
**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

# ESTÁNDAR DE PROGRAMACIÓN

Curso: Programación II

Docente: Ing. Enrique Lanchipa Valencia

**Salluca Valero, Jhon Francisco (2019063633)**
**Paz Huaychani, Frank Kevin (2019063321)**
**Lostaunau Lozano Juan Gonzalo (2019063323)**
**Neira Machaca, Javier André (2017057984)**
**Gómez Ccalli, Jorge Luis ()**

**Tacna – Perú**

**2021**

# Documento de Estándares de Programación
# Versión 1.0

# Historia de Revisión

| Historial de revisiones | | | | |
| --- | --- | --- | --- | --- |
| **Ítem** | **Fecha** | **Versión** | **Descripción** | **Equipo** |
| | | | | |
| 1 | | 1.0 | Versión Final. | |

# Contenido

# Estándar de Programación

## 1. OBJETIVO

Reglamentar la forma en que se implementará el código fuente del proyecto, pasando, por las variables, controles, clases, métodos, 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 mnemotécnicos con lo que se podrá saber el tipo de datos 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 de este.

# 2. DECLARACION DE VARIABLES

Se propone que la declaración de las variables, se ajusten al motivo para la que se requieran. El mnemotécnico 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 indefinida caracteres.
-
- Alcance de la variable

  A medida que aumenta el tamaño del proyecto, también aumenta la utilidad de reconocer rápidamente el alcance de las variables. Esto se consigue al escribir un prefijo de alcance de una letra delante del tipo de prefijo propio, sin aumentar demasiado la longitud del nombre de las variables.

  Todo prefijo debe ser en minúscula

| Alcance | Prefijo | Ejemplo |
| --- | --- | --- |
| Global | g | gstrNombreUsuario |
| Nivel de la clase | *m* | mblnProgresoDelCálculo |
| Local del procedimiento / método | Ninguno | dblVelocidad |
| Público | p | pCantidadUsuario |
| Privado | pr | prCantidadVenta |

- 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, 2… |
| FORMATO | *Mayúsculas la primera parte y luego la segunda con Minúscula* |
| EJEMPLO | numCuenta - pCantidadUsuario |

### 2.1 Descripción de la Variable.

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

| Estructura | Descripción de la Variable |
| --- | --- |
| LONGITUD. MAX. | 1, 2… |
| FORMATO | *Mayúsculas la primera parte y luego la segunda con Minúscula* |
| EJEMPLO | IdCuenta, TipoEstado. |

### 2.2 Variables de Tipo Arreglo

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

**Ejemplos:** listaTiposDoc

## 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.

### 2.3 Tipo de datos

| Nº | Nombre | Mnemónico | Descripción |
| --- | --- | --- | --- |
| 1 | Bolean | Bln | Valor lógico: verdadero y falso |
| 2 | Byte | Byt | Entero de 8 bits sin signo. |
| 3 | Char | Chr | Un carácter UNICODE de 16 bits |
| 4 | Date | Dat | Formato de fecha/hora |
| 5 | Decimal | Dec | |
| 6 | Double | Dbl | Coma flotante, 64 bits (15-16 dígitos significativos) |
| 7 | Integer | Int | Entero de 32 bits con signo. |
| 8 | Long | Lng | |
| 9 | Object | obj | |
| 10 | Short | srt | |
| 11 | Single | sng | |
| 12 | String | str | Cadena de caracteres |
| 13 | DateTime | Dtt | |
| 14 | Float | Flt | Comas flotantes, 11-12 dígitos significativos. |
| 15 | Image | Img | |
| 16 | Money | Mny | |
| 17 | Nchar | Nch | |
| 18 | Ntext | Ntx | |
| 19 | Nvarchar | Nvc | |
| 20 | Real | Ral | |
| 21 | Smalldatetime | Sdt | |
| 22 | Smallint | Smi | |
| 23 | Smallmoney | Smn | |
| 24 | Text | Txt | |
| 25 | TimeStamp | Tms | |
| 26 | Varbinary | Vbn | |
| 27 | Varchar | Vch | |

### 2.4 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 btn.

### 2.5 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 |
| --- | --- | --- |
| Label | lbl | lblNombre |
| TextBox | txt | txtApellido |
| Button | btn | btnLogin |
| RadioButton | rbt | rdtSeleccion |
| CheckBox | chk | chkRuta |
| DropDownList | cmb | cmbDocumentos |

### 2.6 Declaración de variables, atributos y objetos

1. Se debe declarar una variable por línea.

| Título | Descripción |
| --- | --- |
| **Sintaxis** | [TipoVariable] [Nombre de la Variable] |
| **Descripción** | Todas las variables o atributo tendrán una longitud ilimitadad.<br>El nombre de la variable puede incluir más de un sustantivo los cuales se escribirán juntos.<br>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:<br>• Letra Ñ o ñ.<br>• Caracteres especiales ¡, ^, #, $, %, &, /, (, ), ¿, ‘, +, -, *, {, }, [, ].<br>• Caracteres tildados: á, é, í, ó, ú. |
| **Ejemplo** | Public String nombre<br>Indica una variable o atributo que guardará un nombre. |

### 2.7 Declaración de clases

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

| | |
| --- | --- |
| | Indica una clase Empleado |

## 2.8 Declaración de métodos

| Título | Descripción |
| --- | --- |
| **Sintaxis** | NombreProcedimiento[(ListaParámetros)] |
| **Descripción** | El nombre del método constará hasta de caracteres. Ilimitados<br>La primera letra de la primera palabra del nombre será escrita en mayúscula y los siguientes caracteres de la palabra serán con letra minúscula. Esto aplicando para cada palabra que se encuentre. |
| **Observaciones** | En la declaración de métodos no se deberá utilizar caracteres como:<br>• Letra Ñ o ñ.<br>• Caracteres especiales ¡, ^, #, $, %, &, /, (, ), ¿, ‘, +, -, *, {, }, [, ], _.<br>• Caracteres tildados: á, é, í, ó, ú. |
| **Ejemplo** | Protected CalcularSueldo(String Empleado)<br>Indica un método calcularSueldo que recibe una variable por valor de tipo string al ámbito de la clase |

### 2.9 Declaración de funciones

| Título | Descripción |
| --- | --- |
| **Sintaxis** | [TipoDato] NombreFuncion[(ListaParámetros)] |
| **Descripción** | El nombre del objeto constará hasta de caracteres ilimitados, es necesario colocar un nombre que indique la clase a la cual pertenece.<br>La primera letra de la primera palabra del nombre será escrita en mayúsculas<br>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:<br>• Letra Ñ o ñ.<br>• Caracteres especiales ¡, ^, #, $, %, &, /, (, ), ¿, ‘, +, -, *, {, }, [, ], _.<br>• Caracteres tildados: á, é, í, ó, ú. |
| **Ejemplo** | Public int Sumar(int A, int B)<br>Indica una función que suma dos variables enteras |

### 2.10 Control de versiones de código fuente

Cada modificación realizada será guardada de la forma:

| Título | Descripción |
| --- | --- |
| **Formato** | [NOMBRE DOCUMENTO][ _ ][VERSION][NUMERO VERSIÓN] donde la versión es la fases del archivo. |
| **Descripción** | Se generarán archivos con las siguientes extensiones:.zip o .rar.<br>Por ejemplo: Sistema_Versión1.rar |

### 2.11 Controles ADO.NET

Objetos de ADO.NET Aunque hay miles de objetos disponibles como parte de .NET, es probable que se use ADO.NET como parte de las aplicaciones, por lo tanto, algunos estándares para nombrar los objetos de ADO.NET más comunes. A continuación, se listan los prefijos que se utiliza:

| Componente | Prefijo |
| --- | --- |
| **DataSet** | ds |
| **DataTable** | dt |
| **DataView** | dv |
| **DataRow** | drw |
| **Connection*** | cnx |
| **Command*** | cmm |
| **DataAdapter*** | da |
| **CommandBuilder*** | cbd |
| **DataReader*** | dr |

Ejemplos: de declaración de los objetos ADO.net

- drEmps As New SqlDataReader()
- drCust As New SqlDataReader()
- dsEmps As DataSet
- dsCust As DataSet

### Clases.

El nombre de las clases debe ser auto descriptivo de manera que no se requiera, en lo posible, entrar al código de la función para saber qué es lo que realiza.

El estándar para nombres de clases es usar iniciar con las siglas **Cls,** la cual debe estar escrita en mayúscula seguido del nombre que identifica la clase, la primera letra del nombre debe iniciar con mayúscula

- Ejemplos: ClsCuenta, ClsEmpleado, ClsFactura

**Nota:**
- No se hará uso de los caracteres: Espacio en blanco " ", Carácter de subrayado "_".

## 3. Métodos, Procedimientos y Funciones definidos por el Usuario.

El nombre de las funciones y procedimientos debe ser auto descriptivo de manera que no se requiera, en lo posible, entrar al código de la función para saber qué es lo que realiza.
***verbo-Sustantivo***

El estándar para nombres de procedimiento es usar un Verbo que describa la acción realizada seguida por un sustantivo (objeto sobre el cual actúa). Se recomienda:

- Usar un nombre que represente una acción y un objeto. El nombre del procedimiento debe indicar qué hace el procedimiento a.… o qué hace el procedimiento con....
- El verbo debe estar en infinitivo.
- Ser consistente en el orden de las palabras. Si se va a usar ***verboNombre***, siempre usar ***verboNombre***.
- Ser consistente en los verbos y sustantivos usados. Por ejemplo, si tiene un procedimiento ***AsignarNombre***, en vez de C***olocarNombre***.
- Para la acción **modificar cuentas del cliente** se define:
  **ModificarCuenta**
  Verbo: Modificar
  Sustantivo: Cuenta

**Nota:**
- No se hará uso de los caracteres: Espacio en blanco " ", Caracter de subrayado "_".

- La nomenclatura de argumentos o parámetros pasados a los procedimientos/funciones, así como para valores devueltos por funciones sigue las mismas convenciones que la nomenclatura para variables.

## 4. Tipos de Archivos. Formularios

El nombre de los formularios debe ser auto descriptivo de manera que no se requiera, en lo posible, entrar al código de la función para saber qué es lo que realiza.

El estándar para nombres de clases es usar iniciar con las siglas **Frm,** la cual debe estar escrita en mayúscula seguido del nombre que identifica el formulario, la primera letra del nombre debe iniciar con mayúscula

- Ejemplos: FrmCuenta, FrmEmpleado, FrmFactura

**Nota:**

- No se hará uso de los caracteres: Espacio en blanco " ", Carácter de subrayado "_".