Buenas Prácticas .Net

Convención de nombres

https://msdn.microsoft.com/es-es/library/ms184412(v=vs.100).aspx

Espacios de nombres	Clases/Interfaces/Tipos	Métodos/Propiedades	Atributos/Variables	Eventos
Estructura: <Company>. (<Product>| <Technology>) [.<Feature>] [.<Subnamespace>] Por ejemplo: Microsoft. WindowsMobile.DirectX. • Palabras con notación Pascal, separadas por punto y sin espacios. • Nombres plurales, con excepción de marcas.	• Notación Pascal. • Nombres descriptivos. • No usar prefijos en clases. • Finalizar clases derivadas con el nombre de la clase base. • Interfaces con el prefijo de la letra "I". • Nuestros Tipos con prefijo "T". • Utilizar tipos de .net en minúscula.	• Verbos con la primera letra de cada palabra en mayúsculas. • Definir la acción, pero no utilizar detalles de implementación del método en su nombre. • Propiedades: Definición: Camel. Métodos "get": Pascal con misma nomenclatura que la definición.	• Primera letra minúscula, resto de palabras en mayúscula. • No utilizar caracteres que no sean letras. • Elementos interfaz gráfica con prefijo adecuado (txtNombre,…)	• Verbos con la primera letra de cada palabra en mayúsculas • Concepto de anterior y posterior en el verbo. Por ejemplo: Closing y Closed. • No usar conceptos como "After" o "Before". • Controladores de eventos delegados con sufijo EventHandler.

Sangría y espaciado

- Sangría con tabulador, nunca espacios, y por jerarquía.
- Comentarios y llaves al mismo nivel que el código.
- Respecto a los espacios entre líneas
    - Debe considerarse 1 línea en blanco
        - Entre métodos
        - Entre las variables locales y su primera declaración
        - Entre secciones lógicas dentro de un método para mejorar la lectura
    - Usa #region para agrupar piezas de código juntas, con una misma tipología.

Ordenación

Aunque la ordenación de variables y métodos se basa en su visibilidad también se permite la agrupación por funcionalidades siempre respetando la precedencia por visibilidad.

1. Comentarios de clases e interfaces
2. Definición de la propia clase o interfaz
3. Variables static
    1. Public/Protected/Private
4. Varibles instanciables
    1. Public/Protected/Private
5. Contructores
6. Metódos (Agrupados por funcionalidades, no especifica nada acerca del orden de static o no)
    1. Public/Protected/Private/Internal

Y recordad amiguitos!!! No olvidéis mantener un bajo acoplamiento y una alta cohesión!!