# Historial de modificaciones

|  |  |  |
| --- | --- | --- |
| Creado por:  Jaime Pérez | Versión:  1.0 | **20 /ago / 2013** |
| Modificado por: Jaime Pérez | 1.0 | **22/ ago /2013** |
| Revisado por: Enrique Magnani | 1.0 | **22/ ago /2013** |
| Corregido por: Enrique Magnani | 2.0 | **25/ ago /2013** |
| Autorizado por:  ---- | 2.0 | **29/ ago /2013** |

# Convenciones o estándares de Programación

***Consideraciones***

Se necesita poder distinguir sobre el tipo de objeto se trabaja para darle el tratamiento adecuado, por lo que se plantea contar con prefijos que indiquen el uso de los objetos. Con el fin de generalizar el empleo de nombres y, principalmente, de variables se hace necesario que todo el desarrollo de la aplicación sea llevado a cabo bajo los siguientes lineamientos:

## Código de Aplicaciones

Se utilizará un prefijo de dos caracteres, seguido de una Letra mayúscula. Se describen los prefijos:

Formularios (forms): fr[Nombre]

Tablas: tb[Nombre]

Consultas (querys): qr[Nombre]

Objeto: ob[Nombre]

Multi-Objeto: mo[Nombre]

Base Datos: bd[Nombre]

Webforms: wf[Nombre]

### Procedimientos

Utilizar dos caracteres seguidos del nombre del procedimiento.

pr[Nombre]

### Constantes

Los nombres de constantes siempre se escribirán en MAYÚSCULA y deben iniciar con un prefijo que encapsule el módulo o área del programa, por ejemplo FILEMASTER – FM-. De existir más de una palabra se dividirá con guion mayor (underscore o guion bajo).

**Ejemplo:**

Si vamos a necesitar una constante que va a contener un primer nombre (dos palabras) de una persona, debe ser como el siguiente ejemplo:

PRIMER\_NOMBRE

FM\_AGREGAR\_REGISTRO

FM\_FICHA\_CAMPO\_NUMERICO

FM\_ESTADO\_ACTIVO = ‘A’

### Variables

Los nombres de variables deben tener la siguiente sintaxis:

**<alcance><prefijo de tipo de dato><Nombre de la variable>**

Siendo los prefijos:

**Alcance:**

* Global (g)
* Local (l) , todas las variables locales pueden omitir el uso de este prefijo

Prefijo de tipo de Dato:

|  |  |
| --- | --- |
| Nombre | Prefijo |
| String | st, tx |
| Char, varchar | ch |
| Boolean | bo |
| Numerico | nu |
| Integer | in |
| Long | lo |
| List | ls |
| Array | ar |
| Objeto | ob |
| DateTime | dt |
| Date | da |
| Time | ti |
| Multiclase | mc |
| Collection | cl |

El nombre de la variable es definido por el programador, debe ser lo más claro posible, de fácil interpretación del contenido del mismo. Si el nombre de la variable consta de varias palabras, debemos de separar las palabras con un guion mayor.

**Por ejemplo:**

Si vamos a necesitar una variable que va a contener el nombre (dos palabras) de una persona, debe ser como el siguiente ejemplo:

lsNombre\_empleado

Si necesitamos crear una variable que va a contener un estado (una palabra), debe ser como el siguiente ejemplo:

inEstado o stEestado (txEstado también es válido).

**Posición dentro del Código Fuente**

Todas las variables SIN DISTINCIÓN deben quedar declaradas en al cabecera o header del procedimiento, función o evento. A pesar que el código fuente permita otra acción se requiere que todas queden declaradas en la posición indicada.

Ejemplo correcto

Function XYZ

int inLargo = 1

int inAncho = 500

int inAlto = 400

Un mal ejemplo

Function XYZ

int inLargo = 1

int inAncho = 500

If llamarDatos() then

…

end if

int inAlto = 400

### Controles Visuales

Los controles visuales también deben poseer un nombre estándar, el cual se identificará con un prefijo todo en minúscula, seguido por el nombre del control con la primer letra mayúscula. **Ejemplo:** lb\_Informacion

|  |  |
| --- | --- |
| Nombre | Prefijo |
| Label | lb |
| TextBox | tb |
| ComboBox | cb |
| ListBox | lt |
| DialogBox | db |
| Option | op |
| Checkbox | cb |
| Form | fr |
| Module | vm |
| Class | vc |
| Command Button | bt |
| DataGrid | dg |
| GroupdBox | gb |
| TabPage | tp |
| MenuStrip | ms |
| ToolStripMenu | ts |
| StatusStrip | ss |
| PictureBox | pb |
| TabControl | tc |
| Timer | tm |
| ToolStripContainer | tc |
| TabControl | ta |

### Código fuente comentado

Se entiende como código fuente comentado aquellas líneas de código que fueron puestas entre comentarios por fines de reemplazo de código. Este código comentado no tiene funcionalidad alguna dentro del sistema por lo tanto debe ser eliminado por completo del mismo, solo en casos excepcionales

Por ejemplo

//cargarSegundoCampo()

// Me.cmdCondicionVariable.Enabled = False

//cmdSegCampos.Enabled = False

Las anteriores líneas comentadas no deben pertenecer al código fuente estable que se usará para realizar una compilación.

Esta virtud de programación puede ser altamente evaluada durante el proceso de auditoría.