**DOCUMENTO DE ESTÁNDARES DE PROGRAMACIÓN**

**Proyecto :sistema de gestión de la configuración de software**

**Versión 1.0**

**Historia de Revisión**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Historial de revisiones** | | | | |
| **Ítem** | **Fecha** | **Versión** | **Descripción** | **Equipo** |
| 1 | 12/09/2019 | 1.0 | Versión Final. | Lista desarrolladores:  Angel Gonzales Cave  Luis Moreno Mulluni  Yonathan Mamani Calisaya |

**Estándares de Programación**

1. **OBJETIVO**

Reglamentar la forma en que se implementará el código fuente del proyecto, pasando, por las variables, controles, ficheros, archivos y todo aquello que esté implicado en el código,

Mejorar y uniformizar a través de las reglas que se proponen, el estilo de programación que tiene cada programador.

• Los nombres de variables serán nemotécnicos con lo que se podrá saber el tipo de dato de cada variable con sólo ver el nombre de la variable.

• Los nombres de variables serán sugestivos, de tal forma que se podrá saber el uso y finalidad de dicha variable o función fácilmente con solo ver el nombre de la variable.

• La decisión de poner un nombre a una variable o función será mecánica y automática, puesto que seguirá las reglas definidas por nuestro estándar.

• Permite el uso de herramientas automáticas de verificación de nomenclaturas.

Por tanto, se seguirán dichos patrones para un entendimiento legible del código y para facilitar el mantenimiento del mismo.

1. **DECLARACIÓN DE VARIABLES**

Se propone que la declaración de las variables, se ajusten al motivo para la que se requieran. El nemotécnico definido se establece tomando en consideración principalmente lo siguiente:

* La longitud debe ser lo más recomendable posible. No debe ser tan grande de tal forma que el programador tenga la facilidad de manejo sobre la variable y ni tan corta que no pueda describirse claramente. Para el caso establecemos una longitud máxima de variable de 16 caracteres.
* El tipo de dato al que pertenece la variable.

Por lo tanto la estructura de la variable es como sigue:

|  |  |
| --- | --- |
| **Estructura** | **Descripción de la Variable** |
| LONGITUD. MAX. |  1 < 35  |
| FORMATO | Mayúscula la primera letra de cada una de las palabras. |
| EJEMPLO | *EjemploDeUpperCamelCase* |

Siendo el nombre que identifica a la variable: **NumeroCuenta**

* 1. **Descripción de la Variable.**

Nombre que se le asignara a la variable para que se le identifique y deberá de estar asociada al motivo para la cual se le declara.

**Ejemplos:** IdCuenta, TipoEstado, Instalacion

* 1. **Variables de Tipo Arreglo**

En el caso de las definiciones de arreglos de elementos se declarara la variable con el prefijo de “ListaUsuarios”, el cual nos dará entender que se trata de una variable del tipo arreglo la cual contendrá de cero a más datos, según el tamaño declarado.

**Ejemplos:** ListaUsuarios

1. **Definición de Controles**

Para poder determinar el nombre de un control dentro de cualquier aplicación de tipo visual, se procede a identificar el tipo al cual pertenece y la función que cumple dentro de la aplicación.

* 1. **Prefijo para el Control**

El prefijo del control será determinado mediante tres caracteres que estarán conformados por las consonantes más representativas del control, es así, por ejemplo; el control Button, estará asociado al prefijo btnEnviar.

* 1. **Nombre descriptivo del Control**

Formado por la descripción de la función que lleva a cabo el control, esta debe ser descrita en forma específica y clara.

|  |  |  |
| --- | --- | --- |
| **Tipo de control** | **Prefijo** | **Ejemplo** |
| Input(Text) | txt | txtNombre |
| Input(Checkbox) | chk | chkSalud |
| Input(RadioButton) | rdn | rdnSexo |
| Select | cmb | cmbDocumentos |
| Button | btn | btnCalcular |
| TextArea | txa | txaDescripcion |

* 1. **Declaración de variables, atributos y objetos**

1. Se debe declarar una variable por línea.

|  |  |
| --- | --- |
| **Título** | **Descripción** |
| **Sintaxis** | [Nombre de la Variable] |
| **Descripción** | El nombre de la variable puede incluir más de un sustantivo los cuales se escribirán juntos.  Si se tuvieran variables que puedan tomar nombres iguales, se le agregará un número asociado (si está dentro de un mismo método será correlativo). |
| **Observaciones** | En la declaración de variables o atributos no se deberá utilizar caracteres como:   * Letra Ñ o ñ. * Caracteres especiales ¡, ^, #, $, %, &, /, (, ), ¿, ‘, +, -, \*, {, }, [, ]. * Caracteres tildados: á, é, í, ó, ú. |
| **Ejemplo** | public $Nombre  var $Nombre  Indica una variable o atributo que guardará un nombre. |

* 1. **Declaración de clases**

|  |  |
| --- | --- |
| **Título** | **Descripción** |
| **Sintaxis** | Class [Nombre de Clase] |
| **Descripción** | El nombre de las clases tendrá una longitud máxima de 30 caracteres y las primeras letras de todas las palabras estarán en mayúsculas. Tipo se refiere a si la clase será: Private, Public o Protected. |
| **Observaciones** | En la declaración de clases no se deberá utilizar caracteres como:   * Letra Ñ o ñ. * Caracteres especiales ¡, ^, #, $, %, &, /, (, ), ¿, ‘, +, -, \*, {, }, [, ]. * Caracteres tildados: á, é, í, ó, ú. |
| **Ejemplo** | Private Class Empleado  Indica una clase Empleado |

* 1. **Declaración de métodos**

|  |  |
| --- | --- |
| **Título** | **Descripción** |
| **Sintaxis** | NombreProcedimiento[(ListaParámetros)] |
| **Descripción** | El nombre del método constará hasta de 25 caracteres.  La primera letra de la primera palabra del nombre será escrita en mayúscula y las siguientes palabras empezarán con letra mayúscula. |
| **Observaciones** | En la declaración de métodos no se deberá utilizar caracteres como:   * Letra Ñ o ñ. * Caracteres especiales ¡, ^, #, $, %, &, /, (, ), ¿, ‘, +, -, \*, {, }, [, ], \_. * Caracteres tildados: á, é, í, ó, ú. |
| **Ejemplo** | Protected function CalcularSueldo(Empleado $Empleado)  Indica un método CalcularSueldo que recibe una variable por valor de tipo Empleado al ámbito de la clase |

* 1. **Declaración de funciones**

|  |  |
| --- | --- |
| **Título** | **Descripción** |
| **Sintaxis** | NombreFuncion[(ListaParámetros)] |
| **Descripción** | El nombre del objeto constará hasta de 25 caracteres, no es necesario colocar un nombre que indique la clase a la cual pertenece.  La primera letra de la primera palabra del nombre será escrita en mayúsculas  El tipo de dato de retorno se coloca al final y será obligatorio colocarlo. |
| **Observaciones** | En la declaración de objetos no se deberá utilizar caracteres como:   * Letra Ñ o ñ. * Caracteres especiales ¡, ^, #, $, %, &, /, (, ), ¿, ‘, +, -, \*, {, }, [, ], \_. * Caracteres tildados: á, é, í, ó, ú. |
| **Ejemplo** | function Sumar( $A, $B)  Indica una función que suma dos variables |