|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ACTA No. 001** | | | | | | |
| **NOMBRE DEL COMITÉ O DE LA REUNIÓN:**  Generar el plan de mejoramiento al aprendiz: | | | | | | |
| **CIUDAD Y FECHA:** | | Neiva 27/08/2025 | | **HORA INICIO:**  8:00 am | | **HORA FIN:**  12:00m |
| **LUGAR Y/O ENLACE:** | | Ambiente 209-3 NTIC industria CIES | | **DIRECCIÓN / REGIONAL /**  **CENTRO: Huila - Neiva** | | |
| **AGENDA O PUNTOS PARA DESARROLLAR:**   1. Validar las acciones a realizar para alcanzar las competencias y resultados de aprendizaje pendientes por el aprendiz | | | | | | |
| **OBJETIVOS DE LA REUNIÓN:**   1. Concertar los tiempos, actividades y espacios para la ejecución del plan de mejoramiento | | | | | | |
| **DESARROLLO DE LA REUNIÓN** | | | | | | |
| De conformidad con el **Reglamento del Aprendiz SENA** y en razón al **incumplimiento de las actividades solicitadas dentro del plan concertado establecido para el Bootcamp ADSO 2025, programado y autorizado por coordinación**, se formula el siguiente Plan de Mejoramiento.  **Justificación y marco normativo**  El presente plan se sustenta en los siguientes artículos del Reglamento del Aprendiz:   * **Artículo 18:** Define como falta cualquier acción u omisión que afecte el proceso de formación y la convivencia. * **Artículo 19:** Clasifica las faltas en académicas y disciplinarias. El incumplimiento de actividades formativas pactadas se considera una falta académica. * **Artículo 20:** Establece la calificación de faltas (leve, grave, gravísima) según el impacto en el proceso formativo. * **Artículo 21:** Señala los criterios de calificación y análisis, tales como la afectación al desarrollo de la formación, la reincidencia y el nivel de compromiso del aprendiz.   Este plan busca ofrecer una **oportunidad de corrección antes de proceder a medidas sancionatorias**, conforme a los Artículos 27 y 28 del reglamento.  **1. Situación Identificada**  Incumplimiento en la entrega y desarrollo de actividades establecidas en el plan concertado del Bootcamp ADSO 2025, afectando el cronograma aprobado por la coordinación académica.  **2. Objetivo del Plan de Mejoramiento**  Recuperar los compromisos formativos incumplidos, garantizando la finalización adecuada de las actividades programadas y evitando la reincidencia.  Información del aprendiz   |  |  |  | | --- | --- | --- | | **DOCUMENTO** | **NOMBRES** | **APELLIDOS** | |  |  |  |   El aprendiz decide trabajar con un proyecto existente: **NOMBRE PROYECTO** con un grupo de trabajo.  Se revisan los juicios y se establece:   1. Que los resultados pendientes corresponden a la competencia principal del programa de formación que tiene que ver dos competencias.   Establecer requisitos de la solución de software de acuerdo con estándares y procedimiento técnico Resultado 1: Caracterizar los procesos de la organización de acuerdo con el software a construir Objetivos de Mejoramiento:   * Mejorar la identificación de los procesos organizacionales y las áreas de incidencia. * Aplicar técnicas de análisis de procesos con mayor efectividad. * Crear diagramas de procesos claros.   Ejercicios Propuestos:   1. Ejercicio 1: Análisis de la Estructura Organizacional    * Objetivo: Analizar la estructura organizacional de una empresa real o ficticia.    * Instrucciones: Investigar una empresa (puede ser de su entorno) y, utilizando herramientas como un organigrama, identificar los principales procesos internos que afectan directamente al software que se va a construir.    * Resultado Esperado: Identificación clara de los procesos, áreas de incidencia y actores clave para el proyecto de software. 2. Ejercicio 2: Diagramación de Procesos    * Objetivo: Elaborar un diagrama de procesos para un área específica de la organización.    * Instrucciones: Usar herramientas como Lucidchart o Bizagi para diagramar un proceso de la empresa relacionado con el software que se construirá. Incluir las áreas de impacto y los sistemas involucrados.    * Resultado Esperado: Diagrama claro que represente la secuencia de acciones y decisiones en el proceso. 3. Ejercicio 3: Análisis de Alcance del Proyecto    * Objetivo: Definir las fronteras y el alcance del software.    * Instrucciones: Realizar una reunión simulada con clientes o compañeros de trabajo para definir las fronteras y el alcance del sistema. Crear un documento con los resultados y acuerdos.    * Resultado Esperado: Documento que establezca claramente las fronteras del sistema, delimitando lo que estará y no estará dentro del proyecto.  Resultado 2: Recolectar información del software a construir de acuerdo con las necesidades del cliente Objetivos de Mejoramiento:   * Mejorar la capacidad de identificar fuentes de requisitos. * Diseñar instrumentos de recolección adecuados. * Organizar y analizar la información recolectada.   Ejercicios Propuestos:   1. Ejercicio 1: Identificación de Fuentes de Requisitos    * Objetivo: Identificar las fuentes de requisitos dentro de un proyecto de software.    * Instrucciones: Realizar una entrevista simulada con un cliente para identificar los requisitos del sistema. Registrar todas las fuentes de requisitos (clientes, usuarios, investigación documental).    * Resultado Esperado: Lista organizada de fuentes de requisitos con detalles sobre el tipo de información que proporcionan. 2. Ejercicio 2: Diseño de Instrumentos de Recolección de Requisitos    * Objetivo: Diseñar instrumentos para recolectar requisitos.    * Instrucciones: Crear un cuestionario estructurado para recolectar requisitos, asegurándose de que cubra aspectos funcionales y no funcionales del software. Usar normas y procedimientos técnicos.    * Resultado Esperado: Cuestionario que sea efectivo para capturar la información necesaria para el desarrollo del sistema. 3. Ejercicio 3: Análisis de Información Recolectada    * Objetivo: Organizar la información recolectada de manera eficiente.    * Instrucciones: Utilizando la información de un cliente ficticio o real, organizar los requisitos recolectados en una tabla de análisis, clasificándolos según su tipo (funcionales, no funcionales, operacionales, etc.).    * Resultado Esperado: Una tabla bien organizada que permita ver claramente las categorías y prioridades de los requisitos.  Resultado 3: Establecer los requisitos del software de acuerdo con la información recolectada Objetivos de Mejoramiento:   * Mejorar la documentación de requisitos y alinearla con las normativas y estándares. * Mejorar la presentación de los informes de requisitos.   Ejercicios Propuestos:   1. Ejercicio 1: Elaboración de Documentación de Requisitos    * Objetivo: Crear la documentación de requisitos siguiendo normativas y estándares.    * Instrucciones: Redactar una especificación de requisitos utilizando un formato estándar (puede basarse en la norma IEEE 830). Asegurarse de incluir los requisitos funcionales y no funcionales de manera detallada.    * Resultado Esperado: Documento que cumpla con las normativas establecidas, que sea comprensible y detallado. 2. Ejercicio 2: Presentación del Informe de Requisitos    * Objetivo: Presentar los requisitos de manera clara y profesional.    * Instrucciones: Crear una presentación de los requisitos del software, utilizando PowerPoint o cualquier otra herramienta, para compartirla con el cliente. Asegurarse de que la presentación esté clara, organizada y visualmente atractiva.    * Resultado Esperado: Presentación que comunique efectivamente los requisitos de manera visual y concisa.  Resultado 4: Validar el informe de requisitos de acuerdo con las necesidades del cliente Objetivos de Mejoramiento:   * Mejorar el proceso de validación del informe de requisitos. * Aumentar la eficiencia en la validación mediante revisiones y pruebas continuas.   Ejercicios Propuestos:   1. Ejercicio 1: Validación con el Cliente    * Objetivo: Validar los requisitos con el cliente para asegurarse de que todo esté alineado con sus expectativas.    * Instrucciones: Simular una sesión de validación de requisitos con el cliente. Presentar el informe de requisitos y obtener retroalimentación. Realizar ajustes según sea necesario.    * Resultado Esperado: Informe validado por el cliente, con correcciones menores, si es necesario. 2. Ejercicio 2: Prototipo y Validación    * Objetivo: Validar los requisitos utilizando un prototipo o versión preliminar.    * Instrucciones: Crear un prototipo básico del sistema utilizando herramientas como Figma o Adobe XD. Presentar el prototipo al cliente y registrar su feedback.    * Resultado Esperado: Prototipo funcional que permita visualizar los requisitos antes de la implementación final. 3. Ejercicio 3: Validación de Requisitos con Casos de Uso    * Objetivo: Validar los requisitos mediante casos de uso.    * Instrucciones: Crear casos de uso basados en los requisitos del cliente y presentarlos a los usuarios finales para validar si los requisitos cubren sus necesidades.    * Resultado Esperado: Casos de uso que validen los requisitos, con retroalimentación de los usuarios finales.   Evaluar requisitos de la solución de software de acuerdo con metodologías de análisis y estándares 1. Planificar Actividades de Análisis de Acuerdo con la Metodología Seleccionada Competencias Específicas:   * Identificar metodologías de desarrollo de software. * Establecer las actividades de análisis de acuerdo con la metodología seleccionada.  Ejercicios Propuestos:  1. Ejercicio 1: Selección de Metodología de Desarrollo    * Objetivo: Identificar la metodología de desarrollo de software más adecuada para el proyecto a desarrollar.    * Instrucciones: Analizar las características del software a desarrollar (por ejemplo, si es un sistema de gestión, una app móvil, etc.) y seleccionar la metodología de desarrollo más adecuada (ágil, cascada, DevOps, etc.).    * Resultado Esperado: Explicar por qué se seleccionó una metodología determinada y cómo se alinea con las necesidades del proyecto. 2. Ejercicio 2: Establecimiento de Actividades de Análisis    * Objetivo: Establecer actividades claras de análisis en función de la metodología seleccionada.    * Instrucciones: Crear un plan de actividades de análisis detallado según la metodología seleccionada. Esto puede incluir fases como la recopilación de requisitos, la elaboración de diagramas de procesos, y la validación de los mismos.    * Resultado Esperado: Documento que describa las actividades de análisis a realizar en cada fase del proyecto, con una timeline para la ejecución.  2. Modelar las Funciones del Software de Acuerdo con el Informe de Requisitos Competencias Específicas:   * Interpretar el informe de requisitos para modelar las funciones del software. * Elaborar diagramas de casos de uso utilizando UML. * Realizar diagramas de actividades para detallar los casos de uso. * Generar plantillas extendidas de casos de uso. * Representar el negocio en términos de clases abstractas para crear un modelo de dominio consistente. * Elaborar un modelo entidad-relación.  Ejercicios Propuestos:  1. Ejercicio 1: Interpretación del Informe de Requisitos    * Objetivo: Comprender los requisitos del software para modelar sus funciones de manera precisa.    * Instrucciones: Analizar un informe de requisitos proporcionado y extraer las funcionalidades clave del software. Luego, mapear estas funcionalidades en un modelo conceptual del software.    * Resultado Esperado: Documento que demuestre cómo los requisitos se traducen en funciones específicas del software a desarrollar. 2. Ejercicio 2: Diagramas de Casos de Uso con UML    * Objetivo: Crear diagramas de casos de uso utilizando el lenguaje de modelado unificado (UML).    * Instrucciones: Usar herramientas como Lucidchart o Visual Paradigm para crear diagramas de casos de uso para las principales funcionalidades del sistema. Asegurarse de incluir actores, casos de uso y las relaciones entre ellos.    * Resultado Esperado: Diagramas de casos de uso que representen las funcionalidades principales del sistema de acuerdo con los estándares UML. 3. Ejercicio 3: Diagramas de Actividades de Casos de Uso    * Objetivo: Crear diagramas de actividades detallados para cada caso de uso.    * Instrucciones: Para cada caso de uso, crear un diagrama de actividades que muestre las acciones paso a paso dentro del proceso, utilizando símbolos estándar de UML.    * Resultado Esperado: Diagramas de actividades que describan las interacciones y secuencias de eventos de cada caso de uso. 4. Ejercicio 4: Plantillas Extendidas de Casos de Uso    * Objetivo: Desarrollar plantillas extendidas de casos de uso que expresen la intención de las acciones a desarrollar.    * Instrucciones: Crear plantillas extendidas de casos de uso para los casos más críticos del sistema. Las plantillas deben incluir una descripción detallada de cada paso, precondiciones, postcondiciones, excepciones, etc.    * Resultado Esperado: Plantillas extendidas que sirvan como base para el desarrollo detallado del sistema. 5. Ejercicio 5: Creación del Modelo de Dominio con Clases Abstractas    * Objetivo: Representar el negocio utilizando clases abstractas para generar un modelo de dominio consistente.    * Instrucciones: Modelar el negocio en términos de clases abstractas y relaciones entre ellas. Usar herramientas UML para representar clases, asociaciones y herencia.    * Resultado Esperado: Diagrama de clases UML que represente el modelo de dominio de manera consistente. 6. Ejercicio 6: Elaboración del Modelo Entidad-Relación    * Objetivo: Crear el modelo entidad-relación (ER) basado en los requisitos del software.    * Instrucciones: Usar una herramienta de modelado para generar el modelo ER que refleje las entidades principales del sistema y sus relaciones.    * Resultado Esperado: Un diagrama ER que muestre todas las entidades relevantes, sus atributos y relaciones, y que sirva de base para la base de datos del software.  3. Desarrollar Procesos Lógicos a Través de la Implementación de Algoritmos Competencias Específicas:   * Solucionar problemas de lógica proposicional. * Incorporar habilidades propias en la solución de problemas lógicos.  Ejercicios Propuestos:  1. Ejercicio 1: Solución de Problemas de Lógica Proposicional    * Objetivo: Resolver problemas de lógica proposicional aplicando principios matemáticos y lógicos.    * Instrucciones: Resolver una serie de problemas de lógica proposicional utilizando tablas de verdad, operadores lógicos y simplificación de expresiones. Aplicar la solución al desarrollo de un algoritmo.    * Resultado Esperado: Soluciones claras y justificadas que expliquen los pasos tomados para resolver el problema lógico. 2. Ejercicio 2: Desarrollo de Algoritmos para Resolución de Problemas    * Objetivo: Desarrollar un algoritmo para resolver un problema lógico o de cálculo específico.    * Instrucciones: Crear un algoritmo utilizando pseudocódigo o diagramas de flujo que resuelva un problema lógico planteado (por ejemplo, verificar si un número es primo o si una expresión lógica es válida).    * Resultado Esperado: Algoritmo bien documentado y funcional para resolver el problema.  Verificar los Modelos Realizados en la Fase de Análisis de Acuerdo con lo Establecido en el Informe de Requisitos Competencias Específicas:   * Elaborar listas de chequeo para validación. * Evaluar la calidad de los artefactos generados en la fase de análisis. * Realizar mejoras a la documentación. * Desarrollar prototipos iniciales del software.  Ejercicios Propuestos:  1. Ejercicio 1: Elaboración de una Lista de Chequeo    * Objetivo: Crear una lista de chequeo para validar los artefactos de análisis.    * Instrucciones: Desarrollar una lista de chequeo para verificar que los diagramas de casos de uso, diagramas de clases y los informes de requisitos estén completos y sean coherentes con lo solicitado.    * Resultado Esperado: Lista de chequeo detallada para la validación de los artefactos generados. 2. Ejercicio 2: Evaluación del Informe de Análisis    * Objetivo: Evaluar la calidad de los artefactos de análisis con respecto a los requisitos.    * Instrucciones: Tomar un informe de requisitos y evaluar si se cumplen todos los aspectos solicitados en el análisis. Realizar un informe de evaluación con sugerencias de mejora.    * Resultado Esperado: Informe de evaluación con recomendaciones de mejora en la documentación de análisis. 3. Ejercicio 3: Prototipo Inicial del Software    * Objetivo: Crear un prototipo básico del software según los casos de uso identificados.    * Instrucciones: Crear un prototipo básico (puede ser en una herramienta como Figma, Balsamiq o incluso en un IDE) que represente la interfaz de usuario y funcionalidades básicas del software.    * Resultado Esperado: Un prototipo funcional que muestre las interacciones básicas del sistema según los casos de uso identificados.  Indicadores de Éxito de los Ejercicios:  1. Mejora en la Identificación de Requisitos: El equipo es capaz de identificar correctamente todas las fuentes de requisitos y organizarlas de manera eficaz. 2. Documentación Completa: La documentación de requisitos es clara, detallada y cumple con las normativas establecidas. 3. Validación Eficaz: Los requisitos y prototipos son validados por los clientes sin necesidad de cambios sustanciales. 4. Satisfacción del Cliente: El cliente aprueba los requisitos y el prototipo sin objeciones significativas. 5. Comprensión de Requisitos: Los estudiantes son capaces de interpretar correctamente los requisitos y modelar las funciones del software con precisión. 6. Diagramas Correctos y Claros: Los diagramas de casos de uso, actividades y clases son coherentes, completos y siguen los estándares UML. 7. Documentación Completa: Se produce documentación clara y precisa para todas las fases del análisis y modelado. 8. Desarrollo de Algoritmos Eficientes: Los algoritmos desarrollados para la resolución de problemas lógicos son correctos y eficientes, aplicando conceptos de lógica proposicional y desarrollo lógico. 9. **Resolución Eficiente de Problemas con Algoritmos**: Los algoritmos creados resuelven correctamente los problemas propuestos utilizando las estructuras adecuadas. 10. **Calidad del Código**: El código está bien estructurado, con funciones y procedimientos bien definidos que simplifican su mantenimiento y lectura. 11. **Gestión de Archivos Correcta**: Los archivos de datos se gestionan correctamente, permitiendo el ingreso, modificación y eliminación de información. 12. **Validación Exitosa de Modelos**: Los artefactos de análisis son evaluados correctamente mediante listas de chequeo, y el prototipo del software refleja las necesidades del cliente.   Este plan de ejercicios permitirá que el aprendiz desarrolle y perfeccione las competencias necesarias para la creación de software alineado con las necesidades del cliente, siguiendo las metodologías y buenas prácticas del sector.  Fecha máxima de entrega: 26/09/2025 | | | | | | |
| **c** | | | | | | |
| El aprendiz puede asistir con el instructor Carlos Fabian Martínez de manera remota para fortalecer sus competencias.  El aprendiz puede asistir con el instructor Javier Humberto Pinto todos los viernes de 6:00pm a 11:59pm. para fortalecer sus competencias en el ambiente ntic 209-3  El aprendiz puede asistir con el instructor Carlos Julio Cadena todos los jueves de 12:00m a 06:00pm. para fortalecer sus competencias en el ambiente ntic 209-3  El aprendiz puede asistir con el instructor Jesus Ariel Gonzales Bonilla todos los sabados de 6:00m a 12:00m. para fortalecer sus competencias en el ambiente ntic 209-3 | | | | | | |
| **ESTABLECIMIENTO Y ACEPTACIÓN DE COMPROMISOS** | | | | | | |
| **ACTIVIDAD**  **/DECISIÓN** | **FECHA** | | **RESPONSABLE** | | **FIRMA O PARTICIPACIÓN**  **VIRTUAL** | |
| **Acta** | **21/08/2025** | | **Javier Pinto** | |  | |
|  |  | |  | |  | |
|  |  | |  | |  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  | |  | |
| **DE: ASISTENTES Y APROBACIÓN DECISIONES** | | | | | | | |
| **NOMBRE** | **DEPENDENCIA/ EMPRESA** | | **APRUEBA (SI/NO)** | | **OBSERVACIÓN** | | **FIRMA O PARTICIPACIÓN**  **VIRTUAL** |
| **Javier Humberto Pinto**  **DIaz** | **CIES** | | **SI** | | **ok** | |
| **Carlos Julio Cadena** | **CIES** | | **SI** | | **ok** | |  |
| **Carlos Fabian Martinez** | **CIES** | | **SI** | | **ok** | |  |
|  |  | |  | |  | |  |
| De acuerdo con La Ley 1581 de 2012, Protección de Datos Personales, el Servicio Nacional de Aprendizaje SENA, se compromete a garantizar la seguridad y protección de los datos personales que se encuentran almacenados en este documento, y les dará el tratamiento correspondiente en cumplimiento de lo establecido legalmente. | | | | | | | |
| **ANEXOS** | | | | | | | |

Carlos