que se corresponde, el puntaje actual, la cantidad de palabras pasadas (veces que seleccionó ***pasapalabra***) y por último, la cantidad disponible de ***pasapalabra*** que puede elegir (maximo 3 consecutivas como se explicó previamente). Y las siguientes opciones:

- Pasapalabra.
- Adivinar palabra.
- Terminar juego (si elige esta opción no se guarda el puntaje obtenido y vuelve al menú principal).

Al finalizar una partida debe mostrarse el resultado final y volver al menú principal

Entrega:
Debe entregar el código fuente de su implementación, se evaluará todos los conceptos aprendidos en el transcurso de la materia, el código debe ser legible, estar modularizado y bien comentado. Además, debe entregar un informe en el que reporte como fue avanzando el proyecto con las decisiones de diseño del mismo que fue realizando, los problemas que tuvo, los errores que encontró y como fue solucionando dichos problemas y errores. Por último debe entregar un manual de usuario que describa cómo utilizar su juego.

- El proyecto debe realizarse en grupo de dos o tres integrantes cada grupo.

Fechas:
Viernes 4 de Noviembre: explicación del proyecto (sala 101 pab. 2) a las 12 hs. (Muy importante asistir).
Lunes 7 de Noviembre: entregar los integrantes del grupo y análisis del problema (en la clase del práctico).
Jueves 17 de Noviembre: entrega del proyecto completo (informes y código fuente).

NOTA:

- Prestar atención al código (puede verse en el anexo) que se les da para generar el archivo de registro. Tener en cuenta que el módulo que genera dicho archivo de registro, lee las palabras de un archivo .txt, donde cada palabra se encuentra

separada por un salto de línea, por lo que para ir cargando el campo palabra y el campo letra en el archivo de registro se va leyendo caracter por caracter, hasta un fin de línea y así seguir con las siguiente palabra hasta el fin del archivo. Una vez que se leen todos los caracteres que forman una palabra, es decir finaliza el ciclo por fin de línea, se procede a asignar la letra del abecedario que se corresponde con dicha palabra, para esto se utiliza el código ASCII comenzando desde la letra 'a' y así asignarle a las cinco primeras palabras dicha letra, a las siguientes cinco se les asignará la letra 'b', prestar atención a esto ya que si se desea modificar el archivo de palabras para agregar más palabras, se deberá tener en cuenta el índice que lleva la cantidad de palabras que contienen dicha letra. Por ejemplo, si se agrega una palabra más para cada letra del abecedario, el índice que cuenta la cantidad de palabras, en lugar de contar hasta cinco lo tendrá que hacer hasta seis.

ANEXO:

En este anexo se detalla la implementación del archivo de registros del cual deben leer las palabras, observe la acción ***MostrarPalabras*** para guiarse de cómo leer los registros para cargar las palabras en la lista.

```
Program CrearArchReg;
Uses unitListaImplementada;
type
        TPalabra=record
        palabra:TLista;//TLista de caracteres(lse) debe ser
                                letra:char;
                end;
 tarch=file of TPalabra;

var
archPal:Text;
letra:char;
mInfo:TInfo;
miTLista:TLista;
arch:tarch;
regDePal:TPalabra;
i:integer;
caracterInit:Char;

{Inicializa el archivo de registro}
procedure Init(var arch: tarch);
        begin
                assign (archPal,'palabras.txt');
                assign(arch,'palabras.dat');
                Inicializar(miTLista);
                {$I-}
                        reset(arch);
                {$I+}
                if IOResult=0 then
                                reset(arch)
                else
                                rewrite(arch);
        end;

{Carga el archivo de registro a partir de un archivo que contiene palabras}
procedure CargarRegistro(var arch: tarch; var archPal:Text);
begin
        reset (archPal);
        i:=0;
```

```
	caracterInit:='a';
	while(not(EOF(archPal))) do
	begin
		while(not(EOLN(archPal))) do
		begin
			read(archPal,letra);
			mInfo.caracter:=letra;
			mInfo.visible:=true;
			InsertarAlFinal(mInfo,miTLista);
			regDePal.palabra:=miTLista;
		end;
		if(i<5) then //Son cinco palabras entonces a las primeras cinco les asigna la letra 'a'
			begin
				regDePal.letra:=caracterInit;
				write(arch,regDePal);
				i:=i+1;
			end
		else
			begin
				caracterInit:=Succ(caracterInit); //Agrega el siguiente del
abecedario
				regDePal.letra:=caracterInit;
				write(arch,regDePal);
				i:=1;
			end;

		Inicializar(miTLista);
		readln(archPal);
	end;
close(archPal);
close(arch);
end;

{Muestra las palabras que estan cargadas en el archivo de registro}
Procedure MostrarPalabras(var arch: tarch);
begin
	reset(arch);
	while(not(EOF(arch))) do
	begin
		read(arch,regDePal);
		MostrarLista(regDePal.palabra);
		writeln();
	end;
	close(arch);
end;
{Programa Principal}
begin
	Init(arch);
	CargarRegistro(arch,archPal);
	MostrarPalabras(arch);
end.
```