


## ° Normativas de Código

Principios de Desarrollo de Software – Tema IV

Conjunto de reglas y recomendaciones  
 sobre como se de escribir, estructurar y definir el código.  
 Para que en un futuro se puedan escribir sobre él y saber que hace y como.

**SEL** @UC3MPSDLEG  
 @UC3MPSDCOL  
 #S02CIPDS

1



## Motivación de Normativas de Código

- En principio, no podrías pedirle a un equipo que codificase con un estándar común. Los programadores son muy individualistas (“yo lo hago así y funciona”, “mi manera es mas eficiente”,...).
- Si los programadores van a estar tocando partes distintas del sistema, intercambiando compañeros, haciendo recodificaciones, debemos de establecer un estándar de codificación aceptado e implantado por todo el equipo.

**SEL** @UC3MPSDLEG  
 @UC3MPSDCOL  
 #S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

2

2

## Motivación de Normativas de Código

- Con el tiempo el código se va deteriorando con los cambios.
- Se hace difícil de leer y modificar.
- Se gasta mucho esfuerzo en reescribir el código de otro.



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

3

3

## Propósito de las Normativas de Codificación

- Se pretende que este estilo ayude a construir programas correctos, entendibles y fáciles de mantener
- Contiene Reglas y Recomendaciones para escribir código.
- En este documento, una "norma" es una regla que debe cumplirse obligatoriamente;
- una "recomendación" es una regla válida en general, pero que puede admitir o requerir excepciones.



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

4

4

## Aspectos Básicos *(lo mínimo)*

- Nombres y Codificación de Ficheros.
- Organización de Ficheros
- Comentarios
- Secuencias
- Reglas para asignar nombres



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

5

5

## Aspectos Avanzados

- Gestión de errores y excepciones
- Seguridad
- Patrones de diseño
- Rendimiento
- Globalización
- Fiabilidad de mantenimiento
- Otras buenas prácticas



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

6

6

## Organización de Archivos

- **Archivos de Código Fuente**

- **Norma:** Todos los archivos tendrán leyenda de derechos de autor.

*/\* Copyright Mes, año, Autor, Compañía \*/*

- **Norma:** Todos los ficheros fuente, al igual que el resto de ficheros de interés (ejecutables, dll, etc), se gestionarán con SourceSafe.



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

7

7

## Namespaces, clases y variables (I)

- **Namespace Raíz**

- **Norma:** Todos los NAMESPACES partirán del NAMESPACE raíz Tragsatec. En la cabecera de los ficheros se incluirá la instrucción:

```
using Tragsatec;
namespace Tragsatec.Spatial
{
    class Index
    {
        ...
    }
}
```



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

8

8

## Namespaces, clases y variables (II)

- Estilos de Capitalización
  - Estilo “Pascal” – Palabras compuestas con las iniciales en mayúsculas. Por ejemplo:
    - EstoEsUnEjemplo
  - Estilo “Camel” - Palabras compuestas con las iniciales en mayúsculas, excepto la primera. Por ejemplo:
    - estoEsUnEjemplo
  - Mayúsculas – Todas las letras en mayúsculas. Usar sólo en constantes. Por ejemplo:
    - public const PI = 3.1416;
- Norma General:
  - PARA CAMPOS PROTEGIDOS Ó PRIVADOS ->ESTILO CAMEL
  - PARA TODO LO DEMÁS -> ESTILO PASCAL



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

9

9

## Métodos

- Declaración de los métodos
  - **Recomendación:** Nombres de funciones en Inglés
  - **Norma:** Cuando los argumentos de un método no quepan en la misma línea se hará la separación después de una coma y la siguiente línea comenzará indentada al nivel del primer argumento de la función para una mejor visualización
  - **Norma:** Incluir una descripción antes de la declaración de un método público mediante /// ( tres barras). Definir siempre el propósito, los argumentos, sus restricciones y el valor devuelto.
  - **Norma:** Incluir una descripción antes de la declaración de una función privada mediante // (dos barras), para que no se incluya en la documentación generada por Visual Studio



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02CIPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

10

10

## Tratamiento de los errores: Excepciones (I)

- **Norma:** Nunca se mostrarán cuadros de mensaje desde una Clase de un componente reutilizable.
- **Norma:** Si se detecta un error en un argumento de entrada se generará una excepción del tipo `ArgumentException`
- **Norma:** Cuando un método ejecute código que tenga posibilidades altas de tener una excepción se deberá de atrapar el error, añadir información auxiliar, para ayudar al usuario a localizar el problema, y volver a lanzar la excepción
- **Norma:** Las excepciones únicamente deberán ser mostradas por la aplicación principal, nunca por los componentes



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02C.IPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

11

11

## Tratamiento de los errores: Excepciones (II)

```
try
{
    cnx.Open();
}
catch(System.Exception ex) // No dejar vacío!
{
    MessageBox.Show(ex.ToString());
}
finally
{
    if (cnx != null)
        cnx.Close();
}
```



@UC3MPSDLEG  
@UC3MPSDCOL  
#S02C.IPDS

Principios de Desarrollo Software - Tema 4. Normativas de Código

12

12