

1. Consulta a Modelos de Lenguaje (LLM):

Para abordar este problema, consulté a dos modelos de lenguaje (LLM) diferentes: GPT-4 y Claude 2. A continuación, se detallan los prompts utilizados y las respuestas obtenidas.

• Prompt 1 (Consultado a GPT-4):

En C#, tengo una clase Chofer donde el constructor valida el tipo de licencia según la edad del conductor. Sin embargo, si cambio la propiedad TipoLicencia durante la ejecución, no hay validación. ¿Cómo puedo asegurarme de que el tipo de licencia siempre sea válido, incluso después de modificarlo? ¿Cuál sería la mejor práctica para implementar esto?

- Mejor Práctica: Utilizar un método setter personalizado para la propiedad TipoLicencia. Este método debe incluir la lógica de validación necesaria.

- Implementación Sugerida: Definir la propiedad TipoLicencia como una propiedad con un cuerpo personalizado (set) que realice la validación antes de asignar el valor.

- Ejemplo de Código:

```
public class Chofer
{
    public int Edad { get; private set; }
    private string _tipoLicencia;
    public string TipoLicencia
    {
        get => _tipoLicencia;
        set
        {
            if (!ValidarLicencia(value, Edad))
                throw new ArgumentException("El tipo de licencia no es válido
para la edad del conductor.");
            _tipoLicencia = value;
        }
    }
}
```

```
}  
  
public Chofer(int edad, string tipoLicencia)  
{  
    Edad = edad;  
    TipoLicencia = tipoLicencia; // Usa el setter para validar  
}  
  
private bool ValidarLicencia(string tipoLicencia, int edad)  
{  
    // Ejemplo de validación: Licencia tipo "A" solo para mayores de 18  
    años  
    return tipoLicencia switch  
    {  
        "A" => edad >= 18,  
        "B" => edad >= 21,  
        _ => false  
    };  
}  
}
```

- **Prompt 2 (Consultado a Claude 2):**

Estoy trabajando en C# y tengo una clase Chofer. En el constructor, valido el tipo de licencia según la edad. Sin embargo, si cambio la propiedad `TipoLicencia` más adelante, no hay validación. ¿Cómo puedo implementar una validación para evitar que se asigne un tipo de licencia inválido?"

- Recomendación: Usar encapsulamiento para controlar el acceso a la propiedad `TipoLicencia`. Esto implica definir la propiedad como privada y proporcionar métodos públicos para modificarla con validación.

- Alternativa: Implementar un patrón de diseño como `Observer` o `Property Change Notification` si se requiere notificar cambios en tiempo real.

- Ejemplo de Código:

```
public class Chofer
{
    public int Edad { get; private set; }
    private string _tipoLicencia;
    public string TipoLicencia
    {
        get => _tipoLicencia;
        private set
        {
            if (!EsLicenciaValida(value))
                throw new InvalidOperationException("La licencia no es válida
para la edad actual.");
            _tipoLicencia = value;
        }
    }
    public Chofer(int edad, string tipoLicencia)
    {
        Edad = edad;
        TipoLicencia = tipoLicencia; // Validación en el constructor
    }

    public void CambiarLicencia(string nuevaLicencia)
    {
        TipoLicencia = nuevaLicencia; // Usa el setter privado
    }
}
```

```
private bool EsLicenciaValida(string licencia)
{
    return licencia switch
    {
        "A" => Edad >= 18,
        "B" => Edad >= 21,
        _ => false
    };
}
```