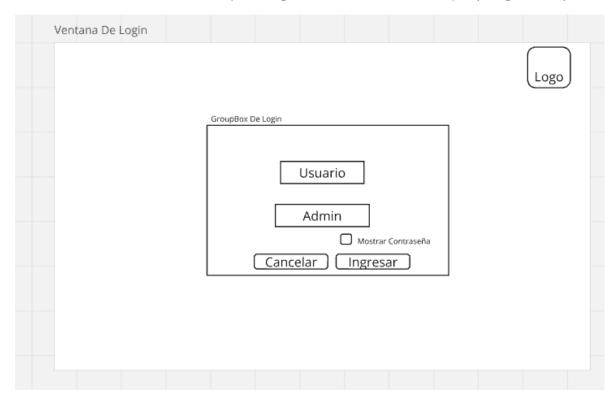
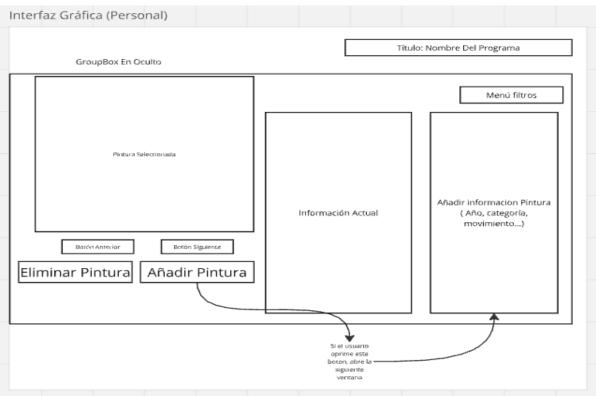
## **Documentación Adicional**

A continuación, una descripción gráfica de los mock-ups (en general):







Para implementar el HashMap que hasta ahora tenemos en el código, le preguntamos a ChatGPT como podríamos incluirlo de forma efectiva.

## Prompt utilizado:

Por ejemplo podría implementar HashMap en el siguiente código?: FrmPersonal form2 = new FrmPersonal(); FrmUsuarioFinal form3 = new FrmUsuarioFinal(); if (txtUsuario1.Text == "Dario Borja Gamboa" && txtContrasena1.Text == "202110395") { form2.ShowDialog(); } else if (txtUsuario1.Text == "Tomás Betancur Delgado" && txtContrasena1.Text == "202210279") { form2.ShowDialog(); } else if (txtUsuario1.Text == "Samuel Bastidas Pamplona" && txtContrasena1.Text == "202210320") { form2.ShowDialog(); } else if (txtUsuario1.Text == "Andrés Camilo Vélez Moncada" && txtContrasena1.Text == "202020074") { form2.ShowDialog(); } else try { // Este Try-Catch verifica que en el caso de que el visitante ponga un string con forma de correo, lo deje ingresar al formulario correspondiente string patronCorreo = @"^[^@\s]+@[^@\s]+\.[^@\s]+\$"; if (!Regex.IsMatch(txtUsuario1.Text, patronCorreo)) throw new Exception("El formato del correo electrónico es inválido."); form3.ShowDialog(); } catch (Exception ex) { MessageBox.Show(ex.Message, "Error de validación", MessageBoxButtons.OK, MessageBoxIcon.Error); }

## Respuesta De La Herramienta:

Sí, puedes simplificar tu código utilizando un **HashMap** (o en el caso de C#, un Dictionary) para gestionar las combinaciones de usuario y contraseña. Esto hará que el código sea más limpio y fácil de mantener, en lugar de utilizar múltiples if-else. Aquí te muestro cómo podrías hacerlo:

```
Implementación con Dictionary en C#:
using System;
using System.Collections.Generic;
using System.Text.RegularExpressions;
using System.Windows.Forms;
public partial class Form1: Form
{
 // Creación de un Dictionary para almacenar usuario y contraseña
  private Dictionary<string, string> usuarios = new Dictionary<string, string>
 {
   { "Dario Borja Gamboa", "202110395" },
   { "Tomás Betancur Delgado", "202210279" },
   { "Samuel Bastidas Pamplona", "202210320" },
   { "Andrés Camilo Vélez Moncada", "202020074" }
 };
  public Form1()
 {
   InitializeComponent();
 }
 private void btnLogin_Click(object sender, EventArgs e)
 {
   FrmPersonal form2 = new FrmPersonal();
```

```
// Validar si el usuario existe en el diccionario y la contraseña es correcta
   if (usuarios.ContainsKey(txtUsuario1.Text) && usuarios[txtUsuario1.Text] ==
txtContrasena1.Text)
   {
     form2.ShowDialog();
   }
   else
   {
     try
     {
       // Verificación de formato de correo
       string patronCorreo = @"^[^@\s]+@[^@\s]+\.[^@\s]+$";
       if (!Regex.lsMatch(txtUsuario1.Text, patronCorreo))
        throw new Exception("El formato del correo electrónico es inválido.");
       form3.ShowDialog();
     }
     catch (Exception ex)
       MessageBox.Show(ex.Message, "Error de validación", MessageBoxButtons.OK,
MessageBoxIcon.Error);
     }
   }
```

FrmUsuarioFinal form3 = new FrmUsuarioFinal();

}

}