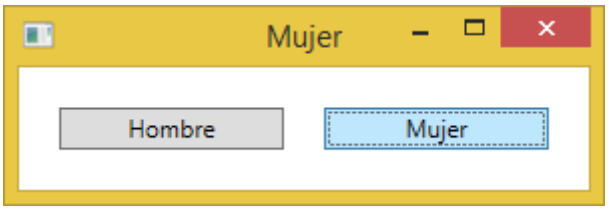
EJERCICIOS 1
Ejercicios básicos

NOTA: Intenta reproducir las capturas de pantalla de forma lo más fidedigna posible.

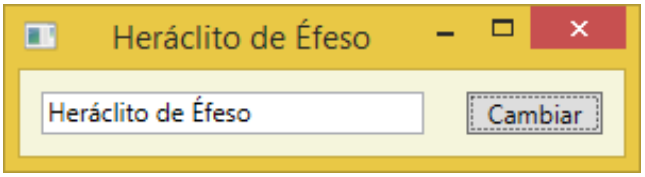
1. Implementa una aplicación que disponga de dos botones. Al pulsar uno de los botones, se cambiará el título de la ventana usando el texto del botón pulsado.



   Para ello, implementa el manejador de evento `Click` de modo que se cambie el título de la ventana usando directamente el contenido (`Content`) del botón pulsado.

2. Modifica el ejercicio anterior para hacerlo creando un `Binding` entre el título y la propiedad Content del botón pulsado. Esto implica que tendremos que actualizar el `DataContext` cada vez que pulsemos un botón con la instancia del botón pulsado.

3. Implementa una ventana WPF que permita insertar el nombre de un usuario en una caja de texto. El texto que se inserta se mostrará en el título de la ventana una vez que se haya pulsado el botón *Cambiar*.



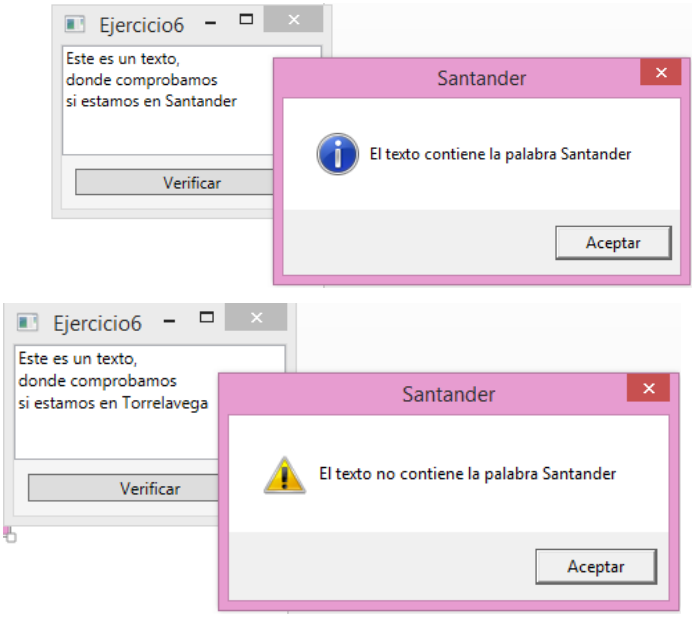
4. Implementa la siguiente ventana WPF. La aplicación contiene 3 imágenes en la parte superior y una caja de texto (no editable) en la parte inferior.

Al pasar el ratón sobre las imágenes de la parte superior (evento `MouseEnter`), se mostrará en la parte inferior el nombre del personaje correspondiente. Este dato se averiguará a través de la propiedad `Name` de la imagen.

Utiliza recursos estáticos para definir las rutas de acceso a las imágenes.

5. Modifica el ejercicio anterior para utilizar un *data binding* entre la caja de texto y la imagen sobre la cual pasamos el ratón.
    a. En el manejador del evento `MouseEnter` tendrás que cambiar `DataContext` para que sea la imagen sobre la cual has pasado el ratón.

6. Implementa una interfaz con una caja de texto y un botón. Al pulsar el botón se analizará si el texto contiene la palabra "Santander". Tanto si la contiene como si no se mostrará un mensaje:



7. Realiza la siguiente ventana. Está formado por 3 ComboBox con los números del 0 al 255, acompañados de sus correspondientes etiquetas y un botón. El valor por defecto de los combos es 0. Al pulsar el botón, el color de fondo del mismo se establecerá con el color indicado por las listas.

NOTA: Para establecer el color de fondo puedes hacerlo con la propiedad ***`Background`****, que recibe un valor de tipo Brush que representa un color. En concreto, la clase* ***`SolidColorBrush`****, que hereda de Brush, se usa para representar colores lisos.*

`SolidColorBrush` *recibe en el constructor un valor de tipo Color. Para generar un color a partir de los valores RGB en decimal usamos el método estático* ***`Color.FromRGB(byte rojo, byte verde, byte azul)`****, que genera el color correspondiente.*

8. Modifica el programa anterior para que el color se establezca automáticamente cada vez que cambiamos el valor de una de las listas. Haz también que el valor se pueda introducir manualmente (escribiendo en lugar de seleccionando).

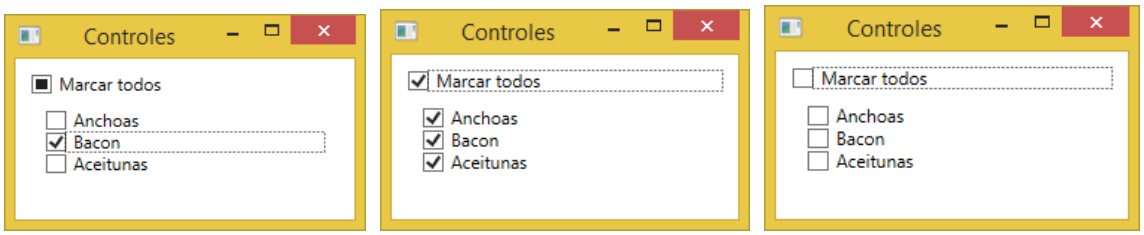
9. Implementa una ventana WPF que tenga el siguiente aspecto:

La aplicación tiene un grupo de botones de radio en la parte izquierda y una etiqueta en la parte derecha, que tiene como icono la imagen de una bandera. Al cambiar el botón de radio, cambiará la imagen de la bandera y la de la barra de título.

Intenta aprovechar el hecho de que los nombres de los países están en inglés, haciendo que coincidan con el nombre del componente.

10. Un **CheckBox** habitualmente se corresponde con un valor booleano, que tiene dos estados: true o false. Estableciendo la propiedad **IsThreeState** a **true**, el **CheckBox** puede tener un tercer estado llamado *estado indeterminado*.

Un uso común de esta propiedad es tener una casilla de *Marcar todo*, con la cual podemos controlar un grupo de casillas hijas y comprobar su estado colectivo:



Será necesario utilizar dos manejadores de eventos:

- El manejador **cbTodos_CheckedChanged** se encarga de los cambios de estado en la casilla que permite habilitar/deshabilitar el resto de casillas.
- El manejador **cbIngrediente_CheckedChanged** se encarga de procesar los cambios de estado en el resto de casillas.

Cuando marcamos o desmarcamos la casilla *Marcar todos*, el resto de casillas se marcan o desmarcan respectivamente. Cuando marcamos o desmarcamos alguna casilla de ingredientes ocurre el suceso contrario: Si todas están marcadas se marca la casilla *Marcar Todos* y si todas están marcadas se desmarca dicha casilla. En caso de que sólo alguna esté marcada, se dejará con el valor **null**, que fuerza el **CheckBox** al estado indeterminado (se muestra con un cuadrado).