Plantilla de lista de verificación de revisión de código

|  |  |  |  |
| --- | --- | --- | --- |
| Estudiante | Cristian Julián Andrade Murillo | Fecha |  |
| Programa |  | Programa # |  |
| Instructor | Fernelly Carvajal Silva | Lenguaje | Java |

|  |  |
| --- | --- |
| Propósito | Guiar la realización de una revisión de código eficaz. |
| General | * Revisar el programa completo para cada categoría de lista de verificación; ¡No intente revisar más de una categoría a la vez! * Al completar cada paso de revisión, marque ese elemento en el cuadro de la derecha. * Complete la lista de verificación para un programa o unidad de programa antes de revisar el siguiente. |

**Nota: Debido a que en la revisión de diseño se evaluó el que los nombres de los atributos y de los métodos cumplen con el estándar de código, si en el código están igual que en el diseño entonces son válidos.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| #Defect log | Ítem de valoración | Descripción | Cumple | No cumple | Observación |
| 1 | Encabezado | Todas las clases llevan su encabezado con todos los campos completos. |  |  |  |
| 2 | Paquetes | Se declaró a que paquete pertenece cada clase. |  |  |  |
| Hay un salto de línea entre el paquete y el encabezado. |  |  |  |
| 3 | Importaciones | Hay un salto de línea entre las importaciones y el paquete. |  |  |  |
| No hay importaciones invalidas |  |  |  |
| No hay importaciones que no se utilizan. |  |  |  |
| Cada importación va en un LOC diferente. |  |  |  |
| 4 | Encapsulamiento | Los atributos de las clases deben ser privados. |  |  |  |
| 5 | Trazabilidad con diseño en el nombre de los atributos de clase. | El tipo de atributo es el mismo que en el diseño. |  |  |  |
| Los nombres de los atributos son los mismos del diseño. |  |  |  |
| Cada uno de los atributos que aparecen en el diseño están en el código. |  |  |  |
| 6| | Trazabilidad con diseño en el nombre de los métodos. | El tipo de método es el mismo que en el diseño. |  |  |  |
| El modificador de acceso del método es el mismo del diseño. |  |  |  |
| Los nombres de los métodos son los mismos del diseño. |  |  |  |
| Aparecen todos los parámetros de los métodos y son los mismos que en el diseño. Su nombre y tipo también deben ser iguales al que está en el diseño. |  |  |  |
| 7 | Trazabilidad con diseño de las variables de método. | El tipo de variable es el mismo que en el diseño. |  |  |  |
| Los nombres de las variables son los mismos del diseño. |  |  |  |
| Cada uno de las variables que aparecen en el diseño están en el código. |  |  |  |
| 8 | operadores lógicos | Los operadores lógicos usados en las diferentes operaciones son los adecuados. |  |  |  |
| 9 | Uso de las llaves | La llave de apertura va en el mismo LOC que el nombre del método o el control de flujo. |  |  |  |
| La llave de cierre va en un LOC individual para esta. |  |  |  |
| En la llave de cierre del do…while, en ese mismo LOC puede ir la expresión del while. |  |  |  |
| En la llave de cierre de un “if” se puede colocar en ese mismo LOC el “else” o el “else if”. |  |  |  |
| 10 | Espacios en blanco | Entre las importaciones no hay espacios en blanco. |  |  |  |
| En la declaración de los atributos no hay espacios en blanco entre estos. |  |  |  |
| Antes de la apertura un control de flujo hay un espacio. |  |  |  |
| Entre el final de las instrucciones y la llave de cierre no hay espacios. |  |  |  |
| Entre la llave de cierre y las instrucciones siguientes hay un espacio en blanco (un salto de línea o un enter). |  |  |  |
| Se debe dejar un espacio entre cada método. |  |  |  |
| 11 | Indentación | Cada instrucción luego de una apertura de llaves debe llevar un TAB de indentación. |  |  |  |
| El cierre de la llave debe estar en el mismo nivel que el de apertura. |  |  |  |
| 12 | Comentarios | Antes de cada método hay un comentario del funcionamiento de este. |  |  |  |
| Todas las declaraciones de atributos de clase o variables de método tienen comentarios para indicar su finalidad. |  |  |  |