**DATOS DEL APRENDIZ:**

|  |  |  |  |
| --- | --- | --- | --- |
| **PRIMER APELLIDO: Suaza**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **SEGUNDO APELLIDO: Sánchez** | **NOMBRE: Jhon Camilo**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **TIPO DOC.** | **NÚMERO DOCUMENTO** | **LUGAR EXPEDICIÓN** | **FICHA** |
| **T.I** | **1077230121** | **Neiva-Huila** | **3145555** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **LUGAR Y FECHA DE APLICACIÓN: Neiva-Huila Formación** | | | **DÍA: 1** | **MES: 05** | | **AÑO: 2025** |
| **NOMBRE INSTRUCTOR:** |  | | | | | |
| **INFORMACIÓN CURRICULAR** | | | | | | |
| **Programa de Formación** | Tecnología en Análisis y Desarrollo de Software | | | | | |
| **Fase Proyecto:** | Análisis | | | | | |
| **Actividad de proyecto:** | ESPECIFICAR EL MODELO CONCEPTUAL DEL SOFTWARE | | | | | |
| **Actividad de Aprendizaje:** | Taller Programación Secuencial | | | | | |
| **Competencia:** | ANÁLISIS DE LA ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE. | | | | | |
| **Resultados de Aprendizaje:** | Desarrollar procesos lógicos a través de la implementación de algoritmos. | | | | | |
| **Evidencia:** | Taller de Programación Secuencial | | | | | |
| **Tipo Evidencia:** | **Conocimiento:** | **Desempeño**: x \_\_\_ | | | **Producto: \_X\_\_** | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criterio** | **Excelente (4)** | **Bueno (3)** | **Regular (2)** | **Insuficiente (1)** | **Evaluación** |
| **Comprensión del problema** | Interpreta completamente el problema; identifica claramente qué se debe resolver. | Interpretación adecuada con leves omisiones. | Interpretación ambigua o parcialmente incorrecta. | No comprende el problema o lo interpreta erróneamente. |  |
| **Datos de entrada y salida** | Identifica clara y correctamente los datos de entrada y salidas esperadas | Identificación mayormente correcta con algún error menor. | Muestra confusión o identifica solo parcialmente los datos. | No identifica o identifica incorrectamente los datos. |  |
| **Diseño del algoritmo** | Algoritmo lógico, eficiente, estructurado y bien planteado. | Diseño claro con algunas áreas de mejora. | Diseño incompleto o poco eficiente. | Diseño incorrecto o no resuelve el problema. |  |
| **Diagrama de flujo** | Correcto, completo, con símbolos y secuencia adecuada. | Contiene detalles menores a mejorar. | Algunos errores de forma, símbolos o lógica. | Incompleto o con errores graves de secuencia. |  |
| **Implementación en código** | Código funcional, sin errores, con buen estilo y organización. | Código funcional con leves errores o desorden. | Código con errores frecuentes o poco organizado. | No compila o no cumple con el objetivo. |  |
| **Prueba de escritorio** | Clara, paso a paso, con valores correctos y coherentes con el algoritmo. | Prueba adecuada, con leves errores. | Prueba incompleta o confusa. | Prueba ausente o incorrecta. |  |
| **Estructuras utilizadas** | Uso adecuado y eficiente de estructuras de control y datos. | Estructuras bien utilizadas con pequeños errores. | Uso básico o parcialmente incorrecto. | Estructuras ausentes o mal empleadas. |  |
| **Análisis y optimización** | Analiza la complejidad y propone mejoras. | Mención general de eficiencia o propuesta simple de mejora. | Análisis superficial o con errores. | No se analiza ni menciona la eficiencia. |  |
| **Documentación y claridad** | Código comentado claramente, nombres Descriptivos, presentación cuidada. | Comentarios presentes pero mejorables. | Comentarios escasos o nombres poco informativos. | Sin comentarios o ilegible. |  |

|  |
| --- |
| Observaciones: |
|  |