**Checklist de Revisión del Diseño**

|  |  |  |  |
| --- | --- | --- | --- |
| Estudiante | Carlos Gerardo Herrera Cortina – A00821946 | Fecha | 14-mar-21 |
| Programa | 3 | # Programa | 3 |
| 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: | |  |  |  |  |
| Completo | Verifíca que el diseño cubra todos los requerimientos aplicables:   * Todos los outputs especificados son producidos. * Todos los inputs necesarios son dados. * Todos los includes requeridos se indican. | ✓ |  |  |  |
| Límites Externos | Donde el diseño asume o descansa sobre límites externos, determina que el comportamiento es correcto en valores nominales, límites, y más allá de ellos. | ✓ |  |  |  |
| Lógica | * Verifica que la secuencia del programa es adecuada.   Stacks, listas, y demás están en orden correcto.  Recursividad desenrolla correctamente.   * Verifica que todos los ciclos se inician, incrementan y terminan apropiadamente. * Examina cada enunciado condicional y verifica todos los casos. | ✓ |  |  |  |
| Límites Internos | Donde el diseño asume o descansa sobre límites internos, determina que el comportamiento es correcto en valores nominales, límites, y más allá de ellos. | ✓ |  |  |  |
| Casos Especiales | * Checa todos los casos especiales. * Asegúrate del desempeño adecuado con valores vacíos, llenos, mínimos, máximos, negativos y cero para todas las variables. * Protege contra condiciones fuera-de-límites, desbordamiento y subdesbordamiento. * Asegúrate de que las condiciones “imposibles” son realmente imposibles. * Maneja todas las condiciones de error o incorrectas posibles. | ✓ |  |  |  |
| Uso Funcional | * Verifica que todas las funciones, procedimientos o métodos están completamente entendidos y son usados apropiadamente. * Verifica que todas las abstracciones referenciadas externamente están definidas con precisión. | ✓ |  |  |  |
| Consideraciones del Sistema | * Verifica que el programa no causa que los límites del sistema sean excedidos. * Verifica que todos los datos sensibles sean de fuentes confiables. * Verifica que todas las condiciones de seguridad son conforme a las especificaciones de seguridad. | ✓ |  |  |  |
| Nombres | Verifica que   * Todos los nombres especiales son claros, definidos, y autenticados. * El alcance de todas las variables y parametros son evidentes o están definidos. * Todos los items se usan con su respectivo alcance. | ✓ |  |  |  |
| Estándares | Verifica que el diseño es conforme a los estándares de diseño aplicables. | ✓ |  |  |  |
| Principio de Responsabilidad Única | Asegúrate de que cada clase tiene una única responsabilidad y no asume otras ya cubiertas por otras clases. | ✓ |  |  |  |