



Universidad Mariano Gálvez de Guatemala

Facultad ingeniería

Ingeniería en sistemas

Curso: Programación 1

Tarea: Validaciones de clase (Bonus Challenge)

Autor: Eduardo Gabriel Visoni Morales

0905-22-1146

Jutiapa, 28 de marzo de 2025

Bonus Challenge:

Lo que hice fue implementar 2 LMMs al programa Visual Studio para que fuera más fácil para el LMM analizar el código e implementarlo de una manera más realista, y que fuera más fácil para mi (para hacer mi prompt) y poder así entender también como funciona en mi código, y no solamente solucionar el problema sin saber el por qué.

LMMAs usados:

- Codeium
- Tabnine

Prompt usado:

“Si ves en este proyecto sobre Programación Orientada a Objetos, en lenguaje C#, en el constructor de la clase chofer se define el "tipo de licencia" y es validado con la edad, eso quiere decir que, si la persona tiene la edad mínima para tener esa licencia, es válido. Pero ¿qué pasa si en el transcurso del programa se cambia la propiedad tipo de licencia? No hay ningún método que lo pueda validar, explícame como puedo arreglar eso si en dado caso existe ese problema. Analiza mi Código, y dame la respuesta de como implementarías la solución. Dame la respuesta en español”

Codeium:

Respuesta: “Entiendo tu preocupación. En efecto, si el tipo de licencia se cambia después de la creación del objeto Chofer, no se vuelve a validar la edad. Para solucionar esto, te recomiendo implementar una propiedad con un setter que valide la edad cada vez que se cambie el tipo de licencia. Aquí te dejo un ejemplo de cómo podrías implementar esto: ”

```
public class Chofer
{
    // ...

    private Tipolicencia _tipolicencia;
    public Tipolicencia Tipolicencia
    {
        get { return _tipolicencia; }
        set
        {
            if (value != _tipolicencia)
            {
                if (!ValidarEdadParaLicencia(value))
                {
                    throw new InvalidOperationException("La edad no es válida para el tipo de licencia seleccionado");
                }
                _tipolicencia = value;
            }
        }
    }

    private bool ValidarEdadParaLicencia(Tipolicencia tipolicencia)
    {
        // Implementa la lógica para validar la edad según el tipo de licencia
        // ...
    }

    // ...
}
```

Tabnine:

Respuesta: "En el código proporcionado, la clase chofer no tiene un método que valide el cambio de la propiedad TipoLicencia después de su creación. Esto puede causar inconsistencias en la información del chofer, ya que la edad y el tipo de licencia podrían no coincidir. Para solucionar este problema, se puede agregar un método CambiarLicencia en la clase chofer que valide la edad y el tipo de licencia antes de actualizar la propiedad. Aquí te proporciono una posible implementación:"

```
public class chofer : IPiloto
{
    public string Nombre { get; set; }
    public int Edad { get; set; }
    public string TipoLicencia { get; set; }
    public int EdadMinima { get; set; }

    public chofer(string nombre, int edad, string tipolicencia, string tipoVehiculo) //Constructor
    {
        this.Nombre = nombre;
        this.Edad = edad;
        this.TipoLicencia = tipolicencia;
        if (!ValidarEdad(edad, tipolicencia))
        {
            // Mensaje de error según el tipo de vehiculo
            // ...
        }
    }

    private bool ValidarEdad(int edad, string tipolicencia)
    {
        // Código de validación de edad y tipo de licencia
        // ...
    }

    public void CambiarLicencia(string nuevalicencia)
    {
        if (ValidarEdad(this.Edad, nuevalicencia))
        {
            this.TipoLicencia = nuevalicencia;
            Console.WriteLine("Licencia cambiada a: {0}", nuevalicencia);
        }
        else
        {
            Console.WriteLine("La edad no es válida para el tipo de licencia seleccionado.");
        }
    }

    public void MostrarInformacion()
    {
        Console.WriteLine("Nombre: {0}", this.Nombre);
        Console.WriteLine("Edad: {0}", this.Edad);
        Console.WriteLine("Tipo de licencia: {0}", this.TipoLicencia);
    }
}
```

Resumen: En general, creo que me gusto usar mas Tabnine, principalmente por el hecho que desde el primer Prompt que hice el ya implemento las funciones o métodos necesarios directamente en mi código sin necesidad de hacer otro Prompt pidiendo que lo hiciera, ya que si vemos en Codeium si hace lo que le pedí, pero es como un “ejemplo” o representación y no lo hace como real, en mi código. Trate de cambiarlo, ver si faltaba alguna configuración e hice varios prompts y de igual manera continuo dándome lo mismo.

Los dos resultados son muy semejantes ya que hizo posible que se mostrara un mensaje de error si no coincide la edad según el tipo de licencia, lo que yo entiendo es que prácticamente se hizo tipo una “condición”. A pesar de haber dado dos resultados semejantes, me quedaría con Tabnine ya que podría ver el código, y entender la lógica del por qué funciona así y saber implementarlo mas fácilmente, y no perder tiempo haciendo prueba y error, o yendo a otro LMM y buscar una solución.