

PIS 2014 Grupo 10

Estándar de Implementación

Versión 1.3

Historia de revisiones

Fecha	Versión	Descripción	Autor
30/08/2014	1.0	Creación del documento.	Leticia Vaz.
30/08/2014	1.1	Revisión de documento.	Joaquín Gatica.
30/08/2014	1.2	Revisión y Verificación del documento.	Gonzalo Sintas.
31/08/2014	1.3	Revision final de SQA.	Federico Blumetto.

Contenido:

- [1. Convenciones Generales](#)
- [2. Convenciones Específicas](#)
 - [2.1. Variables](#)
 - [2.2. Constantes](#)
 - [2.3. Indentación](#)
- [3. Otros](#)
 - [3.1. Comentarios](#)
 - [3.2. Clases, Estructuras e Interfaces](#)
 - [3.3. Parámetros de tipo Genérico](#)
 - [3.4. Enumerados](#)
 - [3.5. Métodos](#)
 - [3.6. Propiedades](#)
 - [3.7. Eventos](#)
 - [3.8. Campos](#)
 - [3.9. Resumen de notaciones](#)
 - [3.10. Bibliografía](#)

1. Convenciones Generales

- El idioma que se utilizará para desarrollar el código fuente es inglés. Esto implica que los nombres de las variables, funciones, clases, constantes, comentarios y todo lo correspondiente al código deberá ser escrito en dicho idioma.
- Se deben elegir nombres que sean fáciles de leer y comprender en inglés. Es preferible que un nombre sea legible a que sea breve. Por ejemplo, una propiedad llamada *"CanScrollHorizontally"* es mejor que una llamada *"ScrollableX"*.
- No se deben usar guiones bajos, guiones ni ningún carácter no alfanumérico.
- No se debe utilizar notación Húngara, la cual consiste en añadir un prefijo en minúscula que indica el tipo. Por ejemplo, si el tipo es un booleano, dicha notación usaría el prefijo *b*.
- Evitar el uso de nombres de identificadores que están en conflicto con keywords de otros lenguajes de programación ampliamente utilizados.
- No se deben usar abreviaciones o contracciones como nombres de identificadores. Por ejemplo, se debe usar *"GetWindow"* en vez de usar *"GetWin"*.
- Evitar el uso de acrónimos que no son ampliamente reconocibles, e incluso si lo son, usarlos solo en caso de ser necesario.
- Usar nombres semánticamente interesantes en lugar de keywords de un lenguaje específico. Por ejemplo, *"GetLength"* es mejor nombre que *"GetInt"*.
- Usar un tipo de nombre genérico del CRL en vez de usar un nombre específico de un lenguaje, en el extraño caso en que un identificador no tiene un significado semántico más allá de su tipo.
- Se debe utilizar notación Pascal para nombrar todos los miembros públicos (métodos, propiedades, eventos, campos), tipos y namespace que estén formados por varias palabras. Dicha notación consiste en que las primeras letras de un identificador y todas las primeras letras de las palabras subsiguientes, que concatenadas forman el identificador, se escriben en mayúscula. Por ejemplo: *"BlackColor"*.
- Se debe utilizar notación Camel para nombrar parámetros. Dicha notación consiste en que la primera letra de la variable se escribe en minúscula y todas las primeras letras de las subsiguientes palabras, que concatenadas forman el nombre de la variable, se escriben en mayúscula. Por ejemplo, *"blackColor"*.

2. Convenciones Específicas

2.1. Variables

- Utilizar notación Camel para nombrar las variables.
- Usar nombres que sean descriptores de las variables.
- Considerar el uso de nombres basados en el significado de la variable en lugar del tipo de la variable.

2.2. Constantes

- Utilizar notación Camel para nombrar las constantes.
- Usar nombres que sean descriptores de las constantes.
- Considerar el uso de nombres basados en el significado de la constante en lugar del tipo de la constante.

2.3. Indentación

- Escribir únicamente una instrucción por línea.
- Escribir únicamente una declaración por línea.
- La sangría debe ser de un tabulador (4 espacios).
- Utilizar los paréntesis para que las expresiones sean más fáciles de comprender. Por ejemplo:

```
if ((val1 > val2) && (val1 > val3))  
{  
    // Take appropriate action.  
}
```

3. Otros

3.1. Comentarios

- El comentario debe estar en una línea independiente, no al final del código.
- El texto debe empezar con una letra mayúscula y finalizar con un punto.
- Entre el delimitador del comentario (//) y la primera letra del texto debe insertarse un espacio.
- No se crean bloques de asterisco alrededor del comentario.

3.2. Clases, Estructuras e Interfaces

- Las clases y estructuras se deben nombrar utilizando sustantivos o frases sustantivas, utilizando la notación Pascal.
- Nombrar interfaces con adjetivos u ocasionalmente con sustantivos o frases sustantivas.
- Finalizar el nombre de una clase derivada con el nombre de la clase base.
- No utilizar prefijos para nombrar las clases
- Utilizar como prefijo la letra I para nombrar las interfaces, de esta forma, se sabe que el tipo es una interfaz.
- Los nombres de las interfaces y los de las clases que son implementaciones estándar de dichas clases debe diferir solo en el prefijo I.

3.3. Parámetros de tipo Genérico

- Los parámetros de tipo genérico se deben nombrar con nombres descriptivos, a menos que un nombre de una sola letra sea auto-explicativo y un nombre completo no agregue valor.
- Utilizar como prefijo la letra T para nombrar los tipos. Por ejemplo,

```
public interface ISessionChannel<TSession> where TSession : ISession{  
    TSession Session { get; }  
}
```

- Indicar las restricciones de un tipo genérico en el nombre del tipo. Por ejemplo, un tipo restringido a *ISession*, debe ser nombrado *TSession*.

3.4. Enumerados

- Se deben usar nombres en singular para los enumerados, a menos que su valor sea campos de bits.
- Se deben utilizar nombres en plural para los enumerados cuyo valor son campos de bits, también llamados flag enum.
- No se deben usar los sufijos *"Enum"*, *"Flag"* ni *"Flags"* en los nombres de los enumerados.
- No se deben utilizar prefijos en los nombres de atributos valuados. Por ejemplo, *"rft"* para reach text enums.

3.5. Métodos

- Se deben utilizar nombres que sean verbos o frases verbales, ya que el significado de los métodos es tomar acción. Por ejemplo,

```
public class String {  
    public int CompareTo(...);  
    public string[] Split(...);  
    public string Trim();  
}
```

3.6 Propiedades

- Se deben utilizar sustantivos, frases de sustantivos o adjetivos en notación Pascal para nombrar las propiedades.
- No se deben utilizar propiedades que coincidan con un método Get, ya que típicamente significa que la propiedad debería ser en realidad un método.
- Las colecciones de propiedades deben nombrarse con frases en plural describiendo los elementos de la colección, en lugar de utilizar una frase en singular seguida de *"List"* o *"Collection"*.
- Las propiedades booleanas se deben nombrar utilizando frases afirmativas, por ejemplo, se debe usar *"CanSeek"* en lugar de *"CantSeek"*.
Opcionalmente, se pueden usar los prefijos *"Is"*, *"Can"* o *"Has"*.
- En el caso que una propiedad tenga un tipo privado asociado, se debe nombrar igual que su tipo. Por ejemplo,

```
public enum Color {...}  
public class Control {  
    public Color Color { get {...} set {...} }  
}
```

3.7. Eventos

- Se deben utilizar verbos o frases verbales para nombrar los eventos. Por ejemplo, "Clicked", "Painting", "DroppedDown"
- Utilizar nombres que hagan referencia a antes o después utilizando los tiempos presente y pasado. Por ejemplo, un evento de cierre que es lanzado antes de que una ventana sea cerrada debería llamarse "Closing" y uno que es lanzado después de que la ventana haya sido cerrada debería llamarse "Closed".
- No se deben utilizar "Before" o "After" como sufijos o prefijos para indicar pre-eventos o post-eventos.
- Los controladores de eventos deben ser nombrados con el sufijo "Event Handler". Por ejemplo,

```
public delegate void ClickedEventHandler(object sender, ClickedEventArgs e);
```

- En los controladores de eventos se deben utilizar los parámetros "Sender" y "e", donde el parámetro "Sender" representa el objeto que es lanzado por el evento y típicamente es de tipo "Object".
- Las clases que son argumento de eventos deben ser nombradas con el sufijo "EventArgs".

3.8. Campos

- Se deben utilizar sustantivos, frases sustantivas o adjetivos en notación Pascal para nombrar los campos.
- No se deben utilizar prefijos para nombrar los campos. Por ejemplo, no se deben utilizar los prefijos "_g" o "_s" para indicar campos estáticos.

3.9. Resumen de notaciones

Identificador	Notación	Ejemplo
Namespace	Pascal	<code>namespace System.Security { ... }</code>
Type	Pascal	<code>public class StreamReader { ... }</code>
Interface	Pascal	<code>public interface IEnumerable { ... }</code>
Method	Pascal	<code>public class Object { public virtual string ToString(); }</code>
Property	Pascal	<code>public class String { public int Length { get; } }</code>
Event	Pascal	<code>public class Process { public event EventHandler Exited; }</code>
Field	Pascal	<code>public class MessageQueue { public static readonly TimeSpan InfiniteTimeout; } public struct UInt32 { public const Min = 0; }</code>
Enum Value	Pascal	<code>public enum FileMode { Append, ... }</code>
Parameter	Camel	<code>public class Convert { public static int ToInt32(string value); }</code>

3.10. Bibliografía

- <http://msdn.microsoft.com/es-es/library/ff926074.aspx>
- [http://msdn.microsoft.com/en-us/library/ms229002\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/ms229002(v=vs.110).aspx)
- http://es.wikipedia.org/wiki/Notaci%C3%B3n_h%C3%BAngara