**Plantilla de estándar de codificación y conteo**

|  |  |
| --- | --- |
| Propósito | Establecer un estándar para el código de los programas |
| Estándar de conteo | * Cuente cada línea física como una LOC (incluyendo lineas en blanco) * Sea consistente en lo que se escriba en cada línea física. * Realice el conteo de la cantidad de métodos contando las etiquetas //#Metodo. * El código generado cuenta. |
| Cabeceras de las clases | Iniciar todos los programas con un encabezado descriptivo. |
| Formato de encabezado | /\*  \* NombreClase: nombre de la clase  \* Autores: nombre de los autores - código  \* Fecha: fecha de inicio del programa  \* Descripcion: corta descripción de lo que hace la clase  \*/ |
| Contenidos | Proveer un listado de los métodos de la clase |
| Ejemplo contenido | /\*  \* Métodos de la clase:  \* metodo1(int entrada)  \* metodo2()  \*...  \*/ |
| Clases reusables | * Utilizar la etiqueta //#Reusable antes de las clases que pueden ser reusadas en otras clases o proyectos. |
| Ejemplo de Clase Reusable | //#Reusable  public class LeerArchivo  {  ...  } |
| Identificadores | Usar nombres descriptivos para las clases, métodos y variables. De tal forma que cualquier persona entienda el propósito de ellos.  No usar acrónimos, abreviaciones, nombres no descriptivos, nombres con una sola letra, “ñ”, o tildes en los nombres. |
| Ejemplo identificador | String nombrePaciente; /\*Correcto\*/  String np, nomp, n; /\*No correcto\*/ |
| Comentarios | * Al iniciar cada método se debe comentar con su propósito, entradas y salidas. * El propósito de cada método no debe superar las 2 líneas * Los comentarios deben explicar las instrucciones que no sean utilizadas comúnmente * Los comentarios deben ser descriptivos y claros * Cada linea del cuerpo de un comentario debe tener inicar con \* * Los comentarios de más de una línea inician con /\* y finalizan con \*/ * Los comentarios de una sola línea inician con // y no tienen finalización |
| Comentario bueno | /\*  \*Propósito: Calcular la suma de dos números enteros  \*Entradas: entero, entero  \*Salidas: entero  \*/  //#Metodo:calcularSuma  int calcularSuma(int valor1, int valor2){  ...  } |
| Comentario malo | /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  Propósito: Operar dos números enteros  Entradas: numero, numero  Salidas: numero  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  //#Metodo  int calcularSuma(int valor1, int valor2){  ...  } |
| Espacios en blanco | * No deben existir espacios entre instrucciones * Usar un espacio entre métodos |
| Identacion | * Estructurar las instrucciones dentro de los corchetes respetando la jerarquía entre ellas para ciclos, métodos, condicionales,etc. |
| Ejemplo de Identacion | for (int i=0;i<4;i++)  {  if(i!=2)  {  print “Mensaje”;  }  } |
| Uso de Mayusculas | * Nombres de las clases * Nombre del proyecto * Nombre del paquete o namespace * Cuando es un nombre de clase, metodos o variables de mas de dos palabras se escriben en forma de “camello” |
| Ejemplo del Uso de Mayusculas | namespace Entidades  {  public class CalculosMatematicos  {  public double desviacionEstandar()  {  double resultadoDesviacion;  ...  }  }  } |