

Universidad Rafael Landívar

Facultad de Ingeniería

Programación Avanzada, Laboratorio sección 02

Ing. Moisés Alonso



Universidad
Rafael Landívar
Tradición Jesuita en Guatemala

PROYECTO NO.1: Social Network

Eduardo Antonio Peláez Cifuentes.

Carnet: 1096917

Guatemala, 15 de octubre del 2018.

INTRODUCCIÓN

El proyecto fue realizado utilizando Windows Forms en lenguaje de alto nivel C#. Para la sección de “newsfeed” se utilizó una lista enlazada simple y para agregar los elementos se utilizó su método `addLast()`, lo cual la hace tener la disciplina de inserción una cola, como lo requiere el enunciado del proyecto. Para la sección de “messenger” se siguió utilizando una lista enlazada simple, agregando elementos con su método `addFirst()`, lo cual hacía que operara como una pila. Finalmente la lista de amigos es una lista enlazada simple normal.

Análisis

Entradas

1. Archivos de texto correspondientes a cada sección
2. Usuario y contraseña

Salidas

1. Sección de “newsfeed” con todas las noticias y su correspondiente imagen si la tuviera.
2. Sección de “messenger” que muestra los contactos que enviaron mensaje y muestra los correspondientes mensajes.
3. Sección de “amigos” que muestra los amigos que se tienen agregados en la red social.
4. Información de los usuarios amigos.

Procesos

1. Leer cada uno de los archivos.
2. Extraer la información para cada una de las estructuras correspondientes.
3. Establecer usuario y contraseña extraídos del archivo “config” para validar el login
4. Después de haber iniciado sesión, se muestra el formulario de página principal.
5. En la sección de “newsfeed” mostrar las noticias. Haciendo uso de una estructura auxiliar para extraer cada campo correspondiente a una noticia.
6. En la sección “messenger” mostrar una sola vez cada usuario que ha enviado un mensaje y al hacer doble click sobre el usuario mostrar los mensajes enviados.
7. En la sección “amigos” mostrar los usuarios amigos.

Restricciones

1. Uso de imágenes de manera dinámica

Diseño

Algoritmos

Mostrar los contactos que enviaron mensaje una sola vez

Dado que se solicita que los contactos que enviaron una imagen aparezcan una sola vez, se desarrolló el siguiente método:

```
Entero contador1
Entero contador2
Vector auxiliar = pila de mensajes a arreglo
Si el email de vector auxiliar[1] es diferente del email de vector auxiliar[2]
    Agregar a listBox el email del contacto
Mientras contador2 < tamaño de pila de mensajes
    Mientras contador1 < contador2
        Si el email de vector auxiliar[cont1] es diferente del email de vector auxiliar[cont2]
            Contador1 = contador1 + 1
        Sino
            Contador1 = 0
            Contador 2 = contador2 + 1
    Terminar ciclo
Si contador1 = contador2
    Agregar a listBox el email del contacto
    Contador1 = 0
    Contador2 = contador2 + 1
```

Bloquear al contacto

Cadena seleccionado

Cadena nombre = ítem seleccionado de listBox

Para i = 0 hasta tamaño cola de noticias

```
    Si nombre es igual a nombre de arreglo de noticias[i] + apellido de arreglo de noticias[i]
        Seleccionado = email arreglo de noticias[i]
```

Nodo auxiliar = head de cola de noticias

Mientras auxiliar sea diferente a null

```
    Si el elemento de auxiliar = seleccionado
        Eliminar nodo de cola de noticias
    Auxiliar = siguiente de auxiliar
```

Eliminar al contacto

Cadena seleccionado

Cadena nombre = ítem seleccionado de listbox

//Para eliminar noticias del usuario

Para i = 0 hasta tamaño cola de noticias

 Si nombre es igual a nombre de arreglo de noticias[i] + apellido de arreglo de noticias[i]

 Seleccionado = email arreglo de noticias[i]

Nodo auxiliar = head de cola de noticias

Mientras auxiliar sea diferente a null

 Si el elemento de auxiliar = seleccionado

 Eliminar nodo de cola de noticias

 Auxiliar = siguiente de auxiliar

//Para eliminar mensajes del usuario

Nodo auxiliar = head de pila de mensajes

Mientras auxiliar sea diferente a null

 Si el elemento de auxiliar = seleccionado

 Eliminar nodo de pila de mensajes

 Auxiliar = siguiente de auxiliar

//Para eliminar amigo

Nodo auxiliar = head de lista de amigos

Mientras auxiliar sea diferente a null

 Si el elemento de auxiliar = seleccionado

 Eliminar nodo de lista de amigos

 Auxiliar = siguiente de auxiliar

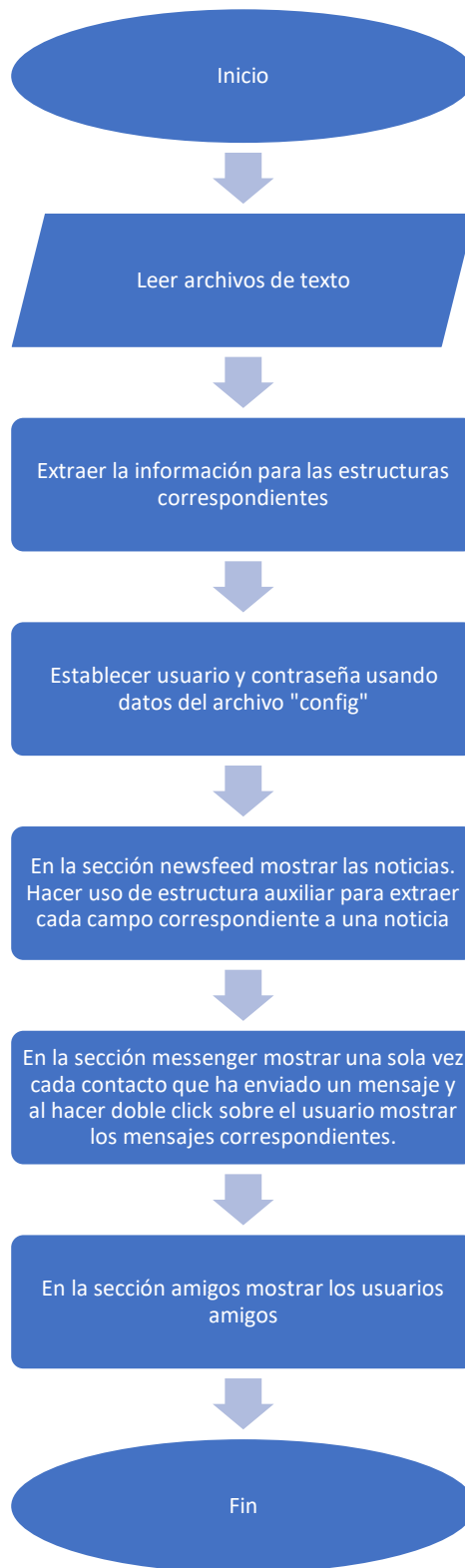
Redimensionar imagen

Función que devuelve la imagen creada en un nuevo tamaño

Redimensionar imagen (imagen imagen Fuente, Tamaño tamaño)

 Devolver (casteo imagen)(nuevo bitmap(imagen Fuente, tamaño))

Diagrama de flujo



Identificación de métodos

Clase “News”

TIPO	MÉTODO
string	getEmail()
void	setEmail(string email)
string	getType()
string	setType(string type)
string	getPath()
string	setPath(string path)
string	getText()
string	setText(string text)
string	toString()
Constructor	<u>News()</u>

Clase “Contact”

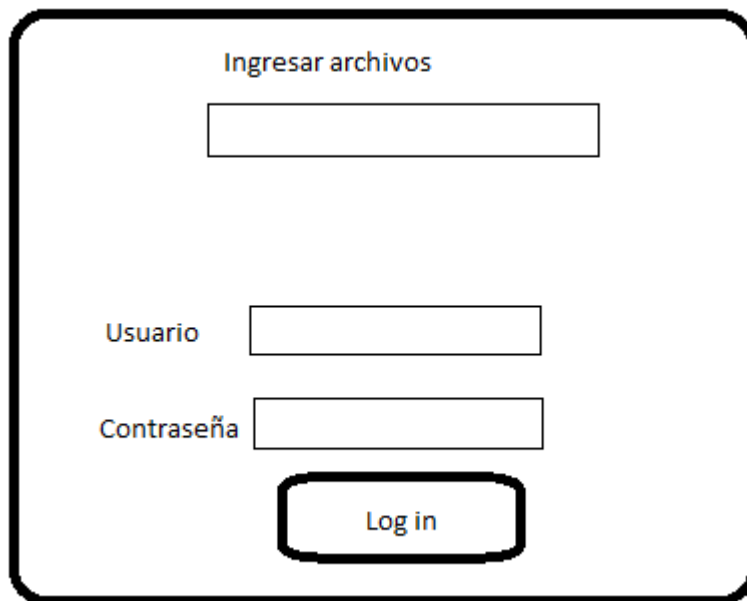
TIPO	MÉTODO
string	getEmail()
void	setEmail(string email)
string	getName()
string	setName(string name)
string	getLastName()
string	setLastName(string lastName)
string	getAge()
string	setAge(string age)
string	getBirthDate()
string	setBirthDate()
string	getStatus()
string	setStatus(string status)
string	toString()
Constructor	Contact()

Clase “Message”

TIPO	MÉTODO
string	getEmail()
void	setEmail(string email)
string	getText()
string	setText(string type)
string	getHour()
string	setHour(string path)
string	getStatus()
string	setStatus(string text)
string	getSend()
string	<u>setSend()</u>
string	<u>toString()</u>
Constructor	<u>Message()</u>

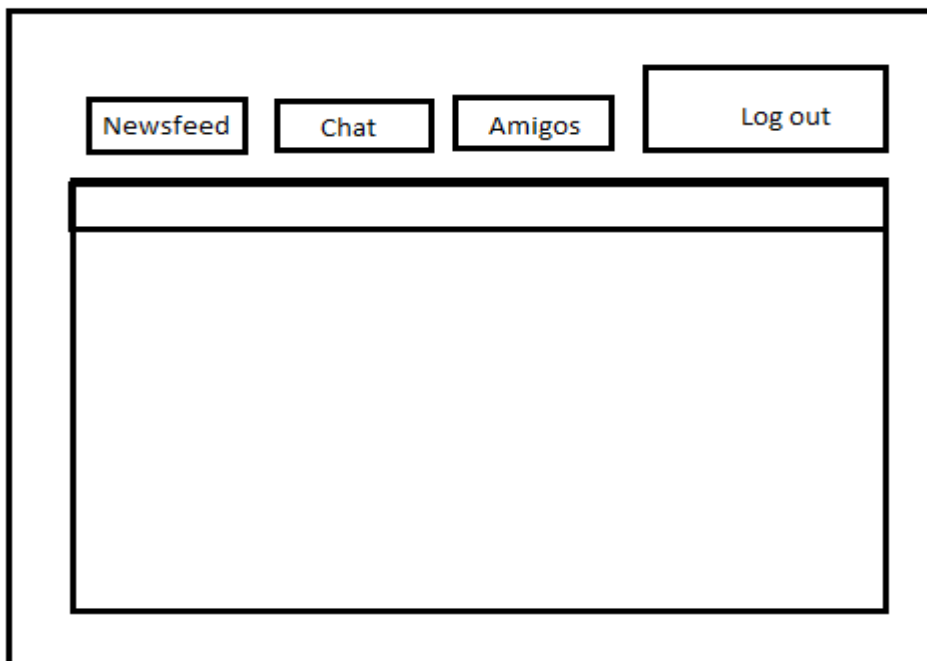
Diseño de pantalla

Diseño de log in.



A login form design enclosed in a rounded rectangle. At the top, the text "Ingresar archivos" is centered above a rectangular input field. Below this, the label "Usuario" is positioned to the left of another rectangular input field. Further down, the label "Contraseña" is to the left of a third rectangular input field. At the bottom center, there is a rounded rectangular button labeled "Log in".

Diseño de pantalla principal



A main screen design enclosed in a rectangle. At the top, there is a horizontal row of four buttons: "Newsfeed", "Chat", "Amigos", and "Log out". Below this row is a large rectangular area, likely a placeholder for content, which is divided into a header section and a main body section by a horizontal line.

Conclusiones

1. La lectura de archivos se realiza con el objeto StreamReader
2. Las estructuras de datos dinámicas no tienen un tamaño fijo, crecen según las capacidades del equipo y su uso.
3. Los nodos son la estructura de datos más básica.

Bibliografía

1. Cómo leer archivos de texto mediante System.IO y Visual C# (s.f). [En red] Recuperado el 13/09/2018. Disponible en: <https://support.microsoft.com/es-es/help/306777/how-to-read-a-text-file-by-using-system-io-and-visual-c>
2. Cómo: Abrir archivos mediante el componente OpenFileDialog (s.f). [En red] Recuperado el 13/09/2018. Disponible en: <https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/how-to-open-files-using-the-openfiledialog-component>
3. How to resize an Image C# (s.f). [En red] Recuperado el 13/09/2018. Disponible en: <https://stackoverflow.com/questions/1922040/how-to-resize-an-image-c-sharp>