**Checklist de Revisión del Diseño**

|  |  |  |  |
| --- | --- | --- | --- |
| Estudiante | Patricia Palula Aguilar | Fecha | 30 sept 2021 |
| Programa | 4 | # Programa | 4 |
| Instructor | Adriana González Ugalde | Lenguaje | C++ |

|  |  |
| --- | --- |
| Propósito | Guiarte en la conducción de una revisión efectiva de tu diseño |
| 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 diseño 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: | | Main | Simpson | FunctionT |  |
| Completo | Verificar que el diseño cumpla con todos los requerimientos aplicables.   * Todos los outputs especificados son producidos. * Todos los inputs necesarios están previstos. * Todos los includes requeridos están establecidos. | X | X | X |  |
| Limites Externos | Cuando el diseño asuma o necesite de limites externos, determina si el comportamiento es correcto en valores nominales, límites y fuera de los límites. | X | X | X |  |
| Lógica | * Verificar que la secuencia del programa es apropiada.   Pilas, listas, y demás estén en el orden adecuado.  Recursión se desenvuelve correctamente   * Verificar que todos los loops son iniciados, incrementados y terminados apropiadamente. * Examinar cada condicional y verificar todos los casos | X | X | X |  |
| Limites Internos | Donde el diseño asumo o necesite de límites internos, determina si el comportamiento es correcto en valores nominales, límites y fuera de los límites | X | X | X |  |
| Casos Especiales | * Revisar todos los casos especiales. * Asegurar la operación con valores vacíos, llenos, mínimos, máximos, negativos y ceros para todas las variables. * Proteger contra condiciones fuera de los límites, overflow y underflow. * Asegurarse de que las condiciones “imposibles” sean imposibles. * Revisar todas las posibles condiciones incorrectas o erróneas | X | X | X |  |
| Uso Funcional | * Verificar que todas la funciones, procedimientos o métodos están completamente entendidos y son usados propiamente. * Verificar que todas las abstracciones referenciadas externamente estén definidas con precisión. | X | X | X |  |
| Consideraciones del sistema | * Verifica que el programa no cause que los límites del sistema sean excedidos. * Verificar que todos los datos relacionados con la seguridad sean de fuentes confiables. * Verificar que todas las condiciones de seguridad cumplan con las especificaciones de seguridad | X | X | X |  |
| Nombres | Verificar que:   * Todos los nombres especiales sean claros, definidos y autenticados. * Los alcances de las variables y parámetros son evidentes o definidos. | X | X | X |  |
| Estándares | Asegurarse de que el diseño sea acorde con todos los estándares de diseño aplicables | X | X | X |  |