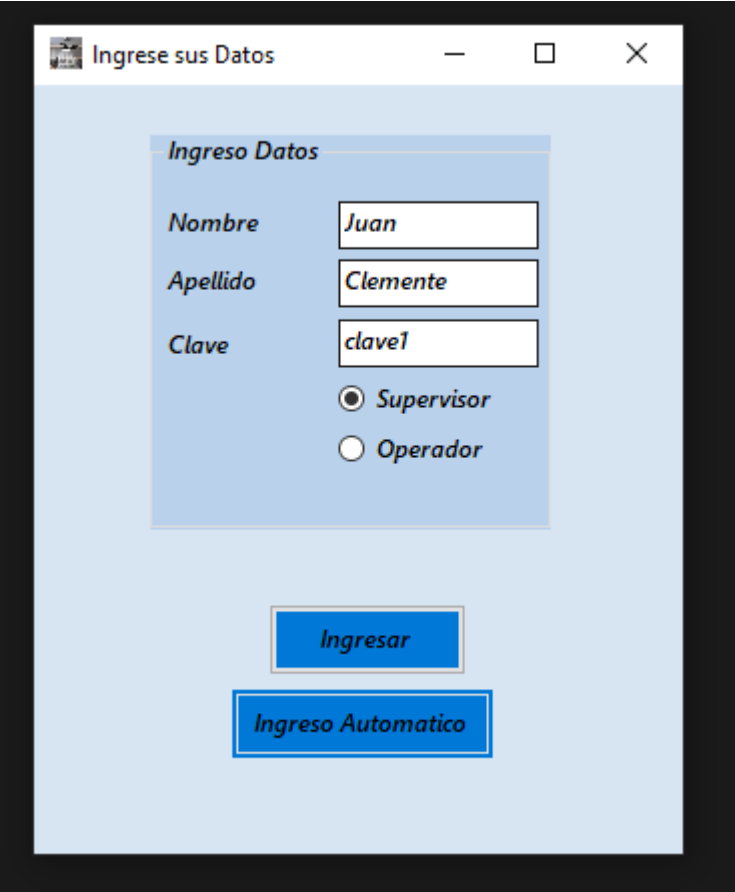


En este proyecto de Fábrica se podrán crear dos productos, los cuales serán:

- Mueble
- Ventilador

Al iniciar nuestro programa lo primero que veremos es un página para poder ingresar nuestros datos. El ingreso automático rellenará todos los campos de la siguiente manera, permitiéndonos entrar a la fábrica en modo supervisor.



The screenshot shows a window titled "Ingrese sus Datos" with a light blue background. Inside, there is a form titled "Ingreso Datos" with the following fields and options:

- Nombre:** Text input field containing "Juan".
- Apellido:** Text input field containing "Clemente".
- Clave:** Text input field containing "clave1".
- Role Selection:** Two radio buttons. The first is selected and labeled "Supervisor". The second is unselected and labeled "Operador".

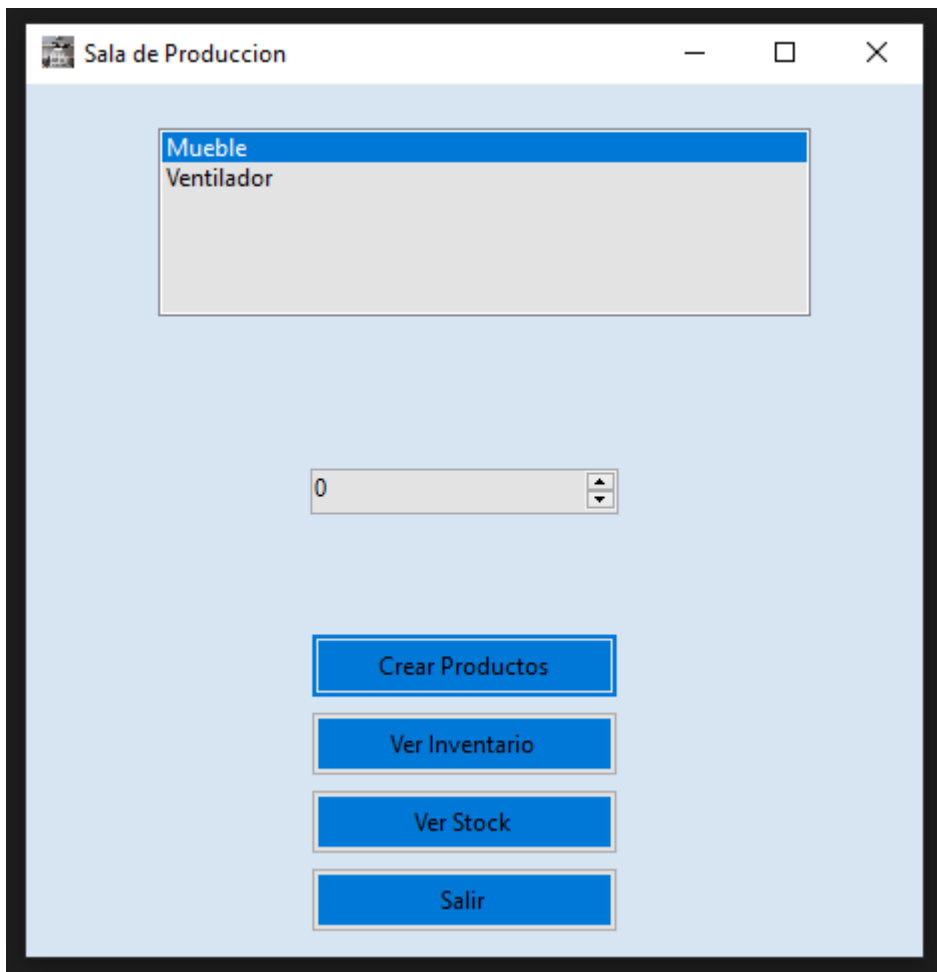
Below the form, there are two blue buttons with white text:

- Ingresar** (top button)
- Ingreso Automatico** (bottom button)

Después de esto, pasaremos a la pestaña de producción, en la que podremos escoger cual producto y en qué cantidad crearlo.

- Al presionar **"CREAR PRODUCTO"** nos saldrá una confirmación de si se pudo realizar la creación del producto en la cantidad deseada.
- Al presionar **"VER INVENTARIO"** podremos ver otro formulario con los datos de la materia prima actuales, en dicha ventana también tendremos la posibilidad de rellenar en 15 unidades cada elemento de nuestro inventario.

- Al presionar “**VER STOCK**” nos aparecerá un anuncio en el que muestra un breve listado de los productos que hemos logrado crear.
- “**SALIR**” finalmente con este botón volveremos a la página de login.



Clase Estáticas: Inventario y Stock.

```

23 referencias
public static class Inventario
{
    16 referencias
    public static Dictionary<string, int> Materiales { get; set; }

    0 referencias
    static Inventario()
    {
        Materiales = new Dictionary<string, int>();
        Materiales["madera"] = 80;
        Materiales["tela"] = 50;
        Materiales["aspa"] = 30;
        Materiales["tornillo"] = 200;
        Materiales["base"] = 25;
        Materiales["bombilla"] = 27;
    }
}

```

Herencia: Utilizada en clases employee y mueble/ventilador.

Colecciones: listas y diccionarios.

Enumerados: Dos.

```
5 referencias
public enum EmployeeRole { Operator, Supervisor }
4 referencias
public enum ProductStatus { Available, NotAvailable}
```

Conversiones implícitas/expícitas:

```
int cant = (int)this.numericUpDown1.Value;
```

Poliformismo:

```
2 referencias
public override string Mostrar()
{
    StringBuilder sb = new();
    sb.Append(base.Mostrar());
    sb.AppendLine($" {Puesto} #{Id}");
    return sb.ToString();
}
0 referencias
public static bool operator ==(Employee a, uint id)
{
    return a.id == id;
}
0 referencias
public static bool operator !=(Employee a, uint id)
{
    return !(a.id == id);
}
```

```
3 referencias
public abstract class User
{
    protected string password;
    protected string name;
    protected string surname;
```