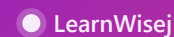


Diseñar el shell de la aplicación

Diseña la estructura exterior de una aplicación real (cabecera, navegación, área de contenido y barra de estado) y haz que se redimensione correctamente.



Objetivo de la lección

Pasa de pantallas sueltas a un **shell de aplicación** sencillo: el diseño exterior compartido (cabecera, navegación, área de contenido, barra de estado) que permanece en pantalla mientras cambia el contenido.

1. Qué es un shell de aplicación

Un **shell de aplicación** es la estructura exterior compartida de una aplicación real. El shell permanece en su sitio; solo cambia el área de contenido a medida que el usuario se desplaza. Un shell sencillo tiene cuatro partes:

Parte del shell	Propósito	Ejemplo para principiantes
Cabecera	Nombre de la aplicación, usuario actual y ruta de navegación.	ServiceDesk Application · Signed in as: Support Agent
Navegación a la izquierda	Las páginas que el usuario puede abrir.	Dashboard, Tickets, Customers, Settings
Área de contenido	Muestra el contenido de la página seleccionada.	Tarjetas del dashboard o una lista de tickets
Área de estado	Muestra la acción reciente o información del sistema.	03:26 PM - Application shell loaded.

2. Page frente a Window/Form1

Wisej.NET te permite construir el shell como una **Page** o como una **Window/Form1**. Cada una crea una experiencia de usuario distinta.

Opción	Cómo se percibe	Cuándo usarla
Page	Una página web que ocupa todo el marco del navegador.	La mejor opción para una aplicación de formación moderna o un shell de dashboard.
Window/Form1	Una ventana de estilo escritorio dentro del navegador que se puede mover, minimizar, maximizar y cerrar.	Adecuada para herramientas de estilo escritorio, ventanas emergentes o cuando quieres intencionadamente un comportamiento de ventana.



Hábito importante de interfaz

Si construyes el shell como una Window/Form1, **quita o deshabilita el botón de minimizar**, salvo que construyas también una barra de tareas o una forma clara de restaurarla. Una ventana minimizada en el navegador puede dar la impresión de que la aplicación ha desaparecido, lo que confunde a los principiantes.

3. El anclaje y el acoplamiento mantienen el diseño utilizable

El **anclaje** (Anchor) indica a un control a qué lados de su contenedor debe permanecer unido. El **acoplamiento** (Dock) hace que un control ocupe un lado o el espacio restante. Juntos mantienen el shell utilizable cuando se redimensiona el navegador.

Área del control	Diseño recomendado	Por qué
Panel de cabecera	<code>Dock = Top</code> (o Anchor Top + Left + Right)	La cabecera debe extenderse a lo largo de la parte superior.
Panel de navegación a la izquierda	<code>Dock = Left</code> (o Anchor Top + Bottom + Left)	La navegación permanece a la izquierda y conserva su altura.
Panel de contenido	<code>Dock = Fill</code> (o Anchor en los cuatro lados)	La página actual usa el espacio restante.
Etiqueta o panel de estado	<code>Dock = Bottom</code> (o Anchor Bottom + Left + Right)	Los mensajes de estado permanecen visibles en la parte inferior.
Botones de navegación	Anchor Top + Left + Right dentro del panel de navegación	Los botones mantienen un ancho limpio y uniforme.

4. Navega desde un único sitio

No disperses la lógica de cambio de página por todos los botones. En su lugar, cada botón llama a **un único método de navegación compartido**. Así la aplicación es fácil de depurar y de cambiar más adelante.

```
private void btnDashboard_Click(object sender, EventArgs e)
{
    NavigateTo("Dashboard");
}

private void btnTickets_Click(object sender, EventArgs e)
{
    NavigateTo("Tickets");
}

private void NavigateTo(string pageName)
{
    pnlContent.Controls.Clear();
    if (pageName == "Dashboard")
        pnlContent.Controls.Add(new DashboardView() { Dock = DockStyle.Fill });
    else if (pageName == "Tickets")
        pnlContent.Controls.Add(new TicketsView() { Dock = DockStyle.Fill });

    lblBreadcrumb.Text = "Home / " + pageName;
    lblStatus.Text = DateTime.Now.ToString("hh:mm:ss tt") + " - Opened " + pageName + "
page.";
}
```

Un solo método vacía el panel de contenido, carga la vista adecuada, actualiza la **ruta de navegación** ([Home](#) / [Tickets](#)) y escribe una línea en el área de **estado**.

5. Los permisos como concepto de diseño

Los permisos no tienen por qué empezar como un sistema de inicio de sesión completo. Pueden empezar como una simple idea de **rol**: un Support Agent puede *ver* Settings pero no guardar cambios. Más adelante esto se convierte en autenticación y autorización reales.

Rol	Dashboard	Tickets	Customers	Settings
Support Agent	Ver	Crear/cerrar	Ver	Solo lectura
Manager	Ver	Crear/cerrar	Crear/editar	Guardar ajustes



Comprobación final

Sabes identificar las áreas de cabecera, navegación, contenido y estado; usar anclaje o acoplamiento en lugar de posiciones fijas; explicar en qué se diferencian Page y Window/Form1; y encaminar cada botón de navegación a través de un único método de navegación compartido.