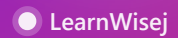


Navegación reutilizable y permisos

Las cuatro vistas que vas a construir, un único shell que gestiona la navegación y una primera aproximación a los permisos.



Objetivo de la lección

Conoce el objetivo que construirás en el laboratorio: un shell reutilizable cuya área de contenido alterna entre cuatro vistas (Dashboard, Tickets, Customers y Settings) mientras la cabecera, la navegación y la barra de estado siguen iguales.

1. Las cuatro vistas

El shell exterior nunca cambia. Cada botón de navegación carga un **UserControl** distinto en el panel de contenido.

Vista	Qué muestra
DashboardView	Tarjetas de resumen (Open Tickets, Customers, System Status) y una acción rápida como <i>Run Health Check</i> .
TicketsView	Una lista de tickets de soporte con las acciones <i>New Ticket</i> y <i>Close Selected</i> .
CustomersView	Una lista de clientes con las acciones <i>Add Customer</i> y <i>View Selected</i> .
SettingsView	El rol actual y un botón <i>Save Settings</i> que introduce el comportamiento de los permisos.



Por qué UserControls

Poner cada página en su propio UserControl mantiene la interfaz repetida organizada y reutilizable, y permite que el shell siga siendo pequeño: solo se encarga de la navegación y del panel de contenido.

2. Un único shell gestiona la navegación

Los nombres exactos pueden cambiar, pero la idea es la misma: el shell crea la vista adecuada, la acopla para que rellene el panel de contenido y actualiza la ruta de navegación y el estado.

```
// Main shell owns the navigation buttons and the content panel.
public partial class MainPage : Page
{
    private string currentRole = "Support Agent";

    public MainPage()
    {
        InitializeComponent();
        NavigateTo("Dashboard");
    }
}
```

```

        ApplyPermissions();
    }

    private void NavigateTo(string pageName)
    {
        pnlContent.Controls.Clear();

        UserControl view;
        switch (pageName)
        {
            case "Tickets":    view = new TicketsView();    break;
            case "Customers":  view = new CustomersView(); break;
            case "Settings":   view = new SettingsView(currentRole); break;
            default:           view = new DashboardView(); break;
        }

        view.Dock = DockStyle.Fill;
        pnlContent.Controls.Add(view);
        lblBreadcrumb.Text = "Home / " + pageName;
        lblStatus.Text = DateTime.Now.ToString("hh:mm:ss tt") + " - Opened " + pageName + "
page.";
    }

    private void ApplyPermissions()
    {
        // Beginner version: a Support Agent can view Settings but not save.
        // Later this grows into real authentication and authorization.
    }
}

```

3. Settings de solo lectura: el primer permiso

La vista Settings es donde los permisos se hacen visibles. Para un **Support Agent**, muestra los ajustes pero deshabilita el guardado: el botón *Save Settings* aparece en gris con una nota como «*Settings are view-only for this role.*». A un Manager sí se le permitiría guardar. Así se introducen los permisos **antes** de que exista cualquier sistema de inicio de sesión.

4. Lista de comprobación del Designer

- Usa **Dock** o **Anchor** en la cabecera, la navegación a la izquierda, el panel de contenido y la barra de estado.
- Quita el botón de minimizar de una Window/Form1, salvo que construyas una barra de tareas o un área de restauración.
- Prefiere una **Page** cuando el shell deba ocupar todo el navegador como una aplicación web normal.
- Mantén la lógica de navegación centralizada en el shell en lugar de duplicarla en cada página.
- Usa UserControls para el contenido repetido: **DashboardView**, **TicketsView**, **CustomersView**, **SettingsView**.



Ya estás listo para el laboratorio

A continuación: mira el recorrido guiado del shell de navegación, supera la evaluación de conocimientos y después construye el shell de navegación reutilizable: cabecera, cuatro botones de navegación, un área de contenido que cambia, un registro de estado y una vista Settings de solo lectura para un Support Agent.