

**Nombre:**.....

**Fecha: 10/12/2025**

Acabas de incorporarte a un equipo de desarrollo y te han asignado una tarea urgente: rescatar el proyecto "Task Manager".

El desarrollador anterior dejó la aplicación a medias. Aunque el código compila y la ventana se abre, la aplicación no funciona (los botones no responden, la lista no se actualiza) y el diseño es visualmente pobre.

#### **BLOQUE A: Reparación de Arquitectura MVVM (6 Puntos)**

##### **1. Modelos Reactivos (1,5 Ptos)**

La clase TaskModel tiene propiedades simples que no avisan a la interfaz cuando cambian.

Implementa la interfaz INotifyPropertyChanged en la clase TaskModel.

Asegúrate de que las propiedades Title e IsCompleted notifiquen sus cambios.

##### **2. Colecciones Observables (1,0 Pto)**

Actualmente, la lista de tareas en MainViewModel es una List<TaskModel> estática.

Sustitúyela por el tipo de colección adecuado de WPF para que la interfaz gráfica se entere automáticamente cuando se añaden o eliminan elementos.

##### **3. Contexto y Binding (1,5 Ptos)**

La Vista y el ViewModel no están conectados, y el formulario de entrada falla.

Asigna correctamente el DataContext de la MainWindow (puedes hacerlo en XAML o en el constructor del Code-Behind\*).

Corrige el Binding del TextBox de "Nueva Tarea". Actualmente usa Mode=OneWay, lo que impide que el ViewModel reciba lo que escribe el usuario. Cámbialo al modo correcto.

##### **4. Lógica de Comandos (2,0 Ptos)**

El AddTaskCommand está vacío. Implementa su lógica en el ViewModel:

- Execute: Debe crear una nueva instancia de TaskModel con el texto introducido, añadirla a la lista de tareas y limpiar la propiedad NewTaskTitle para vaciar el input.

\*Code-Behind se refiere al archivo XXXX.xaml.cs

Nombre:.....

### BLOQUE B: Diseño de Interfaces y XAML (4 Puntos)

#### 5. DataTemplates (2,0 Ptos)

La lista actual (ListBox) muestra el nombre de la clase (MyTasksApp.Models...) en lugar de los datos.

Define un DataTemplate para los ítems de la lista.

Cada ítem debe mostrar:

- Un Boton con un texto y color distinto en función a la propiedad IsCompleted.
- Un TextBlock bindeado a la propiedad Title.

#### 6. Estilos y Triggers Visuales (2,0 Ptos)

Debes arreglar la interfaz para que se vea como las de las imágenes inferiores.

La versión sencilla vale 0,5 Ptos.

### Task Manager DAM

Nueva tarea

Agregar

Aprobar el examen	Aprobar el examen	Aprobar el examen
Aprobar el examen <a href="#" style="border: 1px solid #ccc; padding: 5px 10px; text-decoration: none; color: inherit;">Completar</a>	Aprobar el examen <a href="#" style="border: 1px solid #ccc; padding: 5px 10px; text-decoration: none; color: inherit;">Completar</a>	Aprobar el examen <a href="#" style="border: 1px solid #ccc; padding: 5px 10px; text-decoration: none; color: inherit;">Completar</a>
Aprobar el examen	Aprobar el examen	Aprobar el examen
Aprobar el examen <a href="#" style="border: 1px solid #ccc; padding: 5px 10px; text-decoration: none; color: inherit;">Completar</a>	Aprobar el examen <a href="#" style="border: 1px solid #ccc; padding: 5px 10px; text-decoration: none; color: inherit;">Completar</a>	Aprobar el examen <a href="#" style="border: 1px solid #ccc; padding: 5px 10px; text-decoration: none; color: inherit;">Completar</a>

Nombre:.....

Para la versión avanzada:

Queremos feedback visual cuando una tarea se completa.

Para ello, como puedes ver en la imagen, debes aplicar otro estilo para la tarea en caso de que esté completada, y también el texto debe aparecer tachado (TextDecorations).

### Task Manager DAM

Nueva tarea      Agregar

Aprobar  
el examen

Completar

Aprobar  
el examen

Completada

Al finalizar, comprime la carpeta de tu solución en un archivo .ZIP con el formato:  
APELLIDO1\_APELLIDO2\_NOMBRE zip

\*Code-Behind se refiere al archivo XXXX.xaml.cs