



UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

Ingeniería en Sistemas y Ciencias de la Computación

Bonus Challenge

Estudiante:

Juan Luis Molina Orellana –0905-24-8364

Programacion I

Jutiapa, Marzo 2025



LLM 1: Grok 3

Prompt: "En C#, tengo una clase Chofer con una propiedad TipoLicencia (char) y una validación en el constructor que verifica la edad mínima según el tipo de licencia (ej. 'C' requiere 18 años, 'A' requiere 23). Si cambio TipoLicencia después de crear el objeto, no se valida. ¿Cómo hago para validar siempre que cambie esta propiedad? ¿Cuál es la mejor práctica?"

Respuesta de LLM: Grok sugirió usar un setter personalizado en la propiedad TipoLicencia que llame a un método privado ValidarLicencia con la misma lógica del constructor. Esto asegura que cada asignación sea validada. Es simple, encapsulado y reutiliza el código.

LLM 2: ChatGPT

Prompt: "En C#, tengo una clase Chofer con una propiedad TipoLicencia (char) y una validación en el constructor que verifica la edad mínima según el tipo de licencia. Si cambio TipoLicencia después, no se valida. ¿Cómo puedo validar cada cambio? y ¿Cuál es la mejor práctica?"

Respuesta de LLM: ChatGPT propuso usar un setter con validación y agregar INotifyPropertyChanged para notificar cambios, llamando a ValidarLicencia en el setter y constructor. Es más complejo pero útil para aplicaciones grandes con UI.

Mejor forma de hacerlo:

La mejor práctica para mi caso es la solución de Grok: un setter personalizado con un método ValidarLicencia. Es simple, no agrega overhead innecesario como eventos, y mantiene la validación consistente entre el constructor y los cambios posteriores. La versión con INotifyPropertyChanged es más adecuada para aplicaciones con interfaces gráficas, pero no la necesito en mi programa actual.