**Buenas prácticas ejemplo de los coches**

En base a un ejemplo de modelo vista controlador (MVC), en donde se tiene conocimiento acerca de un listado de coches ordenados por marca, modelos y color, sabemos que la implementación de las buenas practicas de programación incluso en un ejemplo tan sencillo como este.

Pueden llegar a ser un conjunto de reglas ajustables a opcionales u obligatorias (dependiendo de la situación), en donde la intención principal es la de obtener una mejora significativamente de calidad en cuanto a software se refiere; y en base a esto es posible deducir ciertos beneficios como:

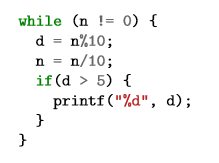
* La facilitación del proceso de desarrollo para programarlo.
* Es posible mantener o mejorar el código legible y de tipo fuente disponible para terceros.
* Puede ser útil para evitar, relacionar o desglosar errores.

En un termino más concreto, las buenas prácticas técnicamente existen en las diversas etapas del proceso de desarrollo de software y son de utilidad principalmente para los desarrolladores de SW, como una especie de cultura agregada y pensado para futuros usos, actualizaciones o consultas de otras personas facilitando la forma de aplicación y entendimiento del mismo (código).

Contemplando a primera instancia la redacción y formato del código a usar de ejemplo en este documento, presenta ciertas estructuras como lo son la buena colocación de llaves, considerando el punto inicio o partida, abriendo la bifurcación entre los parámetros y en estos casos los objetos a desarrollar e implementar.

Se contemplan los parámetros de estructura de programación adecuados a un estándar de buenas prácticas, en donde cada parte tiene su lugar y las “líneas siguen un patrón” para, de esta manera, evitar interferencias o bien malos entendidos al momento de descifrar la apertura, cierre y seguimiento de las llaves de estructuras de control antes mencionadas.

Ejemplo en base a las buenas prácticas de estructura:



La intención de lo anteriormente mencionado, es la de tener un mejor punto de observación y referencia, además, de un consistente estilo de apertura y cierre que evita los problemas al momento de añadir más instrucciones.

Otro consejo que se recomienda es el de mantener el código seguido de las llaves fuera del contexto del programador ordinario, ya que esto influye en la carencia de sentido y mal habito acerca de la contextualización que se le debe de dar a la lectura y/o posible actualización del código.

dentro del ejemplo lo podemos observar ocasionalmente, ya que las palabras no interfieren al abrir o cerrar las llaves. En cuanto a estructuras de separación, se pueden apreciar reconociblemente los espacios después de las comas entre cada nombre torgado.

En cuanto a las propiedades otorgadas se refiere, se puede observar un patrón adecuado, un orden entre llaves y una estructura de programa muy bien definida principalmente; además, en cuestión de recalcar o mantener informado a un futuro desarrollador es posible ver el agregado de comentarios colocados después de algunas líneas, principalmente manteniendo coherencia en aquellas de mayor importancia y dando seguimiento a la forma de comprensión destinada a alguien más, y aun más claro que eso es posible observar y obtener del código un entendimiento sencillo pero eficaz sin necesidad de dar tantas vueltas a su desarrollo.

Por otra parte, muchos de los caracteres siguen un lineamiento acerca de los poco conocidos caracteres especiales en mayor parte originarios del idioma español, es por ello que en ocasiones resultaría imposible la nula presencia de errores ocasionales y de ignorancia causal, teniendo, además, alteraciones ante la apertura de código en otros equipos haciendo del idioma nativo (ingles) un omiso consejo de estructura adecuada.