

Windows Forms

1 - Cambiar nombre del formulario: Frm_Login

(Name): Login

Text: Login

StartPosition: CenterScreen

MaximizeBox: false

MinimizeBox: false

TextBox de usuario:

- (Name): txt_Usuario
- borderstyle: fixed single
- PlaceholderText: Ingresa usuario

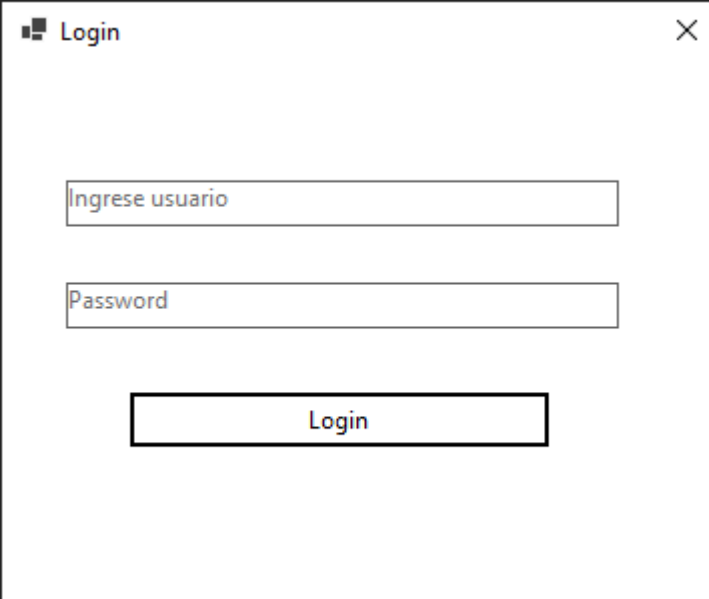
TextBox de password:

- (Name): txt_Pass
- borderstyle: fixed single
- PasswordChar: *
- PlaceholderText: Password

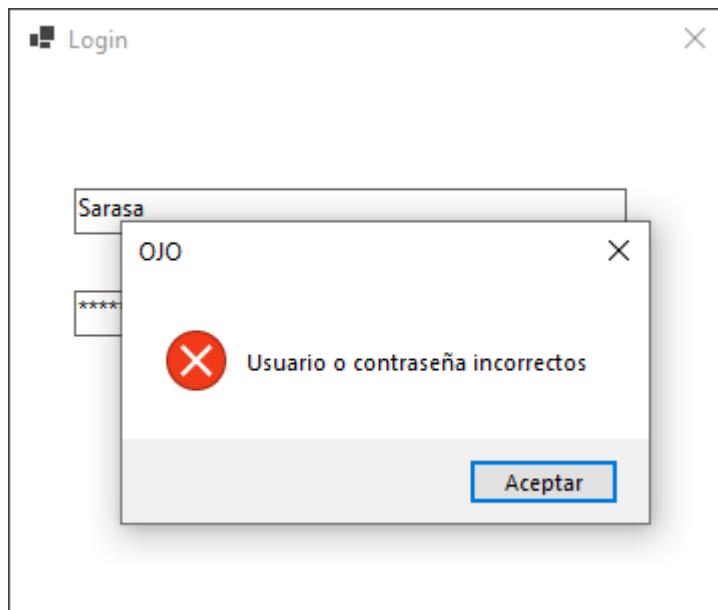
Button:

- (Name): btn_Login
- text: Login
- flatAppearance
 - bordercolor: black
- FlatStyle: flat

Botón login: Que busque en el diccionario si el usuario (que es la key) tiene la misma contraseña (que es el value) y si la contraseña es correcta cierra el form, si es incorrecta tira mensaje de error



The image shows a screenshot of a Windows Forms application window titled "Login". The window has a standard Windows title bar with a minimize button, a maximize button, and a close button (X). Inside the window, there are two text input fields. The first field has a placeholder text "Ingresa usuario" and the second field has a placeholder text "Password". Below these fields is a button labeled "Login". The button has a black border and a flat appearance.



2 - Un formulario que dé de alta a una persona y que lo agregue a una lista

3 - Formulario modal , MDI, y Normal. Tenemos una aplicación que corre en 2 monitores, en uno hay un formulario donde escriben el mensaje y tiene un formulario modal que setea el formato (tamaño y color), mientras que en la otra, un donde se muestra

COLECCIONES

List<T>

4 - Se crea una lista de enteros y se definen los dos primeros enteros con el valor 1. Son los dos primeros valores de una sucesión de Fibonacci, una secuencia de números. Cada número sucesivo de Fibonacci se obtiene con la suma de los dos números anteriores.

5 - Crear una lista de mascotas. Pepe, Juana, Romeo, Carmela, Dumbo, Bizcocho, y Tutuca. Utilizar el método IndexOf para buscar si Chester existe en la lista. (Debería devolver -1 si no está). Y encontrar el método que los ordena de manera alfabética.

Queue

6- Ezequiel mandó a Lucas a formar una fila, para comprar tres tickets para ver el show de Iron Maiden. PERO cada persona puede comprar solo un ticket por vez y después deben volver a empezar la fila. Cada ticket tarda un minuto en salir, así que la fila se mueve de manera fluida.

Se te dará una matriz / lista / vector con todas las personas en cola y la posición inicial de Lucas, así por ejemplo, sabiendo que Lucas está en la tercera posición (que

consideraremos igual al índice, 2: él es ¡el tipo que quiere 3 boletos!) y la cola inicial es [2, 5, 3, 4, 6].

El primer tipo recibe su boleto y la cola va ahora así [5, 3, 4, 6, 1], luego [3, 4, 6, 1, 4] y así sucesivamente. Al final, Lucas estará haciendo cola durante 12 minutos, ¡historia real!

Pilas

7- Dada una pila de N ints de manera que el primer elemento representa la parte superior de la pila y el último elemento representa la parte inferior de la pila. Necesitas sacar al menos un elemento de la pila. En cualquier momento, puedes convertir la pila en una cola. La parte inferior de la pila representa el frente de la cola. No puedes volver a convertir la cola en una pila. Tu tarea es eliminar exactamente K elementos de modo que la suma de los K elementos eliminados se maximice.

Formato de salida :

Imprime la máxima suma posible de los K elementos eliminados

8 - A Romeo le gusta saltar de un edificio a otro. Pero solo salta al siguiente edificio más alto y se detiene cuando no hay un edificio más alto disponible. La resistencia necesaria para un viaje es la suma de todas las alturas en las que Romeo salta hasta detenerse.

Si las alturas son [1 2 4], y comienza desde 1, pasa a 2 la resistencia requerida es $1+2=3$, luego de 2 a 4. La resistencia para todo el viaje es $1+2+4=7$. Encuentre la resistencia máxima necesaria si puede comenzar su viaje desde cualquier edificio.