



**UNIVERSIDAD DE SANTIAGO DE CHILE FACULTAD DE  
INGENIERÍA DEPARTAMENTO DE INGENIERÍA  
INFORMÁTICA**

**Paradigmas de Programación**

**Laboratorio N°4**

Alumno: Jose Toro  
Sección: B-2.  
Profesor: Víctor Flores Sánchez.

Santiago – Chile  
1-2021

## TABLA DE CONTENIDOS

|                        |  |    |
|------------------------|--|----|
| 1.1.1                  | Descripción del problema.....  | 4  |
| 1.1.2                  | Descripción del paradigma utilizado.....   | 4  |
| 2.1.1                  | Análisis del problema respecto a sus requisitos específicos. ....                                      | 5  |
|                        | Para el desarrollo de la solución se solicitan los siguientes requisitos funcionales específicos:..... | 5  |
| 2.1.2                  | Diseño de la solución. ....  | 6  |
| 2.1.3                  | Aspectos de implementación. ....   | 7  |
| 2.1.4                  | Instrucciones de uso. ....   | 8  |
| 2.2                    | Resultados y autoevaluación.....   | 9  |
| 2.3                    | Conclusión.....  | 11 |
| CAPÍTULO 3. ANEXO..... |  | 12 |

## **CAPÍTULO 1. INTRODUCCIÓN**

En este informe se describe el desarrollo de la solución al cuarto laboratorio de esta asignatura, se revisara el problema, se describirá el paradigma utilizado esta entrega, se hará un contraste de resultados conseguidos con este paradigma en comparación a los tres anteriores, además se presentaran instrucciones de uso para la solución.

### **1.1.1 Descripción del problema**

En esta entrega, se continua el mismo problema que en los anteriores, crear una red social que sea funcional y cumpla con determinados requisitos, ya sean, registrarse en ella, iniciar sesión, cerrar sesión, publicar, comentar , seguir a otros usuarios, etc, sin embargo, el problema esta vez, también solicita una implementación de una interfaz grafica, todo esto utilizando el paradigma orientado a eventos, utilizando Java o C# para , para la realización de la solución se utilizo C# con WPF (Windows Presentation Foundation) como biblioteca de entorno grafico.

### **1.1.2 Descripción del paradigma utilizado**

El paradigma utilizado en esta ocasión fue el orientado a eventos, sin embargo, también se utilizó POO en la lógica del programa, ya sea para la creación de publicaciones, usuarios, comentarios, se utilizo un diseño prácticamente idéntico que en el anterior laboratorio, ya que la lógica del código seguía siendo la misma, en donde se utiliza el nuevo paradigma es en cuanto a lo visual, el paradigma orientado a eventos tiene como principal característica que los eventos “dirigen” el flujo y estructura del programa, lo controrario a como seria programas en imperativamente (visto en ramos anteriores como puede ser C en métodos, o Python en fundamentos de programación), en este paradigma el flujo es controlado por el usuario a través de los eventos que este mismo provoca, un evento es una acción que el programa puede detectar, por ejemplo, cuando uno aprieta el botón de cerrar un programa, esta acción provoca que se ejecute determinada parte del código, la cual se encarga de cerrar el programa o una ventana de este, en esta solución se observan distintos eventos para simular el funcionamiento de una red social.

## **CAPÍTULO 2. SOLUCIÓN DEL PROBLEMA**

### **2.1.1 Análisis del problema respecto a sus requisitos específicos.**

Para el desarrollo de la solución se solicitan los siguientes requisitos funcionales específicos:

- Implementación de distintas clases y estructuras para cumplir con todos los siguientes requisitos funcionales que se listaran a continuación.
- Función authentication: Función que permite registrar usuarios, iniciar sesión y cerrar sesión.
- Función post: Permite al usuario actualmente conectado a la red social realizar una publicación.
- Función follow: Permite al usuario actualmente conectado a la red social poder seguir a otro usuario.
- Función share permite al usuario actualmente conectado a la red social compartir el contenido de una publicación.
- Interacciones través de interfaz gráfica: La posibilidad de que el usuario interactúe de manera mas “directa” e intuitiva con el programa
- 

Además de las funciones anteriormente mencionadas, se da a elegir la posibilidad de implementar mas funciones de forma complementaria, esas funciones complementarias son:

- Función comentar: Permite al usuario actualmente conectado a la red social realizar comentarios.
- Función like : Permite al usuario actualmente conectado a la red social “reaccionar” a publicaciones o comentarios mediante likes.

### 2.1.2 Diseño de la solución.

El diseño de la solución se basó en la anterior entrega, ya que, el laboratorio correspondiente a POO, contenía toda la lógica básica que esta solución requería, por esto, corresponde al diagrama de análisis para el diseño de esta solución ( ver Anexo - Imagen 1)

Para poder implementar la interfaz gráfica, al igual que la anterior entrega se utilizó el patrón de arquitectura MVC, lo que permite tener el programa dividido en partes.

- Modelo: Contiene las definiciones de las estructuras y clases del proyecto.

- Control: Encargado de ocupar elementos del modelo para realizar las acciones solicitadas (por ejemplo utiliza al modelo usuario, redsocial, para registrar un nuevo usuario), para luego “entregarle” estos cambios a la vista.

- Vista: contiene la representación gráfica que ve el usuario, comunicada con el controlador. Estando definida la arquitectura que se utilizara, la solución se implementó utilizando división en subproblemas (conocimientos previos), en primer lugar, se crean las clases del modelo, luego se crea la lógica contenida en el controlador, con todo esto listo, se procedió a realizar distintos bosquejos de cómo sería la vista, optando al final por el más minimalista y simple de entender, con todo esto terminado, se crea cada ventana con la que el usuario tendrá contacto, ya sea la ventana para crear una publicación, la ventana para seguir a un usuario, o también, la ventana que permite visualizar todas las publicaciones de la red social de una manera más “ordenada”. Con todas las ventanas terminadas se comenzó a unir finalmente, el control con la vista, mediante los eventos que genere el usuario. Por ejemplo, si el usuario presiona el botón “registrarse” este botón desencadena en el código que se abra una nueva ventana, la cual está unida al control y realizando un llamado a la función Register, y luego se conectaron todas las ventanas al control para tener una red social funcional.

La solución se implementó en el IDE Visual Studio, utilizando el lenguaje de programación C# utilizando la biblioteca gráfica WPF, esta biblioteca separa la parte gráfica de una ventana con su parte lógica, se parados en lenguaje declarativo XAML y C# respectivamente. Aun teniendo que trabajar con el lenguaje XAML, el uso de Visual Studio lo hizo prácticamente sin que yo notase que no estaba utilizando C# debido a la facilidad y ayuda que entrega el mismo IDE al momento de realizar interfaces gráficas.

Finalmente la el diagrama de la solución final se puede ver en el anexo imagen 2.

### **2.1.3 Aspectos de implementación.**

Como ya fue dicho anteriormente este programa fue realizado en el lenguaje C#, usando el IDE Visual Studio y la biblioteca grafica WPF, utilizar este conjunto de opciones y no utilizar la otra opción de Java fue porque quería intentar un lenguaje nuevo, sin embargo, al momento de escribir en C# en comparación a Java la diferencia es minúscula, por no decir nula, se debe mencionar que debido al conjunto de opciones elegidas, la ejecución de este programa solo puede ser utilizando el sistema operativo Windows.

#### **2.1.4 Instrucciones de uso.**

Este laboratorio a diferencia de los anteriores es mucho mas intuitivo y simple de usar, debido a su interfaz grafica, la única instrucción que debe ser aclarada si o si, es que este programa solo puede ser ejecutado en el sistema operativo Windows.

Para poder usar el programa de manera adecuada:

- abrir el ejecutable ( termina en .exe)

- una vez cargado el programa es posible comenzar a utilizarlo, esta red social esta “poblada” con mas usuarios, unas cuantas publicaciones y comentarios, para poder probar sus funcionalidades sin tener que realizar todo desde cero.

Además se anexan unas imágenes con el programa ya ejecutándose.

## 2.2 Resultados y autoevaluación.

Requerimientos no funcionales:

|  |   |
|--|---|
| Lenguaje y herramientas de trabajo:                        | Logrado: el lenguaje utilizado es C# versión 8.0, con el IDE Visual Studio 2019, WPF como biblioteca de entorno grafico   |
| Interacciones con el programa:                             | Logrado: todas las interacciones se deben realizar mediante la interfaz grafica.  |
| Uso del paradigma  | Logrado: la implementación y diseño sigue el POO y a su vez el lineamento del paradigma dirigido por eventos, ya que el programa reacciona a las acciones del usuario, y es este quien decide el flujo del programa |
| Separación modelo – vista:                                 | Logrado: al igual que en la anterior entrega se utiliza la arquitectura MVC la cual permite tener un acoplamiento muy bajo por no decir nulo y una alta cohesión.   |
| Diagrama de análisis y diseño                              | Logrado: ambos diagramas se encuentran en el anexo  |
| Historial: al menos 10 commits distribuidos en una semana. | No logrado: si bien se logra el mínimo de commits, estos fueron subidos en un plazo muy corto, no cumpliendo la distribución en una semana.   |
| Prerrequisitos de funciones:                               | Logrado: cada función tiene su prerrequisito creado e implementado.   |



#### Requerimientos Funcionales:

Los requerimientos funcionales, no presentan problemas, cumpliendo con las restricciones solicitadas, fueron probados por mi y un usuario externo el cual posee conocimientos prácticamente nulos en el ámbito de la programación, para garantizar una comprensión de la mayoría de los posibles usuarios.

## **2.3 Conclusión.**

Considerando los resultados mencionados, es posible notar que se cumple con la mayoría de los objetivos ( excluyendo el uso de github) para esta entrega.

En un comienzo aun con mis conocimientos previos por ya haber cursado este ramo con anterioridad y ramos anteriores, el funcionamiento de la biblioteca grafica WPF y como conectar la parte visual con el resto del programa fue un gran desafio ya que en un comienzo no lograba comprender el paradigma en su totalidad, una gran dificultad que se me presento fue crear una interfaz “bonita” ya que mis manejos de espacios, colores son mas bien nulos en cuanto a embellecer un código se trata, ya que suelo centrarme en que sea funcional , mas que en otros aspectos, sin embargo , creo que logre una interfaz grafica no “hermosa” pero tampoco horrible. Para resumir, la principal dificultad es crear una interfaz grafica, bonita y funcional, si no se tiene el tiempo ni el conocimiento del paradigma, ni de la biblioteca a utilizar.

### CAPÍTULO 3. ANEXO

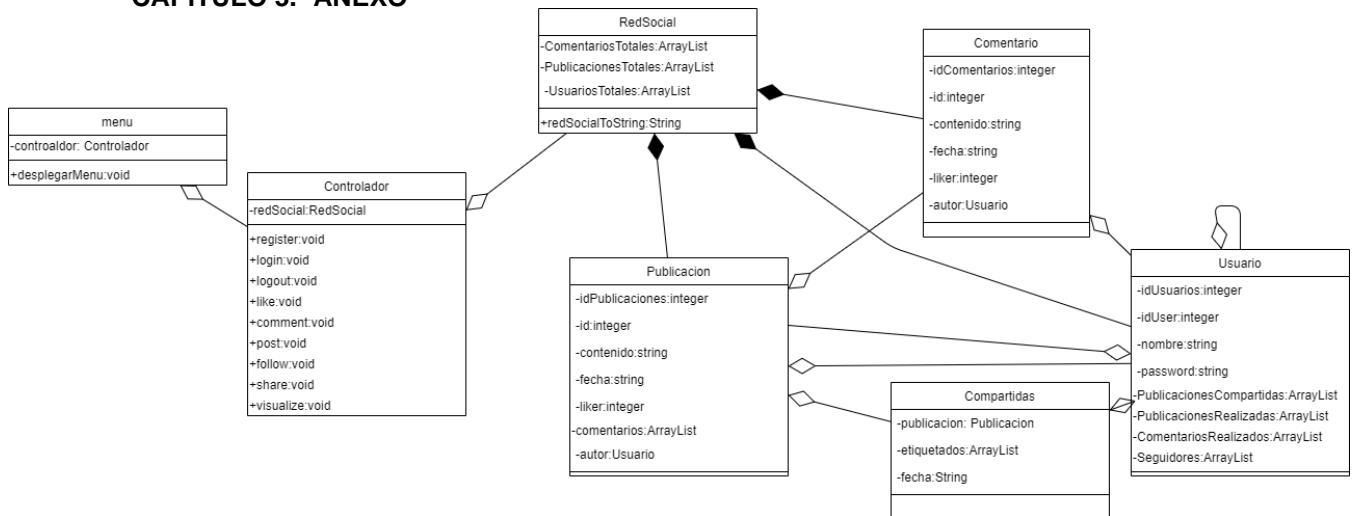


Imagen 1 : diagrama de análisis

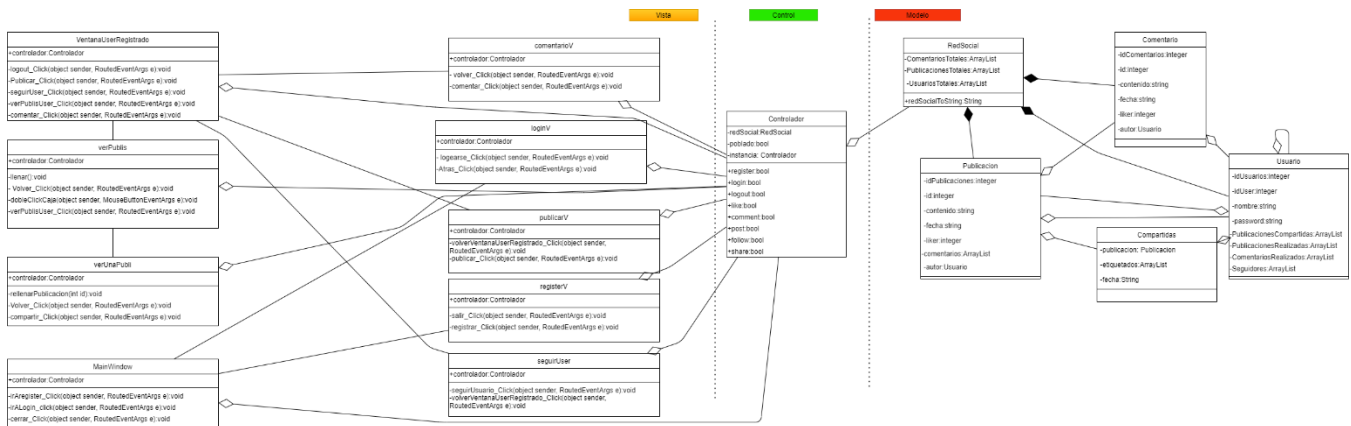


Imagen 2: diagrama de solución.

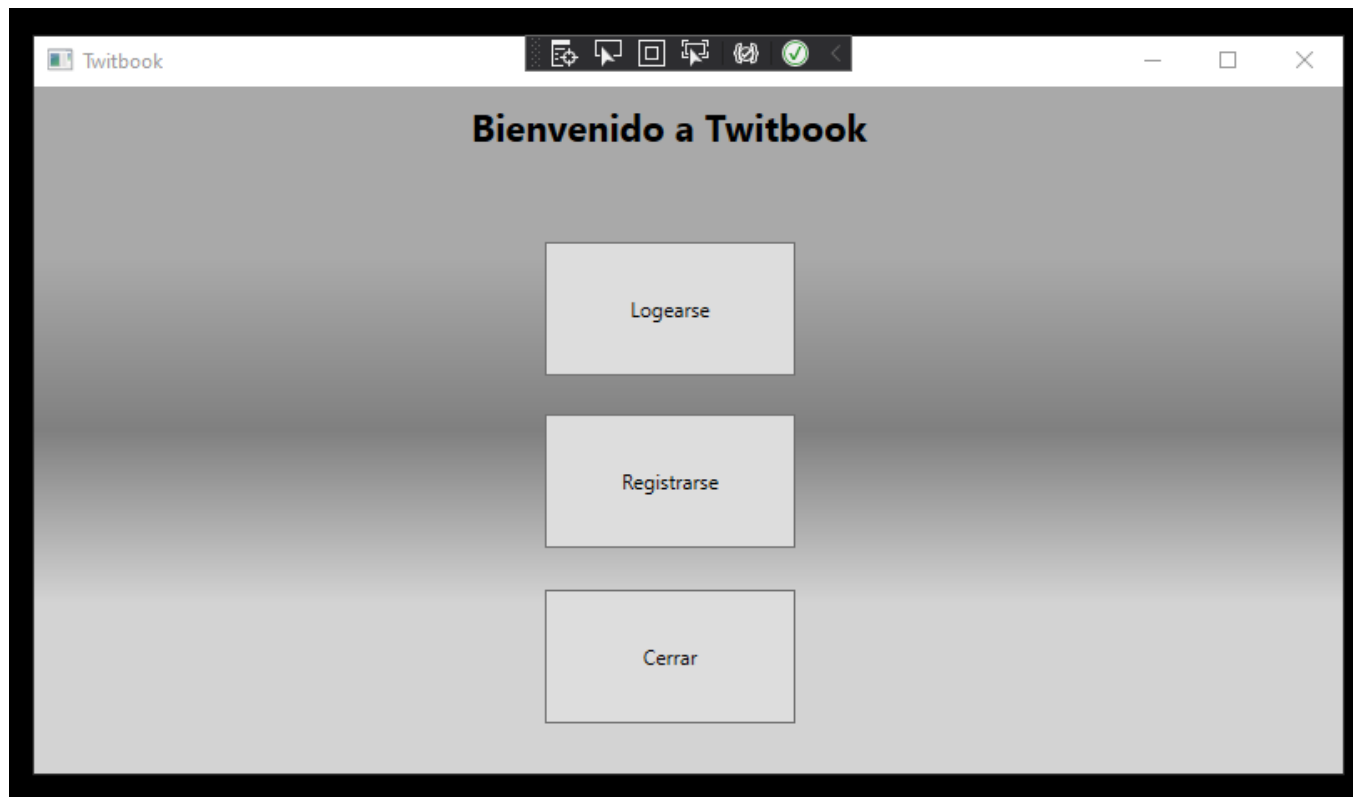


Imagen 3: muestras de ejecución

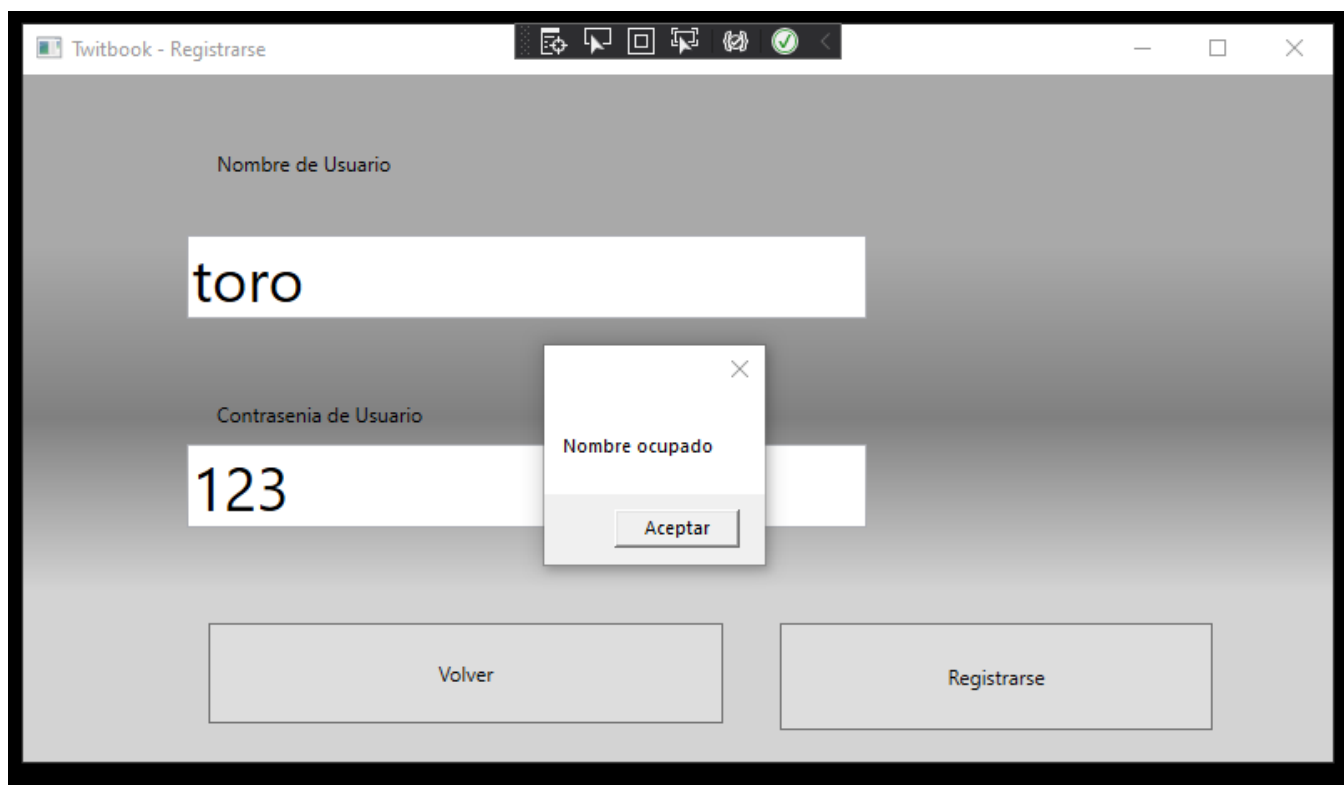


Imagen 4: registrarse con username utilizado

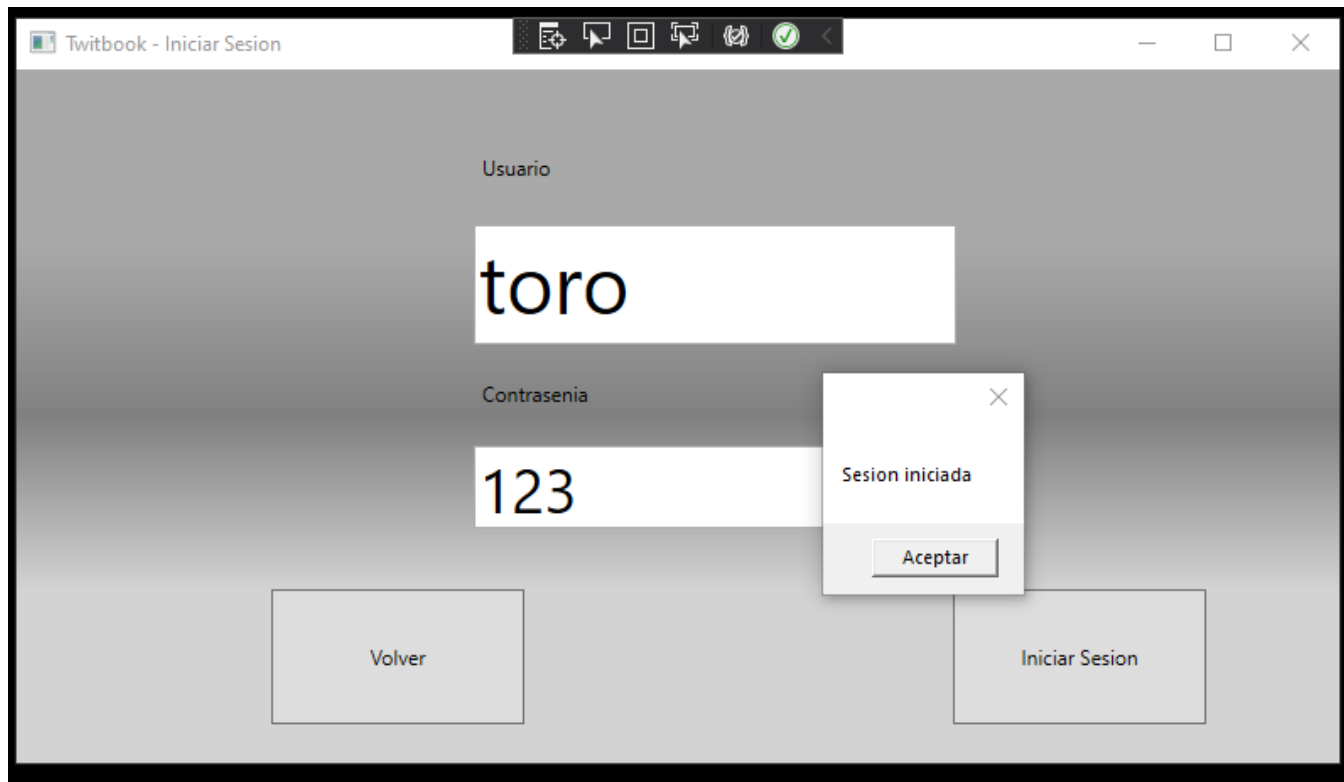
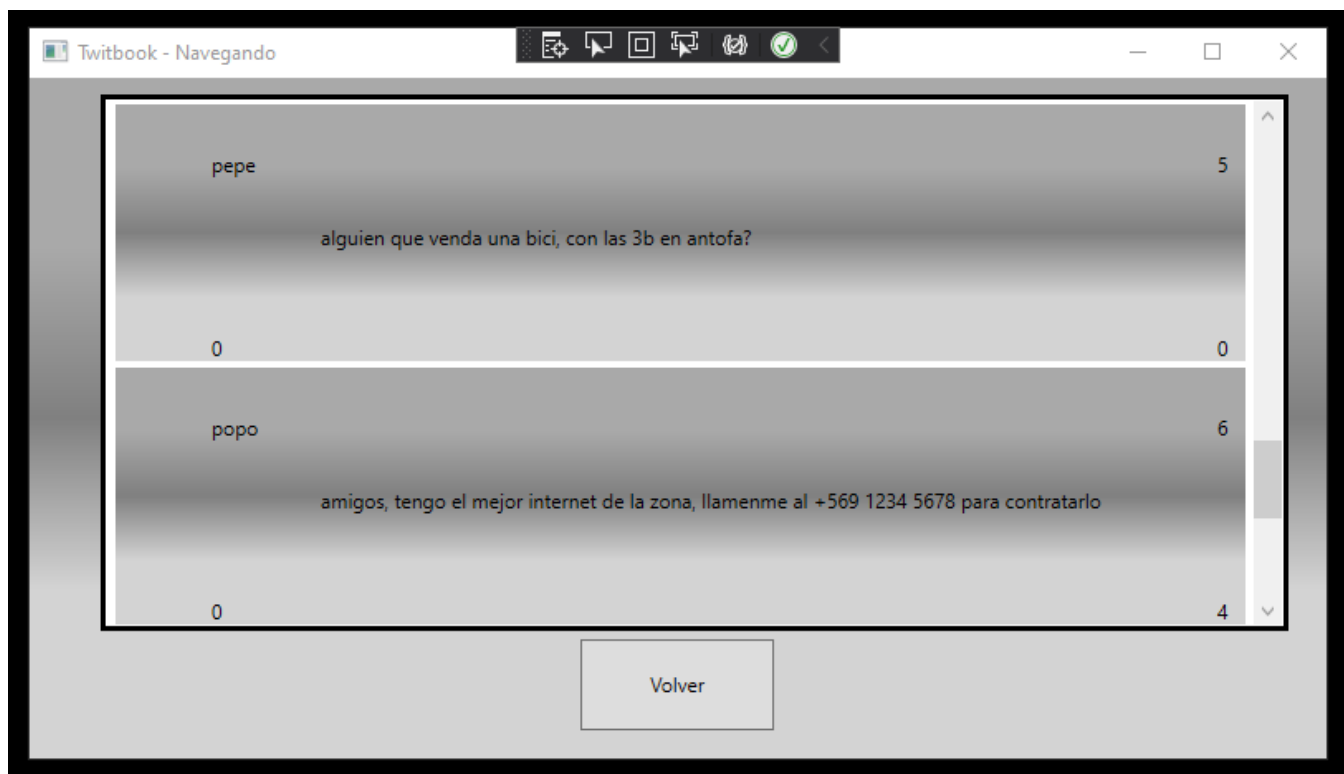
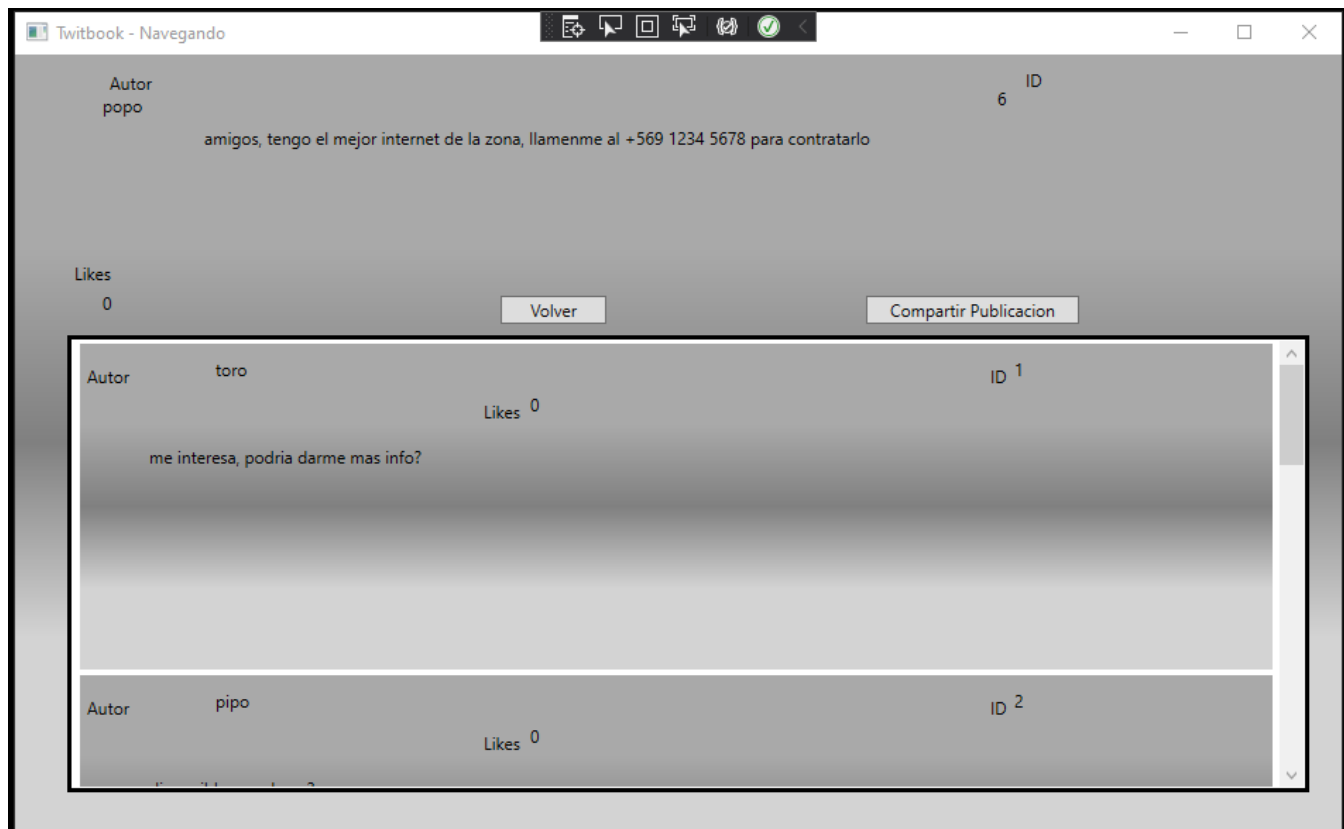


Imagen 5: logearse con usuario(de los que hice para poblar la base).



**Imagen 6: visualizando publicaciones de base.**



**Imagen 7: visualizando una sola publicación y sus comentarios.**