

Estándar de Codificación

Departamento de Ciencias Computacionales
Escuela de Ingeniería y Tecnologías de Información
Campus Monterrey
Comité de Programación
Tecnológico de Monterrey Campus Monterrey

A continuación se definen los estándares que se utilizarán en todos los cursos independientemente del lenguaje de programación que se utilice. También, se presentan algunos estándares diferenciadores para cada curso. En cada materia se pueden agregar estándares específicos del lenguaje de programación. Finalmente, se presenta una recomendación que le pudiera ser útil.

Estándares a aplicar en todas las materias relacionados con:**Nombres de identificadores**

- El nombre de los identificadores deben ser representativo, proporcionando información sobre su contenido.
- Los nombres de las variables y funciones/métodos deben utilizar la notación *camelCase* (la primera letra del identificador debe ser minúscula, si está formada por varias palabras, todas excepto la primera, deben iniciar con mayúscula (*nombreDescriptivo*)).
- Los nombres de las constantes deben utilizar solamente letras mayúsculas, si está formada por varias palabras, éstas deben estar separados por el carácter "_" (*NOMBRE_DESCRIPTIVO*).
- Los nombres de los identificadores no deben contener acentos, diéresis ni letra ñ aunque el IDE lo acepte.

Estructura del código

- Los elementos de un estatuto deben estar separados por un espacio en blanco.
- Una literal numérica con decimales debe estar formada por al menos un dígito antes y un dígito después del punto.
- Las expresiones sólo deben utilizar los paréntesis requeridos.
- Una línea de código solamente debe tener un estatuto.
- Los estatutos contenidos en una estructura de control o función/método deben tener la sangría de un *Tab* con respecto a la estructura que lo contiene.
- La llave que abre "{" debe ponerse en la misma línea del encabezado de la estructura de control o función/método.
- La llave que cierra "}" debe estar alineada al lado izquierdo de la estructura de control o función/método que está delimitando.
- Las funciones/métodos o secciones de código deben estar separadas por un renglón en blanco.
- El código solamente debe incluir variables que se utilizan.

Comentarios

- Los comentarios deben ser frases que den valor agregado para facilitar al programador el entendimiento del código. Se deben hacer comentarios de valor y evitar hacer comentarios que son obvios al leer el estatuto referenciado.
- Las primeras líneas de los archivos de código deben incluir al menos el siguiente bloque de comentarios
 - Descripción breve del programa.
 - Una línea con el/los autor/autores: nombre(s) y matrícula(s).
 - Una línea con la fecha de creación/modificación.
- Las funciones/métodos siempre deben incluir comentarios antes de su definición.
- Los comentarios de funciones/métodos deben contener una breve descripción del propósito de:
 - la función,
 - de cada parámetro y
 - del valor de retorno.
- Los comentarios se pueden incluir en secciones de código cuando se requieran para especificar su objetivo.
- Las líneas de código sin uso no deben quedar como comentario.

Nombre de Clases

- El nombre de la clase debe emplear la notación *PascalCase* (la primer letra en cada palabra debe ser mayúscula (*NombreDescriptivo*)) y debe estar en singular.

ESTÁNDARES DIFERENCIADORES POR MATERIA

Los siguientes diferenciadores se deben enfatizar y evaluar en el curso que se especifica y deben ser reforzadas en los cursos que le preceden.

Fundamentos De Programación

- Cada función/método debe realizar una sola tarea y hacerla completa.
- En el bloque de estatutos de un ciclo no utilizar *break* o *continue*.
- Las variables deben estar definidas con un valor inicial.

Programación Orientada A Objetos

- El estatuto *return* debe contener la condición que se requiere para obtener el valor de retorno de la función/método cuando ese sea el caso.
- Dentro de un ciclo se pueden incluir estatutos de decisión siempre y cuando éstos no verifiquen casos que solamente se cumplen una vez y que pueden ser colocados antes o después del ciclo.
- Los programas deben ser modulares.
- Utilizar parámetros en lugar de variables globales.

Estructura de Datos

- Las variables booleanas no deben ser comparadas con los valores *true* o *false*.
- El código y el manejo de los recursos de memoria deben ser eficientes.
- El código, o segmento de código, desarrollado por otros debe ser referenciado explícitamente en comentarios dentro del código y en la documentación correspondiente.

Elaborado en el Departamento de Ciencias Computacionales del Campus Monterrey basado en documentos de estándar previos diseñados por el comité de programación del mismo departamento.

Profesores responsables: Ing. Armandina Leal Flores, Ing. Yolanda Martínez Treviño, Dr. Juan Arturo Nolasco Flores.