|  |  |
| --- | --- |
| **Estándares de codificación** | **Descripción** |
| Organización de ficheros, importaciones y paquetería | La paquetería, importaciones y ficheros siempre están en la cabecera de la clase |
| Longitud de línea | No hay longitudes máximas definidas para cada línea de código esta debe ocupar solo un espacio de línea |
| División de línea | No hay división de líneas de código |
| comentarios | Los comentarios se colocaran antes del inicio de cada clase y/o método con una breve descripción de su funcionalidad, el autor y los parámetros de entrada y salida. **Solo se utilizaran comentarios de bloque ejemplo:**  **/\*  \* Esto es un comentario  \* de bloque  \*/** |
| Espacios en blanco | Se colocan espacios en blanco al final de clase, método, comentario y de la paquetería asociada al archivo.java. |
| Clases | Durante el desarrollo de clases se deben seguir las siguientes reglas de formateo: No incluir ningún espacio entre el nombre del método y el paréntesis inicial del listado de parámetros. El carácter inicio de bloque ("{") debe aparecer debajo de la línea que contiene la sentencia de declaración. Los nombres de clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúsculas. Al final de cada sentencia de declaración debe ir un parámetro /\*--\*/ |
| Métodos | Los métodos deben ser verbos escritos en minúsculas. Cuando el método esté compuesto por varias palabras cada una de ellas tendrá la primera letra en mayúsculas. No incluir ningún espacio entre el nombre del método y el paréntesis inicial del listado de parámetros. El carácter inicio de bloque ("{") debe aparecer debajo de la línea que contiene la sentencia de declaración. Al final de cada sentencia de declaración debe ir un parámetro /\*++\*/ |
| Variables | Las variables se escribirán siempre en minúsculas. Las variables compuestas tendrán la primera letra de cada palabra componente en mayúsculas. |
| Constantes | Tendrán que escribirse en mayúsculas. Cuando los nombres de constantes sean compuestos las palabras se separarán entre sí mediante el carácter de subrayado "\_". |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la Definición** | **Java LOC Standard** | **Lenguaje** | **Java** |
| **Autor** | Kevin Ray Marzola Otero | Fecha | 19/02/2017 |
|  |  |  |  |
| **Tipo de conteo** | **Tipo** | **Comentarios** |  |
| Lógico/Físico | Lógico | Ignorar todos los comentarios, etiquetas propias de Java y espacios en blanco. |  |
|  |  |  |  |
| **Tipos de declaraciones** | **Incluidos** | **Comentarios** |  |
| Ejecutables | Si |  |  |
| No Ejecutables | No |  |  |
| Comentarios | No | Ir a la clarificación A |  |
| Cabeceras del programa | No |  |  |
| corchetes | Si | Ir a la clarificación D |  |
| Identificadores | Si |  |  |
| Etiquetas | No | Se refiere a etiquetas del estilo @etiqueta |  |
| Herencias e instancias | Si |  |  |
| Paquetería | Si |  |  |
| Métodos | Si | Ir a la clarificación B |  |
| Clases | Si | Ir a la clarificación C |  |

|  |  |
| --- | --- |
| **Clarificaciones** | **Ejemplos/Casos** |
| A | Solo se aceptarán comentarios que tengan las siguientes estructuras: /\*\* \*\*/, //, /\* \*/ |
| B | Las clases que estén dentro del archivo cargado deben empezar y finalizar con --. Ejemplo: public class Constantes /\*--\*/ { } |
| C | Los métodos que estén dentro del archivo cargado deben empezar y finalizar con ++. Ejemplo: public static final Constantes /\*++\*/ { } |
| D | Los corchetes deben ir separados de la lógica Ejemplo: if(!temporal.values().isEmpty())  {  completo = true; } |