Trabajo Práctico
Cursos K1014 y K1015
1° Cuatrimestre 2011

Proyecto de desarrollo de Software para la simulación de un correo electrónico.

1° Etapa:

Condiciones generales:

- El trabajo práctico será realizado en grupos integrados por 4 o 5 alumnos como máximo. Se recomienda utilizar subprogramas en el desarrollo de los programas, y que estos sean realizados en forma individual por los alumnos del grupo que luego serán unificados en un solo programa.
- Se adjunta el diseño de los registros del archivo, todos los grupos utilizarán los mismos nombres de los campos.
- Presentar una carpeta con carátula que especifique los datos de los integrantes con toda la documentación del programa: enunciado, estrategia, representación gráfica del algoritmo, codificación en Pascal, juegos de datos de entrada y resultados de la ejecución.
- Los temas a aplicar en el trabajo práctico son: tipos de datos simples, registros; leer y grabar registros de un archivo; ingreso de datos por teclado y su posterior consistencia, presentación de menús por pantalla, generación de archivos de texto para su posterior impresión; manejo de archivos; procedimientos y funciones; manejo de arrays.

Desarrollo del trabajo práctico:

Se desarrollará un software para obtener un correo electrónico interno para los alumnos de AyE de Datos.
Se dividirá en dos etapas.

En esta **primera etapa** se desarrollará un programa que presente el siguiente menú:

1. Crear archivo binario de usuarios y mensajes
2. Acceder a AMail 2011
3. Salir de la Aplicación

Opción 1:

Para crear el archivo binario de usuarios se debe:

a) Crear con algún editor texto sin formato un archivo **USUARIO.TXT**, donde se grabará una línea con el nombre del usuario (máximo 16 caracteres, **obligatorio** comenzar con una letra) y otra con el password del usuario (máximo 8 caracteres, letras y/o números). La **Cantidad Máxima de usuarios es 100.**

b) Una vez creado el archivo de texto, grabar por cada usuario que se encuentre en el archivo de texto un registro en un archivo binario **USUARIOS.DAT,** con orden natural y con el siguiente diseño. El campo 3 del registro tendrá un valor inicial igual a -1. También se debe crear el archivo binario de mensajes **MENSAJES.DAT** vacío. Aplique el procedimiento ***AsignarAbrirCrearArchivo,*** en el manejo de los archivos

N° Campo	Descripción del campo	Definición de Tipos

1	Id_Usuario (16 caracteres)	Type
		str16=string[16];
		str8=string[8];
		TregUsuarios = Record
		Id_Usuario: str16;
		Password: str8;
		PosIni: integer;
		End;
		TarchUsuarios = file of TregUsuarios;
		Var
		AUSUARIOS: TarchUsuarios;
		RUSUARIOS: TregUsuarios;

2	Password (8 caracteres)	
3	Posición Inicial Archivo de Mensajes (integer, valor inicial -1)	

Opción 2:

Para acceder a AMAIL, el usuario ingresará por teclado la dirección web www.amail2011.com y se mostrará por pantalla la siguiente información

Bienvenido a Amail2011
La visión de correo electrónico de AyE

Acceder a Amail2011 con su Cuenta AyE

Nombre de Usuario:
Password:

1.Acceder 2.Volver al menú anterior

Se solicitarán los datos de usuario y contraseña. Para el ingreso de la contraseña aplique la función ***ObtenerClave***, y seleccionar una de las dos opciones:

Opción 1.Acceder

- Se deberá buscar en la estructura de datos correspondiente, el usuario y validar que la contraseña sea correcta.
- Si no lo fuera, notificará que hubo un error, permitiendo nuevamente el ingreso de los datos

Una vez que los datos de usuario y contraseña fueron validados se presentará un nuevo menú con las opciones de:

1- Redactar
2- Recibidos
3- Enviados
4- Salir

En esta etapa **solo desarrollaremos la opción 1-Redactar**. Se mostrará la siguiente pantalla:

Bienvenido a Amail2011 La visión de correo electrónico de AyE

De:
Para:
Asunto:
Mensaje:

1- Enviar 2- Cancelar

Por cada mensaje enviado se deberá:

- Generar dos registros de mensaje de acuerdo al diseño del archivo de mensajes **MENSAJES.DAT**, uno para el destinatario (como mensaje recibido) y otro para el que lo envía (como mensaje enviado).
- Los registros de los mensajes se irán grabando al final del archivo.
- Si es el primer mensaje del usuario que envía o recibe se actualizará el campo 3 del registro correspondiente en el archivo de Usuarios, con la posición que corresponda al primer mensaje.

N° Campo	Descripción del campo	Definición de Tipos

1	Id_Usuario (16 caracteres)	Type
		str16=string[16];
		str20=string[20];
		str73=string[73];
		TregMensajes = Record
		Id_Usuario: str16;
		Fecha:longint;
		Destinatario: str16;
		Asunto: str20;
		Mensaje: str73;
		Senial: char;
		End;
		TarchMensajes = file of TregMensajes;
		Var
		AMENSAJES: TarchMensajes;
		RMENSAJES: TregMensajes;

2	Fecha (aaaammdd)	
2	Destinatario (16 caracteres)	
3	Asunto (20 caracteres)	
4	Mensaje (73 caracteres)	
5	Señal de recibido y/o enviado (´R', 'E', char)	

Genere dos archivos de texto con los siguientes listados:

Listado de Usuarios

Id_Usuario	Password	Pos Ini

Listado de mensajes por Usuarios

Id_Usuario xxxxxxxxxxxx

Fecha	Destinatario	Asunto	Senial
dd/mm/aaaa	xxxxxxxxxxxx	xxxxxxxx	x

Restricciones:

- Memoria para Arrays: 5000 bytes
- Accesos al disco: - Un solo recorrido secuencial al archivo de Usuarios.
  - Acceso directo en ambos archivos.
- Incorpore y aplique en el programa los siguientes subprogramas

Procedimiento que verifica si un archivo existe

```
// incluir en el programa principal la siguiente unidad y definir tipo de registro y archivo

uses SysUtils; // Para la función FileExists

type
Tregistro = record
```

```
          // definir registro
          end;
TArchivo = file of Tregistro;

// procedimiento que asigna el archivo si el archivo no existe lo crea en ambos casos lo abre para leer/grabar

Procedure AsignarAbrirCrearArchivo(var Archivo:Tarchivo; NombreFisicoArchivo:string);
 begin
  Assign(Archivo, NombreFisicoArchivo); // Asignamos el archivo
  // función que provee el free pascal para verificar si el archivo existe
  if not FileExists(NombreFisicoArchivo) then      // nombre físico del archivo
    begin
      Rewrite(Archivo); // Si no existe lo creamos
      Close (Archivo);
    end;
   Reset(Archivo);// Lo abrimos para lectura escritura
 end;
```

Función que lee de teclado la clave y muestra ***

```
// incluir la unidad en programa principal y definir tipo de dato

uses crt;

type
   str8= string [8];

// función que obtiene clave del teclado sin hacer eco, usa la cadena como un array

function ObtenerClave(): str8;
   var
    cadena  : str8;
    n       : integer;
   caracter: char;
 begin
     n:=1;
     cadena:= '';      { Genera la Cadena Vacia}
     caracter:=Readkey;  {Lee un caracter del teclado sin mostrarlo por pantalla}
     while (n<=8) and (caracter <> chr(13)) do {Mientras la cantidad de caracteres sea menor o igual a 8
                                                y el caracter leido no sea enter}
       begin
          if (n>1) and (n<=8) and (caracter=chr(8)) { chr(8) es el caracter de retroceso, dieron retroceso??}
            then
              begin
                write(chr(8));    {retrocedo en la pantalla, borro, y decremento el contador de caracteres}
                write(' ');
                write(chr(8));
                Dec(n)
              end
          else
             begin
               cadena[n]:= caracter; {Concatena el caracter a la cadena}
               write('*');                  {Por cada caracter leido muestra un asterisco}
               inc(n);
             end;
        if(n<=8) then
              caracter:=Readkey; {Lee nuevamente un caracter}
    end;
cadena[0]:= chr(n-1);       {asigno longitud a la cadena}
ObtenerClave:= cadena;  { retorna la cadena}
end;
```