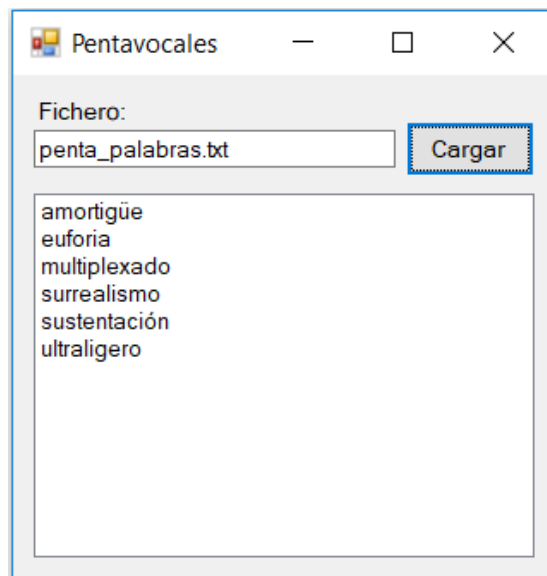


Examen de Programación

Final Mayo – Curso 2016/17

1. Escribe el programa visual “**Pentavocales**”:



El programa tendrá un *TextBox* en el que escribiremos el nombre de un fichero de texto y un botón que será el que inicie el proceso de leer el fichero y escoger las palabras.

Lo que tendremos que hacer es seleccionar del fichero las palabras que contengan las cinco vocales (p.ej.: murciélago) (o sea, al menos una ‘a’, una ‘e’, una ‘i’, una ‘o’ y una ‘u’) y ponerlas en el *ListBox* que hay debajo. Las palabras deberán ponerse en minúsculas, no deberán salir repetidas y deberán estar en el *ListBox* en orden alfabético. Tened en cuenta que las vocales pueden ir acentuadas y llevar diéresis.

Se recomienda hacer una función que te devuelva un *bool* y que te diga si una palabra es pentavocálica o no.

Os paso un par de ficheros con palabras pentavocálicas, uno conteniendo una lista de palabras y otro un texto, para que probéis el funcionamiento de vuestro programa.

2. Escribe las clases **EquipoBaloncesto** y **LigaBaloncesto**, que nos permitirá guardar la información de una liga y la clasificación de sus equipos.

La clase **EquipoBaloncesto** tendrá las siguientes características:

- Atributos: **nombre** que será un *string* y luego 4 enteros: **victorias**, **derrotas**, **puntosAnotados** y **puntosRecibidos**.
- Propiedades: una para cada uno de los atributos, todas de sólo lectura.

- Constructores:
 - Un constructor que recibe un solo parámetro, el **nombre** y pone por defecto el resto de valores a 0.
 - Un constructor que recibe los cinco parámetros: **nombre**, **victorias**, **derrotas**, **puntosAnotados** y **puntosRecibidos** y los guarda en los correspondientes atributos.
- Métodos:
 - Un **ToString()** que nos devolverá el nombre del equipo solamente.

La clase **LigaBaloncesto** tendrá las siguientes características:

- Atributos: **listaEquipos** que será una lista de equipos de baloncesto.
- Constructores: un constructor vacío que inicializará la lista.
- Métodos:
 - Un método **CargarLiga** que recibe un *string* con el nombre de un **fichero** y nos carga los datos del fichero (que será un CSV como el que os paso como ejemplo) y lo guarda en la **listaEquipos**.
 - Un método **GuardarLiga** que recibe un *string* con el nombre de un **fichero** y nos guarda el contenido de la **listaEquipos** en un fichero CSV.
- Propiedad:
 - Una propiedad **Clasificacion** que nos devuelve un *string* con un listado de la clasificación de los equipos (aquí está el tomate). Un ejemplo de lo que nos tiene que devolver esta propiedad podría ser:

Real Madrid	25-7	+303
Baskonia	23-9	+252
Valencia Basket	23-9	+241
FC Barcelona Lassa	22-10	+205
Iberostar Tenerife	22-10	+189
Unicaja	22-10	+166
Herbalife Gran Canaria	21-11	+272
MoraBanc Andorra	16-16	-76

La clasificación de los equipos debe aparecer ordenada por orden de victorias y, en caso de empate a victorias, por orden de diferencia de puntos (puntos anotados menos puntos recibidos).

3. Escribe el programa visual “Validar Registro”:

Usuario	Contraseña	Edad	Correo electrónico
juan	12345678	44	juan@juan.com
pepe	87654321	19	pepe@ibm.com

El programa constará de los siguientes elementos visuales:

- Seis *TextBoxes* para escribir datos correspondientes al **nombre de usuario**, **contraseña** (que hay que escribir dos veces), **edad** y **dirección de correo electrónico** (que también hay que escribir dos veces).
- Cuatro *Buttons* para **Validar** el texto introducido en los *TextBoxes* anteriores. Si la validación es correcta, se deshabilitan el *TextBox* (o *TextBoxes*) correspondiente y se deshabilita también el propio *Button* y ese elemento quedará como validado. Una vez validados los cuatro elementos, se activará el botón que nos permite añadir los datos a la lista.

Si la validación es incorrecta, se deberá dar un mensaje de error (con un *MessageBox*) indicando el problema para que el usuario pueda resolverlo y reintentar la validación.

Las reglas de validación para cada elemento son las siguientes:

- **Nombre de usuario:** deberá tener entre 4 y 20 caracteres y sólo están permitidas letras. Además, el nombre de usuario no se podrá haber usado anteriormente (o sea, que no esté ya en la lista).
- **Contraseña:** deberá tener entre 8 y 16 caracteres. Las dos contraseñas que se escriben deberán coincidir.

- **Edad:** la edad debe de ser un número comprendido entre 18 y 100 años.
- **Correo electrónico:** la dirección de correo electrónico debe contener una arroba '@' (y sólo una). El resto de caracteres podrán ser números, letras o el carácter '.' (punto). Las dos direcciones de correo deben coincidir, evidentemente.
- Un *Button* para **Añadir a la lista** que una vez que se hayan validado todos los datos correctamente, se activará para que pueda pulsarse y añadirá los datos a un *ListView*.
- Otro *Button* para **Cancelar** el proceso de entrada de datos, borrando todos los *TextBoxes* y dejando cada control en su estado inicial.
- El *ListView* estará en modo detalles y solamente servirá para almacenar las líneas introducidas desde el formulario. No habrá que leer ni guardar los datos en ningún archivo.
- Si queréis que queden bonitos los *TextBox* de las contraseñas, la propiedad **PasswordChar** nos permite que salga un carácter en lugar de la contraseña de verdad.
- Al ir pulsando el tabulador nos irá saltando de un componente visual al siguiente. El orden en el que salta es el orden en el que hemos creado los componentes, que normalmente no será el que nos interese. Para que quede bien, hay que tocar la propiedad **TabIndex** de cada elemento y ponerlos en orden.