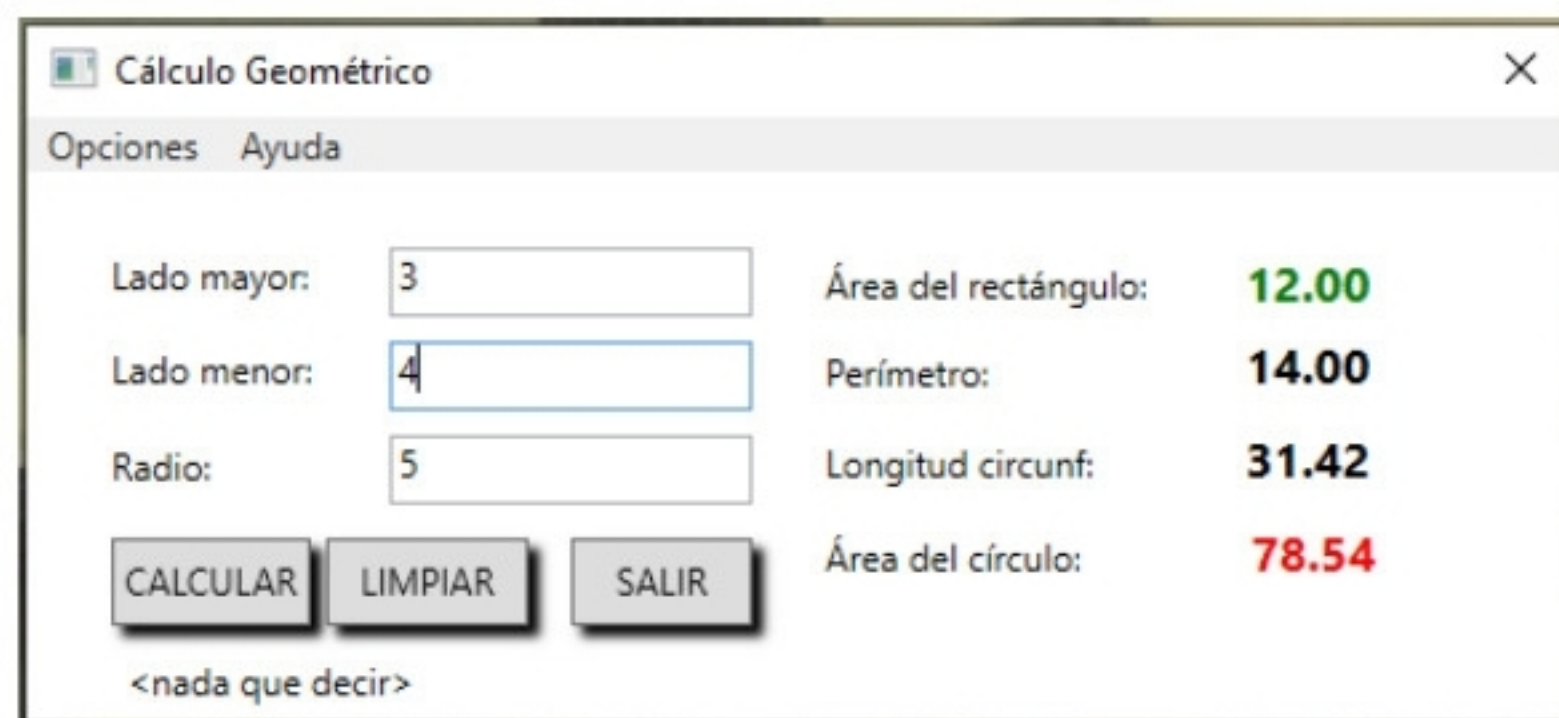


Realizar un proyecto WPF con el nombre 'Geometria\_MVVM' para implementar el cálculo de la longitud de una circunferencia, el área del círculo y el perímetro y área de un rectángulo. La interfaz de usuario deberá cumplir con las siguientes pantallas:

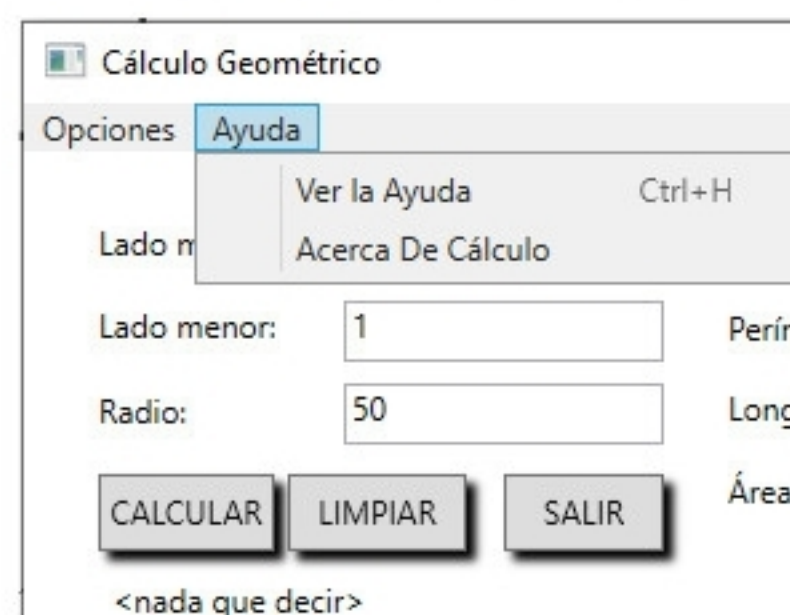
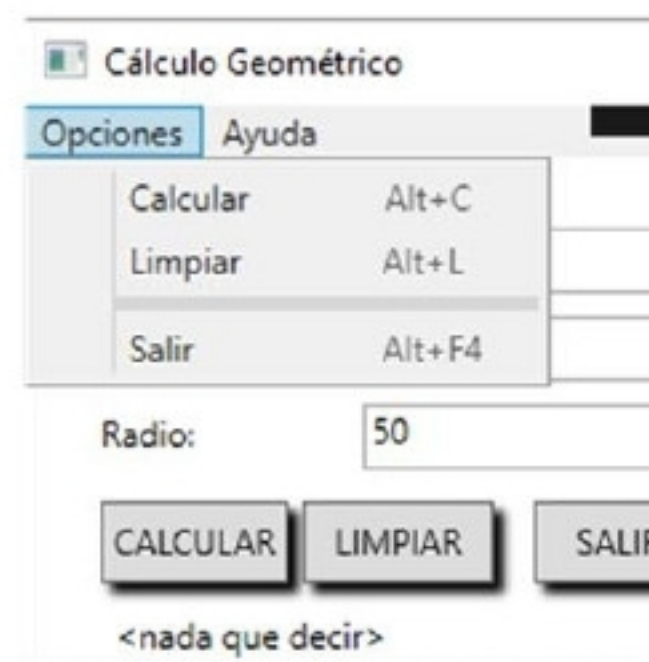


## Ventana Principal

### CONDICIONES A CUMPLIR:

#### Archivo winGeometria.xaml (2 puntos)

- Implementa la IU principal con un menú superior.
- No debe contener código behind, salvo el mínimo necesario para abrir las ventanas de Ayuda y AcercaDe
- No debe controlar la validez de los datos de entrada, en caso de introducir datos erróneos son las propiedades las responsables. En caso de error, se actualizará de forma automática la información de la etiqueta <nada que decir>, vinculada a una propiedad llamada **Mensaje**, de tipo string.
- Todos los botones realizan su acción a través de un Command.
- Todos los valores calculados deben tener 2 decimales exactamente.
- Todos los valores calculados se actualizan automáticamente al cambiar algún dato de entrada, o cuando se pulse el botón CALCULAR. La etiqueta <nada que decir> mostrará **<Datos recalculados>**
- El botón LIMPIAR deja vacías todas las entradas de texto, todos los cálculos se quedan a 0 en consecuencia. La etiqueta <nada que decir> mostrará **<Datos restaurados>**
- El área del rectángulo cambia de color de fuente (Foreground) según la escala de valores siguientes:
  - Entre 0 y 15 -> Green
  - Entre 15 y 20 -> Yellow
  - Mayor de 20 -> Red
- El área del círculo cambia de color de fuente (Foreground) según la escala siguiente:
  - Entre 0 y 15 -> Green
  - Entre 15 y 50 -> Yellow
  - Mayor de 50 -> Red



#### Archivo GeometriaMV.cs (2 puntos)

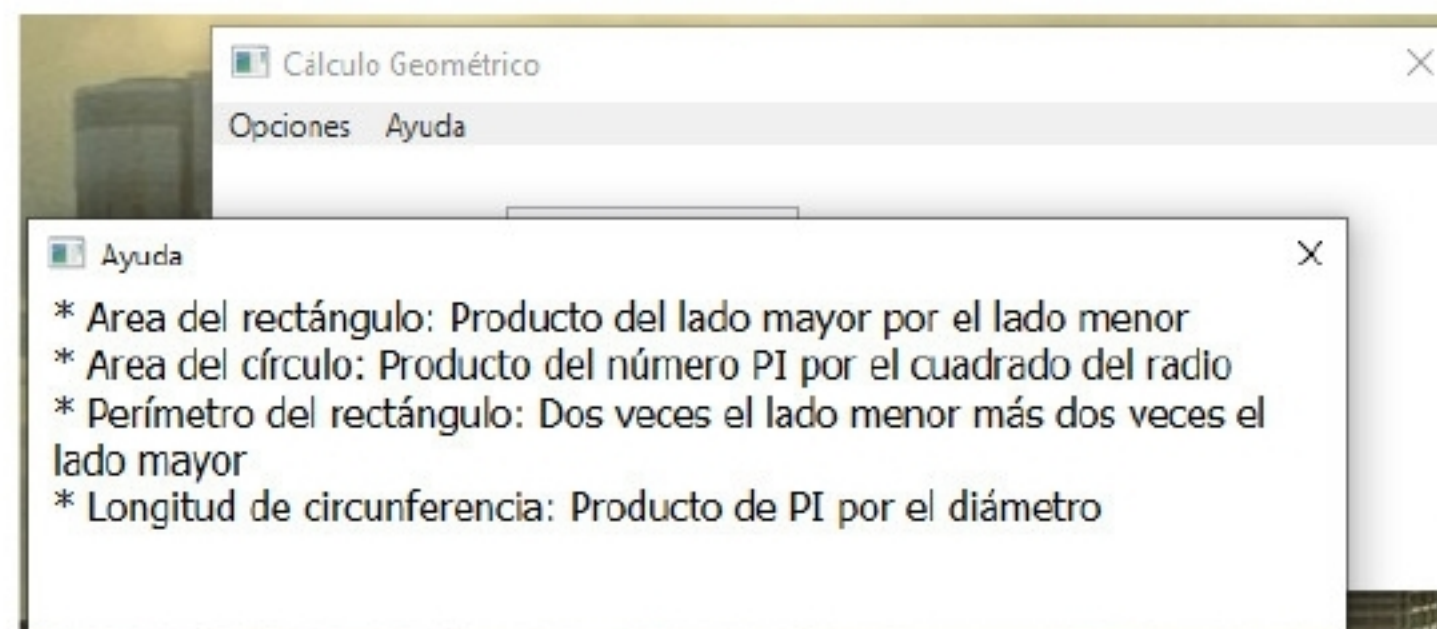
- Gestiona el Modelo-Vista (reglas de negocio) con la clase GeometriaMV. Hereda de la clase Notificador
- Utiliza un objeto de la clase Geometria para proporcionar todos los bindings que necesita la IU. Se encarga de notificar a la IU de cualquier cambio en las propiedades enlazadas.
- Cada vez que recoge un valor de la IU, intenta asignarlo a la propiedad correspondiente de la clase Geometria. Si ésta le devuelve una excepción, asigna a la etiqueta **Mensaje = String.Format("Error: {0}", e.Message)** con la información devuelta por la excepción.



## Archivo Geometria.cs (1 punto)

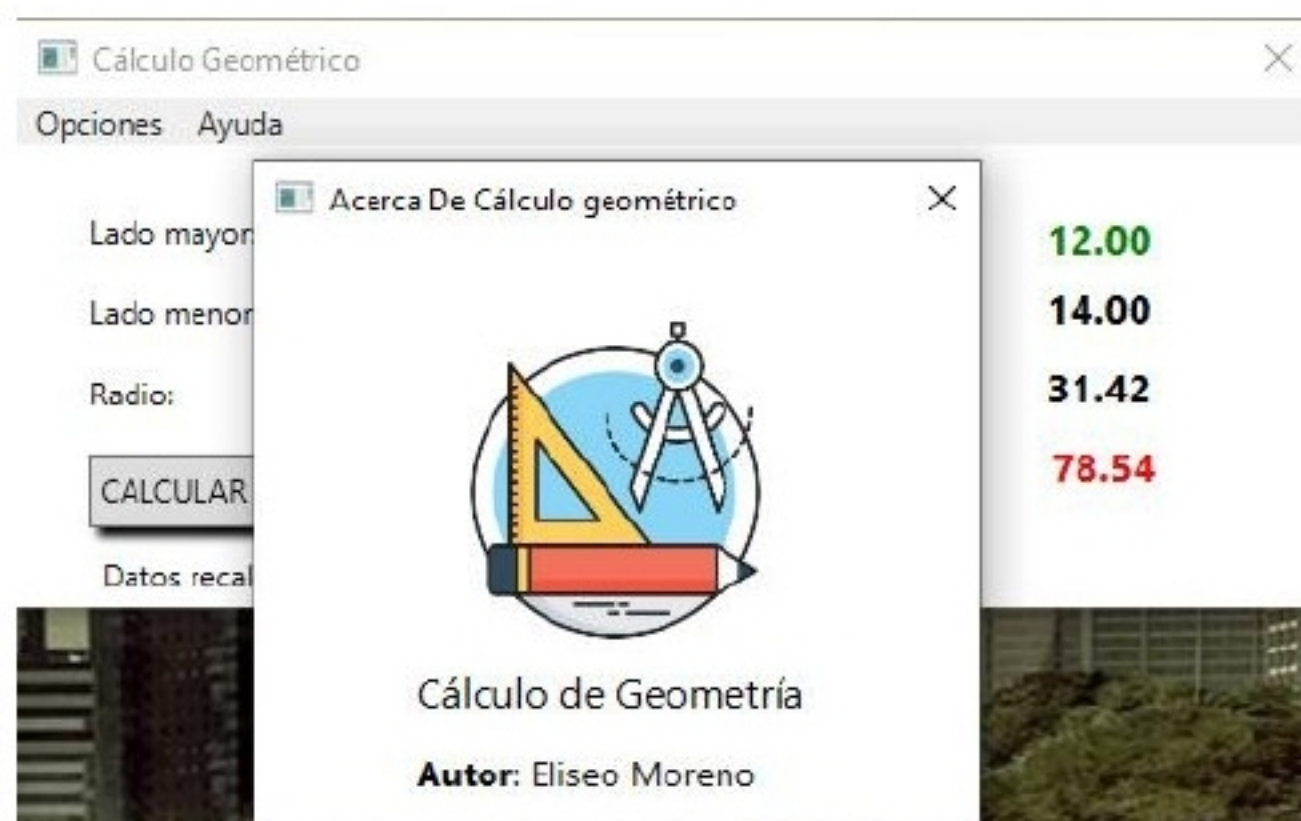
- Gestiona el Modelo con la clase Geometria. Se encarga de realizar todos los cálculos geométricos.
- Implementa las propiedades LadoMayor, LadoMenor, Radio, Perimetro, AreaRectangulo, LongitudCircunferencia y AreaCirculo, todas de tipo **double nullable**. Estas propiedades solo son visibles para la clase de negocio, nunca para la IU.
- Las propiedades calculadas (Perimetro, AreaRectangulo, LongitudCircunferencia y AreaCirculo) son de solo lectura.
- En caso de recibir un valor negativo en alguna de sus propiedades (lados o radio), debe lanzar **Exception("Valor negativo")**;
- En caso de recibir un valor null, deben devolver 0.

## Archivo Ayuda.xaml (0.5 puntos)



- Implementa la ventana **modal** que aparece al pulsar la opción correspondiente del menú. Únicamente necesita un objeto TextBlock dentro del Grid.
- Abre y lee el archivo **ayuda.txt** que debe estar en la carpeta **recursos** del proyecto. Puedes usar un objeto **StreamReader** para recorrerlo completo tras asegurarte que existe el archivo en dicha ruta. El código puede ir en el code-behind.

## Archivo AcercaDe.xaml (0.5 puntos)



Implementa la ventana **modal** que aparece al pulsar la opción correspondiente del menú.

Necesita la imagen **geometria.png**, recogida de la carpeta de recursos del proyecto, y dos Textblock, para mostrar el nombre del programa y el nombre del autor (tu nombre y apellidos), respectivamente.