



Estándares de Programación

Proyecto ERP

Seminario AD
ERP

Estándares de Programación

Este documento describe reglas y recomendaciones para el desarrollo de aplicaciones en la plataforma Visual Studio .NET, y su objetivo es definir el formato y estilo que será implementando para el desarrollo de proyecto ERP de seminario de análisis de sistemas.

VARIABLES

DEFINICION DE FORMATO DE VARIABLES:

Existen distintos sistemas de notación de nombres de variables de las cuales se pueden desatacar las siguientes, para la estética en la codificación.

1. Notación Húngara

Se estableció la notación HUNGARA que consta de identificar en mayúsculas el tipo de variable seguido del nombre de la viable en letras minúsculas.

Agregándole a esta notación un prefijo que indique el nombre del módulo de donde proviene el origen de la variable, Ejemplo.

PARA LOS CONTROLES (textbox, label, Grid etc) se utilizara notación húngara. El motivo de la utilización de notación húngara para los controles y uso exclusivo de los controles, será para no confundir nombres de variables nuestras que declaremos en el programa con los controles que Visual Studio proporciona. Los prefijos a utilizar son:

Controles	Ejemplo
Button	BtnNombre
CheckBox	ChbxNombre
CheckedListBox	ChlbxNombre
ComboBox	CbxNombre
DateTimePicker	DtpNombre
Label	LblNombre
LinkLabel	LnkNombre
ListBox	LbxNombre
ListView	LstvNombre
MaskedTextBox	MtxtNombre
MontCalendar	MclrNombre
NotifyIcon	NicNombre
NumericUpDown	NudNombre
PictureBox	PicNombre
ProgressBar	PrbNombre

RadioButton	RbtnNombre
RichTextBox	RtxtNombre
TextBox	TxtNombre
ToolTip	TipNombre
TreeView	TvwNombre
WebBrowser	WbrsNombre

Menús y Punteros

Punteros
ContextMenuStrip
MenuStrip
StatusStrip
ToolStrip
ToolStripContainer

Ejemplo

PtrNombre
CmsNombre
MspNombre
SspNombre
TspNombre
TscNombre

Datos y Otros

DataSet
DataGridView
BindingSource
BindingNavigator
DataGrid
DataRow
DataTable
Decimal
GridItem
Icon
Image
ImageList
LinkArea
Object
Panel
Timer
FlowLayoutPanel
GroupBox
SplitContainer
TabContrl
TableLayoutPanel
Otros
Clases
Form

Ejemplo

DsNombre
DgvNombre
BgsNombre
BgnNombre
DgNombre
DrNombre
DtNombre
DcNombre
GiNombre
IcNombre
ImgNombre
ImglNombre
LnkaNombre
ObjNombre
PnlNombre
TmrNombre
FlpNombre
GbxNombre
SpcNombre
TctrlNombre
TlpNombre
Ejemplos
ClsNombre
FrmNombre

Si se utilizan controles no establecidos en esta sección, pueden crear el prefijo respetando las reglas de las RECOMENDACIONES GENERALES que se encuentra más adelante.

PARA LAS DEMAS VARIABLES (que no sean controles) se utilizara la notación Pascal Case, donde la primera letra será mayúscula como se muestra en el ejemplo.

Ejemplo: *NombreVendedor Correcto*
 Nombre_Vendedor Incorrecto
 Nombre-Vendedor Incorrecto

“Una de las recomendaciones más importantes de la programación con estilo, es usar nombres cortos en identificadores locales, y nombres largos y descriptivos en identificadores globales”

RECOMENDACIONES GENERALES:

1. Evitar nombres totalmente en MAYUSCULAS o en minúsculas.
2. Siempre escoger nombres específicos y con significado en sí mismos.
3. Variables y propiedades deben describir la entidad que representan no el tipo o tamaño.
4. Evitar el uso de abreviaturas a menos que el nombre completo sea excesivo
- 5. Evitar a abreviaturas que su longitud sea mayor a 5 caracteres.**
6. Nunca incluir el nombre de la clase a los nombre de las propiedades.

Ejemplo:

Cliente.Nombre **NO** Cliente.NombreCliente.

ESTILO DE CODIFICACION

Las siguientes secciones describen el camino preferido para implementar código fuente, para así crear un código legible, claro y consistente que sea fácil de comprender y mantener.

1. Nunca declarar más de un espacio de nombres por archivo.
2. Evitar colocar múltiples clases en un mismo archivo.
3. Siempre usar Tab e indentación de tamaño 4.
4. Declarar cada variable independientemente – nunca en la misma sentencia. Colocar la sentencia "using" o "Imports" en la parte superior del archivo. El grupo de los espacios de nombres de .NET colocarle por encima de los espacios de nombres particulares y todos en ordenados alfabéticamente. Ejemplo:

```
[VB.NET]
Imports System
Imports System.Collections
Imports System.Data
Imports System.Data.SqlClient
Imports System.Xml
```

5. Agrupar la implementación interna de las clases por el tipo de miembros en el siguiente orden:
 - a. Campos
 - b. Propiedades
 - c. Constructores y destructores
 - d. Métodos
 - e. Enumeradores, Estructuras o Clases internas
6. Ordenar las declaraciones a partir de los modificadores de acceso y visibilidad:
 - a. Privados
 - b. Internos
 - c. Protegidos
 - d. Público

“La utilización de TAB o indentación de tamaño 4, será un factor importante para la legibilidad del código fuente, a continuación un ejemplo con indentación y sin indentación mostrado en C:”

COMENTARIOS

1. Utilizar comentarios en el interior de los métodos para explicar aspectos globales del algoritmo
2. No utilizar comentarios para explicar código obvio

DOCUMENTACION EN CODIGO

Al inicio de cada hoja, forma o espacio de programación dentro del proyecto se debe documentar obligatoriamente el segmento u hoja de programación para saber a qué se refiere el segmento de programación que es lo que hace dentro de la funcionalidad del software

Ejemplo:

Modulo: Sistema relación con los clientes

Encargado: Rubén López

Fecha: 01 de Agosto del 2015

Especificación: Especificación documentada.

Modificaciones: Actualización Trigger

Otros & Observaciones: Observacion de cambios afectar

DIAGRAMAS RELACIONALES

La distribución visual de las tablas (relaciones entre ellas) debe de estar ordenadas correctamente de acuerdo a las relaciones entre tablas.

No colocar las tablas demasiado alejadas con las que se relacionan para una mejor visualización general.

El formato del nombre de la tabla será de la siguiente forma:
tb_nombremodulo_nombretabla

Ej: tb_icv_huesped, tb_bnc_menu tb_pln_nombre etc.

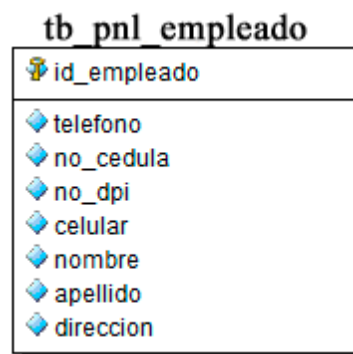
Los nombres de los atributos y tablas serán completamente en minúsculas para evitar problemas futuros.

Ej: nombre, apellido, celular, dirección, cedula, dpi etc.

El nombre del id de cada tabla será compuesto de la siguiente forma: **id_nombre**. Si el nombre del atributo está compuesta por 2 palabras o más serán separadas por "_". Pero se recomienda que se trabaje con una sola palabra que, a menos que se diera el caso.

Ej: no_cedula, codigo_postal etc.

En la siguiente figura se encuentra un ejemplo que implica a los incisos anteriores.



Este es un ejemplo, porque en lugar de "no_cedula" y "no_dpi" podría ser "cedula" y "dpi", dependiendo de su criterio.

El nombre de los atributos debe describir la funcionalidad que represente el atributo. No colocar nombres ilógicos y/o ambiguos.

Se deberá mantener un borrador en papel para una mejor visualización del modulo completo y llevarlo cuando se trabaje con ellas por ejemplo: dudas, integración, etc.

IDENTIFICACION DE MODULOS

Los distintos módulos dentro del sistema final serán identificados por medio de distintos colores. Estos estarán presentes en la imagen de fondo de cada una de las pantallas y la barra de menú de cada uno de los módulos. Por ejemplo, para Contabilidad el fondo de la imagen será azul así como la barra de menú. A continuación se detalla una tabla en donde se indican los colores de los distintos módulos:

ID	Modulo	Color	Color de la barra de menú
MRP	Manufacturing Resource Plannig	Rojo	Snow
CRM	CustomerRelationship Management	Celeste	LightSteelBlue
HRM	Human Resource Management	Verde	Beige
SCM	Supply Chain Management	Morado	Lavender
FRM	Finance Resource Management	Naranja	SeaShell
ADM	Administrative	Azul	Blue

Ver carpeta Zip "Backgrounds".

ICONOS

Los iconos que se utilizaran se encuentran adjuntos en la carpeta Zip "Iconos".