**Checklist de Revisión del Código**

|  |  |  |  |
| --- | --- | --- | --- |
| Estudiante | Erick Francisco González Martínez |  |  |
| Programa | 7 | # Programa | 7 |
| Instructor | Ing. Adriana González Ugaide. | Lenguaje | C++ |

|  |  |
| --- | --- |
| Propósito | Guiarte en la conducción de una revisión efectiva de tu código |
| General | * Escribe en el primer renglón el nombre de las partes que vas a revisar (aquellas que son nuevas o que van a ser modificadas) * Revisa tu código parte por parte. No continúes con la siguiente parte hasta que no hayas revisado por completo la parte anterior. * Cuando estés revisando una parte, revisa una categoría a la vez * Cuando termines de revisar una categoría para una parte, márcala como revisada |

|  |  |  |
| --- | --- | --- |
| Nombre de las partes: | | InputReader |
| Completo | -Verificar que se cumplen todos los requerimientos del sistema | ✓ |
| Incluir | -Verificar que todas las clases estén llamadas en la clase principal  -Verificar que todas las librerías estén llamadas en sus respectivas clases | ✓ |
| Inicialización | -Verificar constructores  - Vacíos  - Con parámetros  - Copia  -Validación de los parámetros de los constructores | ✓ |
| Llamadas | -Verificar las llamadas de acuerdo al formato de la función  Funciones miembro  Funciones amigas  -Verificar los modificadores de acceso  Público  Privado  -Verificar llamado de destrucción de objetos  -Verificar el formato de las llamadas  Tipo  Cantidad | ✓ |
| Apuntadores | -Verificar apuntadores a basura  -Verificar apuntadores a vacío | ✓ |
| Formato de salida | -Verificar líneas de salida  Espacio  Tabuladores  Delimitadores  Saltos de Línea  -Verificar sintaxis de texto | ✓ |
| Estándar | -Verificar la documentación de funciones y variables  -Verificar la separación de clases en diferentes archivos | ✓ |

|  |  |  |
| --- | --- | --- |
| Nombre de las partes: | | OutputHandler |
| Completo | -Verificar que se cumplen todos los requerimientos del sistema | ✓ |
| Incluir | -Verificar que todas las clases estén llamadas en la clase principal  -Verificar que todas las librerías estén llamadas en sus respectivas clases | ✓ |
| Inicialización | -Verificar constructores  - Vacíos  - Con parámetros  - Copia  -Validación de los parámetros de los constructores | ✓ |
| Llamadas | -Verificar las llamadas de acuerdo al formato de la función  Funciones miembro  Funciones amigas  -Verificar los modificadores de acceso  Público  Privado  -Verificar llamado de destrucción de objetos  -Verificar el formato de las llamadas  Tipo  Cantidad | ✓ |
| Apuntadores | -Verificar apuntadores a basura  -Verificar apuntadores a vacío | ✓ |
| Formato de salida | -Verificar líneas de salida  Espacio  Tabuladores  Delimitadores  Saltos de Línea  -Verificar sintaxis de texto | ✓ |
| Estándar | -Verificar la documentación de funciones y variables  -Verificar la separación de clases en diferentes archivos | ✓ |

|  |  |  |
| --- | --- | --- |
| Nombre de las partes: | | Main |
| Completo | -Verificar que se cumplen todos los requerimientos del sistema | ✓ |
| Incluir | -Verificar que todas las clases estén llamadas en la clase principal  -Verificar que todas las librerías estén llamadas en sus respectivas clases | ✓ |
| Inicialización | -Verificar constructores  - Vacíos  - Con parámetros  - Copia  -Validación de los parámetros de los constructores | ✓ |
| Llamadas | -Verificar las llamadas de acuerdo al formato de la función  Funciones miembro  Funciones amigas  -Verificar los modificadores de acceso  Público  Privado  -Verificar llamado de destrucción de objetos  -Verificar el formato de las llamadas  Tipo  Cantidad | ✓ |
| Apuntadores | -Verificar apuntadores a basura  -Verificar apuntadores a vacío | ✓ |
| Formato de salida | -Verificar líneas de salida  Espacio  Tabuladores  Delimitadores  Saltos de Línea  -Verificar sintaxis de texto | ✓ |
| Estándar | -Verificar la documentación de funciones y variables  -Verificar la separación de clases en diferentes archivos | ✓ |