Normas de Estilo Para el Proyecto NavajaValirya

Organización de los ficheros de código fuente

- Mantener las clases y los ficheros cortos, no más de 2.000 líneas de código.
- Crear un fichero para cada clase, con el mismo nombre y la extensión .cs

Indentación

- Utilizar **Tab** para indentar el código, **nunca espacios**.
- Cuando una expresión no quepa en una línea de código, dividirla de acuerdo a estos principios:
    - Nueva línea después de coma
    - Nueva línea después de un operador aritmético
    - Evitar, siempre que sea posible, dividir los paréntesis.

Espacios en blanco

Tiene que haber un espacio en blanco:

- Después de coma o punto y coma.
- Alrededor de los operadores

Líneas en blanco

Las líneas en blanco se usan para:

- Separar las declaraciones de variables del resto del código.
- En las clases, separar las declaraciones de campos de los métodos y propiedades.
- Separar los métodos y propiedades entre sí.
- Dentro un método, para separar las diferentes secciones lógicas.
- Usar 2 líneas en lugar de una para hacer más evidente la separación de código.

Comentarios

- Usar comentarios de documentación en lugar de comentarios de bloque para documentar los que hacen las clases, métodos, etc.
- Los comentarios de una línea deben estar en líneas independientes, no al final de líneas de código, excepto en declaraciones de variables y para señalar el final de una estructura larga.

Llaves

- Su colocación se debe hacer en líneas reservadas para cada una de ellas, sin ninguna otra instrucción en la línea. Ambas deben ir en la misma columna que la instrucción que la precede.
- Se debe evitar en la medida de lo posible omitir las llaves, incluso en aquellos bloques que encierran una única sentencia.

Declaraciones

- Declarar las variables agrupadas por tipos en la misma línea a ser posible.
- Usar nombres autoexplicativos, para evitar comentar las variables.
- Inicializar las variables después de la declaración de las mismas dejando una línea en blanco entre la declaración y la inicialización. Si existe sólo una variable de un tipo se puede inicializar en la misma línea en la que se declaró.

Convenios de nombres

Estilos

Se utilizarán los siguientes estilos explicados más detalladamente en la asignación de nombres.

- **Estilo PasCal**

La primera letra de cada palabra debe ser mayúscula. Las iniciales siguientes deben ser todas mayúsculas.

- **Estilo caMel**

La primera letra de cada palabra debe ser minúscula. Las iniciales siguientes deben ser todas mayúsculas.

Asignación de nombres

Un nombre debe describir el significado del elemento, es decir, qué hace o qué representa.
No usar el guion bajo.

- **Nombres de clases:**
    - Estilo caMel.
    - Usar sustantivos preferentemente.
    - Las clases de excepción deben finalizar con Exception.
- **Nombre de interfaces:**
    - Estilo caMel.
    - Usar sustantivos o adjetivos que describan comportamiento.
    - Se puede añadir el prefijo form para indicar el formulario.
- **Nombres de campos:**
    - Estilo PasCal.
    - Se nombrarán con la inicial del campo al que representa.

    Ejemplo: Botón = Label = LTextoLargo, etc…
    - Usar sustantivos o adjetivos.
- **Nombres de constantes:**
    - Estilo en mayúsculas precedido de una k minuscula.
    - Ejemplo: kVALOR.

- **Nombres de variables:**
    - Estilo caMel.
    - Usar i, j, k, l, m, etc para contadores locales de bucles triviales, declarándolos lo más cerca posible del bucle.
- **Nombre de métodos:**
    - Estilo caMel
    - Usar verbos preferentemente.
- **Nombre de propiedades:**
    - Estilo PasCal
    - Usar el mismo valor del campo que encapsula.
- **Nombres de enumeraciones:**
    - Estilo caMel para el nombre de la enumeración como para los valores.
    - Usar nombres en singular para enumeraciones que obligan a elegir un sólo valor y nombres en plural para enumeraciones que permiten escoger valores.
- **Namespaces:**
    - Los espacios de nombres permiten nombrar a clases diferentes con el mismo nombre, pero no pueden estar duplicados, por lo tanto una buena práctica es llamarlos con el nombre del programa o librería seguido del nombre de dominio de la organización, sección o programador.

Buenas prácticas de programación

- Declarar los campos siempre como **privados**, usando las propiedades para acceder a ellos.
- No usar nunca la sentencia **goto** y no abusar de **break**, **continue** o **exit**.
- No exceder de 4 niveles de anidamiento. Si esto sucede, se debe agrupar en funciones.
- Reutilizar bibliotecas y clases existentes.
- Evitar copiar y pegar código.
- Usar el mismo idioma para nombrar los elementos.
- Revisar periódicamente los comentarios, eliminando los desactualizados y los bloques de código comentados que ya no sirvan.
- Usar los comentarios que empiezan por //TODO para marcar tareas a realizar en un futuro.