Web\_venta

Documento de Estandarización de Código

Versión 1.0

HISTORIAL DE REVISIONES

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 10/12/2020 | 1.0 | Creación de documento | AAGD,CER |
|  |  |  |  |

**ESTANDARIZACION DE CODIFICACION**

1. **Creación de Librería de Clases**

Empezarán con el nombre proyecto seguido del punto (.) y un nombre que denote a la libería de clases.

*Ejemplo:*

Web\_venta.InterfacesUsuario

Web\_venta.InterfaceAdministrador

1. **Creación de Clases**

Los nombres de las clases deben empezar con mayúsculas.

Para nombres compuestos la primera letra de cada palabra empieza con mayúsculas.

*Ejemplo:*

Persona.cs

TipoArticulo.cs

1. **Organización de la Clase**
   * **Encabezado de la Clase**

*Ejemplo:*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** Clase: TipoArticulo.cs \*

*\** Espacio de nombres: WEB\_venta.CapaNegocio \*

*\** Autor: Angel Gonzalez \*

\*------------------------------------------------------------------------------------------------------------\*

\* Descripción: \*

\* \*

\* \*

\*------------------------------------------------------------------------------------------------------------\*

\* Fecha Autor Razón \*

\* 10/12/2020 Angel Gonzalez Creación \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

* + **Sentencias using**

*Ejemplo:*

using System;

* + **Nombre de Espacio o Namespace**

*Ejemplo:*

namespace WEB\_VENTA.CapaNegocio

* + **Denotación de la Clase**

*Ejemplo:*

Class TipoArticulo

1. **Especificaciones de las Clases**
   * **Constantes**

Las constantes deber ser escritas completamente en mayúsculas

*Ejemplo:*

Const Hora = 7

* + **Variables**

En general, los nombres de las variables deben ser significativos, descriptivos y escritos en minúsculas de tal forma que cualquier persona pueda saber su función y valoración. En variables con nombres compuestos cada palabra se escribe la primera letra con mayúscula.

Es recomendable una declaración por línea a excepción de las variables temporales que pueden ser denotadas en la misma línea.

*Ejemplo:*

string cedula;

string Nombre;

string correo\_electronico;

string procedimiento\_Almacenado;

Las variables temporales se las denota con caracteres.

*Ejemplo:*

int i, j,@, k;

Las variables de instancia deben ser declaradas con el mismo nombre de la clase en minúsculas.

*Ejemplo:*

DataSet dataset;

Las variables estáticas deben ser declaradas con un subguión adelante

*Ejemplo:*

static string \_nombreBaseDatos;

* + **Métodos**

En general, los métodos deben ser verbos significativos y descriptivos de acuerdo a la clase que se este implementando, de tal forma que cualquier persona pueda saber su función, debe ser escrito en mayúscula la primera letra y lo demás en minúsculas En métodos con nombres compuestos cada palabra se escribe la primera letra con mayúscula.

*Ejemplo:*

void Eliminar(int id) {…}

OperadorDatos ObtenerOperadorDatos() {… }

*Ejemplo:*

void Guardar()

{

try

{

string nombre;

int edad;

……..

}

catch( Exception exc)

{

throw exc;

}

}

1. **Sentencias**
   * **Sentencias compuestas**

Las sentencias compuestas son sentencias que contienen listas de sentencias encerradas entre llaves. La llave de apertura se debe poner al inicio de una nueva línea, la llave de cierre debe empezar una nueva línea y ser ubicada al mismo nivel que el principio de la sentencia.

*Ejemplo:*

try

{

for(..;..;..)

{

…….

}

}

catch( Exception exc)

{

throw exc;

}

* + **Sentencias de retorno**

Una sentencia return con un valor no debe usar paréntesis.

*Ejemplo:*

return id;

return Nombre;

1. **Comentarios**

Cada método deber tener su respectivo comentario de linea, bloque o comentario para el parametros.

* + **Comentarios de Línea o parametro**

*Ejemplo:*

// Este es un comentario de una sola línea

* + **Comentarios de Bloque**

*Ejemplo:*

/\*

Este es un comentario

de varias líneas

\*/

1. **Regiones**

Permite especificar un bloque de código que se puede expandir o contraer cuando se utiliza la característica de esquematización del editor de código de Visual Studio.

Sólo se crearán estos tipos de regiones:

* + Region a Variables Privadas
  + Region a Propiedades Publicas
  + Region a Metodos de Persistencia
  + Region a Metodos de Clase

BUSQUE MAS EJEMPLO O ELIMINE ESTE

*Ejemplo:*

