

Universidad Tecnológica de la Riviera Maya

Aguilar Alvarez Arturo

Calidad desarrollo de software

TSU en Tecnologías de la Información y la Comunicación

Asesor de la materia: Ing. Cristian Bernal

Lunes30 de enero del 2017

Tabla de contenido

[Definición de línea de código 2](#_Toc474878032)

[Comentarios 2](#_Toc474878033)

[Uso de las llaves 3](#_Toc474878034)

[Convención de nombre 3](#_Toc474878035)

[Nombre de archivo 4](#_Toc474878036)

[Nombre de clases 4](#_Toc474878037)

[Nombre de funciones 4](#_Toc474878038)

[Nombre de variables 4](#_Toc474878039)

[Nombre de constantes 4](#_Toc474878040)

[Nombre de tablas 4](#_Toc474878041)

[Estructura de control 4](#_Toc474878042)

[If 4](#_Toc474878043)

[Switch 4](#_Toc474878044)

[Estructuras loops 5](#_Toc474878045)

[Do While 5](#_Toc474878046)

[For 5](#_Toc474878047)

[Foreach 5](#_Toc474878048)

[Clases y metodos 5](#_Toc474878049)

[Variables/propiedades 5](#_Toc474878050)

[Constante 5](#_Toc474878051)

[Código 6](#_Toc474878052)

[Base de datos 6](#_Toc474878053)

# Definición de línea de código

GUÍA DE ESTILOS DE C# Óscar Álvarez Guerras

## Comentarios

Los comentarios facilitan la comprensión del código para una lectura rápida.

Los comentarios se pueden escribir en diferentes estilos dependiendo de su longitud y su propósito.

Reglas de generales:

* Los comentarios en general se escriben en líneas que no contienen código y antes del código que queremos clarificar. Esta regla se aplica siempre si el comentario tiene más de una línea.
* Sólo en dos casos se permite poner en la misma línea un comentario y una instrucción: comentarios a una definición de variable, que explica la finalidad de esta variable, y un comentario para indicar final de una estructura del lenguaje.

Ejemplo:

/// Descripción genérica de la clase

/// Descripción detallada de la clase

/// Descripción breve de la variable private int NumPruebas;

// Contador pruebas realizadas

### Uso de las llaves

La forma que tiene C# de agrupar instrucciones en bloques es utilizar las 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.

Ejemplo:

for (i = 0; i < 10; i++)

{

…

}

## Convención de nombre

Los identificadores que dan nombre a la aplicación y a los archivos nos proporcionan una fácil documentación, es por eso que se necesita definir una serie de reglas que nos ayuden a elegir los nombres correctos de las clases métodos, variables.

* Estilo PasCal Este convenio determina que cada letra de cada palabra debe ser mayúscula, excepto la primera palabra. Ejemplo: clienteServidor.
* Elegir nombres comprensibles y en relación con la tarea que corresponda al objeto nombrado.
* Utilizar palabras completas. Las abreviaturas pueden resultar confusas.
* Cuando se trabaje con siglas o acrónimos, éstos deben mantener todos sus caracteres es mayúsculas.
* Variables, clases, y métodos deben están en singular dentro de la aplicación
* Los nombres de las tablas dentro de la base de datos deben ser plural
* Las columnas de las tablas de la base de datos deben están en singular

## Nombre de archivo

Los nombres de los archivos iniciaran con la letra minúscula Estilo PasCal.

Los nombres estarán en singular.

## Nombre de clases

Las clases estarán con el formato Estilo PasCal y estarán en plural.

## Nombre de funciones

Las funciones

## Nombre de variables

Las variables estarán en singular con el formato Estilo PasCal.

## Nombre de constantes

Las variables estáticas estarán definidas todo en mayúsculas.

## Nombre de tablas

Las tablas de la base de datos estarán en plural pero los campos de la tabla estarán en singular.

## Estructura de control

### If

if (condition)

{

Console.WriteLine("The variable is set to true.");

}

else

{

Console.WriteLine("The variable is set to false.");

}

### Switch

switch (caseSwitch)

{

case 1:

Console.WriteLine("Case 1");

break;

case 2:

Console.WriteLine("Case 2");

break;

default:

Console.WriteLine("Default case");

break;

}

# Estructuras loops

### Do While

int x = 0;

do

{

Console.WriteLine(x);

x++;

} while (x < 5);

### For

for (int i = 1; i <= 5; i++)

{

Console.WriteLine(i);

}

### Foreach

int[] fibarray = new int[] { 0, 1, 1, 2, 3, 5, 8, 13 };

foreach (int element in fibarray)

{

System.Console.WriteLine(element);

}

### Clases y metodos

namespace ProgrammingGuide

{

// Class definicion.

public class MyCustomClass

{

public int Number { get; set;

}

// metodo.

public int Multiply(int num)

{

return num \* Number;

}

// instancia constructor.

public MyCustomClass()

{

Number = 0;

}

}

### Variables/propiedades

Ejemplo

### Constante

Ejemplo de constante

# Código

El código se encuentra en el siguiente repositorio:

<https://github.com/ArturoAguilarAlvarez/GestionClientes.git>

## Base de datos

create database GestionCientes;

use GestionCientes

create table categorias(

idCategoria int unique IDENTITY(1,1),

nombre varchar(30),

primary key(idCategoria)

);

create table clientes(

idCliente int unique IDENTITY(1,1),

nombre varchar(20),

apellidos varchar(30),

telefono varchar(15),

correo varchar(30),

idCategoria int,

primary key(idCliente),

foreign key(idCategoria)references categorias(idCategoria));