2. Evidencia obligación 1. (PDF) Planeación FPI en (4) fichas de formación asociadas:

- PROGRAMACION DE SOFTWARE: FICHA 296**1345**
    a. Guías de Aprendizaje
    b. Material de apoyo
    c. Instrumentos de Evaluación

- SISTEMAS TELEINFORMÁTICOS: FICHAS 296**1347,** 296**1395**, 296**1381**
    a. Guías de Aprendizaje
    b. Material de apoyo
    c. Instrumentos de Evaluación

PROGRAMACIÓN DE SOFTWARE

FICHA: 2961345

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- **Denominación del Programa de Formación:** Técnico en Programación
- **Código del Programa de Formación:** 233104
- **Nombre del Proyecto Formativo**: Aplicaciones multiplataforma para áreas administrativas del sector educativo, industrial y servicios de micro y pequeñas empresas de la zona de influencia del CIAS.
- **Fase del Proyecto:** Planeación
- **Actividad de Proyecto Formativo:** Elaborar propuesta de solución técnica respecto a las necesidades identificadas.
- **Competencia:** Establecer requisitos de la solución de software de acuerdo con estándares y procedimiento técnico
- **Resultados de Aprendizaje:** Interpretar el informe de requisitos de la solución conforme a la identificación de las necesidades del negocio.
- **Duración de la Guía de Aprendizaje**: 80 horas

2. PRESENTACIÓN

En el desarrollo de software, la correcta gestión de requisitos es fundamental para garantizar que los proyectos cumplan con las expectativas del cliente y los objetivos planteados. Esta guía te acompañará en el proceso de adquirir y aplicar conocimientos esenciales sobre la identificación, análisis, priorización y documentación de requisitos.

A lo largo de esta guía, te enfocarás en comprender la importancia de los estándares como el IEEE 830 y en utilizar herramientas de modelado como UML. Estas habilidades no solo te permitirán crear documentos de requisitos claros y completos, sino que también te prepararán para enfrentar los desafíos del desarrollo de software en un entorno profesional.

El enfoque práctico de las actividades te permitirá aplicar la teoría en casos reales, ayudándote a desarrollar un pensamiento crítico y una capacidad analítica que serán clave en tu carrera. Al finalizar, estarás mejor preparado para documentar y gestionar los requisitos de software, contribuyendo al éxito de los proyectos en los que participes.

"Cada línea de código comienza con un buen requisito; domina los requisitos y dominarás el desarrollo."

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividades de reflexión inicial:

Descripción de la actividad: Reflexionar sobre la importancia de los requisitos en el ciclo de vida del desarrollo del software y su impacto en la satisfacción del cliente.
Ambiente requerido: Aula de clase, computadores, escritorios.
Estrategias o técnicas didácticas activas: Debate guiado, análisis de casos de éxito y fracaso en proyectos de software.
Materiales de formación: Artículos recientes sobre tendencias en la gestión de requisitos, presentaciones sobre la importancia de los requisitos en el ciclo de vida del desarrollo de software.
Material de apoyo: Artículos sobre casos de éxito y Fracaso en la gestión de Requisitos.
Duración de la actividad: 8 horas

Contenido:

En esta primera actividad, vas a reflexionar sobre cómo los requisitos juegan un papel crucial en el desarrollo de software. Imagina que estás participando en un proyecto de software: ¿qué pasaría si los requisitos no están bien definidos desde el principio? A través de un debate guiado, compartirás tus ideas y escucharás las experiencias de tus compañeros. Además, analizaremos juntos casos reales de proyectos que han tenido éxito o han fracasado debido a la gestión de requisitos. Al final de esta actividad, entenderás por qué es vital documentar y gestionar correctamente los requisitos para satisfacer las necesidades del cliente y asegurar el éxito del proyecto.



Figura 1 Reflexión sobre la importancia de los requisitos de software en el aula

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje:

Descripción de la actividad: Introducción a los estándares de especificación de requisitos, como el IEEE 830, y su relevancia actual en la industria del software.
Ambiente requerido: Laboratorio con acceso a internet.
Estrategias o técnicas didácticas activas: Estudio de normas y plantillas ERS, investigación sobre su aplicación en proyectos reales.
Materiales de formación: Documentación técnica y ejemplos de ERS.
Material de apoyo: Guías de estándares IEEE y plantillas ERS.
Duración de la actividad: 16 horas

Contenido:
Durante esta actividad, te sumergirás en el mundo de los estándares de especificación de requisitos, comprendiendo por qué son esenciales en el desarrollo de software. Aprenderás sobre el estándar IEEE 830 y cómo se aplica en la creación de documentos de requisitos. A través de la revisión de ejemplos y la práctica con plantillas ERS, te familiarizarás con la estructura y los elementos clave que deben incluirse en un documento de requisitos efectivo. Esta actividad te permitirá reconocer la importancia de seguir un estándar para garantizar que los requisitos sean claros, completos y comprensibles para todo el equipo de desarrollo.



Figura 2 Sala de formación sobre estándares de requisitos de software.

3.3 Actividades de apropiación:

Descripción de la actividad: Elaboración de un informe de especificación de requisitos utilizando técnicas modernas de priorización y análisis.
Ambiente requerido: Laboratorio de computación.
Estrategias o técnicas didácticas activas: Taller práctico de creación de informes y uso de herramientas de modelado como UML.
Materiales de formación: Software de modelado, guías de uso de herramientas.
Material de apoyo: Recursos digitales y tutoriales de las herramientas.
Evidencias de aprendizaje: Informe de requisitos documentado.
Instrumentos de evaluación: Lista de chequeo.
Duración de la actividad: 32 horas

Contenido:

En esta actividad práctica, pondrás en acción todo lo que has aprendido hasta ahora. Utilizando técnicas modernas de análisis y priorización, trabajarás en la elaboración de un informe de especificación de requisitos para un proyecto simulado. Aprenderás a utilizar herramientas de modelado como UML para representar visualmente los requisitos y asegurarte de que están alineados con las necesidades del cliente. Al finalizar esta actividad, habrás creado un documento de requisitos detallado y bien estructurado, que servirá como guía para el desarrollo de la solución de software.



Figura 3 Elaboración de un informe de requerimientos.

3.4 Actividades de Transferencia el Conocimiento:

Descripción de la actividad: Implementación de la matriz de trazabilidad de requisitos en un caso de estudio.
Ambiente requerido: Laboratorio de computación.
Estrategias o técnicas didácticas activas: Proyecto en equipo, revisión cruzada entre pares.
Materiales de formación: Plantillas de matrices de trazabilidad, software de gestión de versiones.
Material de apoyo: Ejemplos de matrices de trazabilidad, guías de buenas prácticas.
Evidencias de aprendizaje: Matriz de trazabilidad completa y revisada.
Instrumentos de evaluación: Análisis del proyecto final con la matriz de trazabilidad.
Duración de la actividad: 24 horas.

Contenido:
En esta actividad final, aplicarás todo el conocimiento adquirido para crear una matriz de trazabilidad de requisitos. Esta herramienta es fundamental para asegurar que todos los requisitos definidos están siendo considerados en cada etapa del desarrollo. Trabajarás en equipo para aplicar esta técnica en un caso de estudio, lo que te permitirá experimentar cómo se mantiene el control sobre los requisitos a lo largo del ciclo de vida del proyecto. A través de la revisión cruzada entre pares, mejorarás tu matriz y asegurarás su precisión y utilidad. Al finalizar, tendrás una comprensión profunda de cómo la trazabilidad de requisitos ayuda a gestionar el proyecto de manera eficiente y eficaz.



4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Planeación	Elaborar propuesta de solución técnica respecto a las necesidades identificadas.	Cuestionario para evaluar comprensión del IEEE 830 y la identificación de sus elementos clave.	**Evidencia de conocimiento** Cuestionario	utiliza los estándares de especificación de requisitos de acuerdo con las características del proyecto.	**Técnica:** Prueba Escrita **Instrumento de evaluación** Cuestionario
Planeación	Elaborar propuesta de solución técnica respecto a las necesidades identificadas.	Informe de requisitos para un proyecto simulado, utilizando UML para representar los requisitos y asegurar su alineación con las necesidades del cliente.	**Evidencia de Desempeño** Informe Técnico	Documenta la especificación de requisitos según las necesidades del cliente	**Técnica:** Evaluación del informe **Instrumento de evaluación:** Lista de chequeo
Planeación	Elaborar propuesta de solución técnica respecto a las necesidades identificadas.	Desarrollo de una matriz que rastrea los requisitos a lo largo de las etapas del proyecto, demostrando un control	**Evidencia de Producto** Matriz de trazabilidad de requisitos	utiliza los estándares de especificación de requisitos de acuerdo con las características del proyecto. Documenta la especificación de requisitos	**Técnica:** Observación y revisión **Instrumento de evaluación:** Lista de chequeo

		efectivo del proceso.		según las necesidades del cliente	

5. GLOSARIO DE TÉRMINOS

Análisis de Requisitos: Proceso de definir, documentar y gestionar las necesidades y expectativas de los usuarios finales y otras partes interesadas para asegurar que el software cumple con sus objetivos.

Casos de Uso: Descripciones detalladas de cómo los usuarios interactuarán con el sistema para cumplir un objetivo particular, frecuentemente utilizados en la etapa de análisis de requisitos.

Ciclo de Vida del Desarrollo del Software: Conjunto de etapas por las cuales pasa un proyecto de software desde su concepción hasta su implementación y mantenimiento. Estas etapas incluyen la planificación, análisis, diseño, implementación, prueba, despliegue y mantenimiento.

Control de Versiones: Sistema que gestiona y registra los cambios en los documentos, código fuente, y otros elementos de un proyecto de software a lo largo del tiempo. Herramientas como Git son comunes en la gestión de versiones.

Diagrama de Casos de Uso: Gráfico UML que muestra las interacciones entre los usuarios (actores) y el sistema, indicando las funciones principales que el sistema debe cumplir.

Documento de Especificación de Requisitos: Informe que detalla los requisitos funcionales y no funcionales del software, incluyendo las expectativas del cliente, las restricciones técnicas, y cualquier otra información relevante para guiar el desarrollo del sistema.

Especificación de Requisitos del Software (ERS): Documento que describe las funciones, características y restricciones del software a desarrollar, asegurando que se cumplan las expectativas del cliente y los usuarios finales.

Historia de Usuario: Breve descripción de una funcionalidad deseada desde la perspectiva del usuario, utilizada para capturar las necesidades de manera sencilla y manejable, especialmente en metodologías ágiles.

IEEE 830: Estándar de la IEEE para la especificación de requisitos de software, que proporciona una guía sobre cómo documentar los requisitos de manera clara, completa y comprensible.

Matriz de Trazabilidad de Requisitos: Herramienta que relaciona los requisitos iniciales del proyecto con las etapas subsecuentes del desarrollo, asegurando que todos los requisitos están siendo cumplidos y que no se omite ninguna necesidad del cliente.

Prioridad de Requisitos: Técnica utilizada para determinar el orden de importancia y la secuencia en que los requisitos deben ser implementados en un proyecto de software, basada en factores como valor para el negocio, costo, riesgo, y urgencia.

Software Requirements Specification (SRS): Documento que captura todos los requisitos del software, incluyendo los funcionales, no funcionales, y de interfaz, basado en un estándar como el IEEE 830.

Técnicas de Modelado: Métodos utilizados para crear representaciones abstractas de sistemas de software, a menudo a través de diagramas UML, para visualizar la estructura y comportamiento del software.

Trazabilidad: Capacidad de relacionar los elementos de un sistema (como requisitos, código, pruebas) entre sí, de modo que se pueda rastrear cada requisito desde su origen hasta su implementación y validación final.

UML (Unified Modeling Language): Lenguaje de modelado unificado utilizado para especificar, visualizar, construir y documentar los artefactos de un sistema de software. UML es ampliamente utilizado en la creación de diagramas que representan la estructura y el comportamiento de un sistema.

6. REFERENTES BILBIOGRÁFICOS

IEEE Computer Society. (1998). *IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998).* IEEE.

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *The Unified Modeling Language User Guide*. Addison-Wesley.

Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques*. Springer.

Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach*. McGraw-Hill.

Sommerville, I. (2011). *Ingeniería del Software*. Addison-Wesley.

van Lamsweerde, A. (2009). *Requirements Engineering: From System Goals to UML Models to Software Specifications*. John Wiley & Sons.

Wiegers, K., & Beatty, J. (2013). *Software Requirements*. Microsoft Press.

Young, R. R. (2004). *Effective Requirements Practices*. Addison-Wesley.

SENA. (2023). *Guía práctica para la utilización de UML en proyectos de software.* Biblioteca Virtual SENA. https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1j5choe/sena_biblioteca_eniRIT52.5UML

SENA. (2023). *Estudio sobre UML y su aplicación en el desarrollo de software.* Biblioteca Virtual SENA. https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1i756fj/TN_cdi_dialnet_primary_oai_dialnet_unirioja_es_ART0000700241

SENA. (2023). *Guía para la gestión de requisitos en el desarrollo de software.* Biblioteca Virtual SENA. https://sena-primo.hosted.exlibrisgroup.com/permalink/f/1j5choe/sena_biblioteca_eniEPT3GIT

7. CONTROL DEL DOCUMENTO

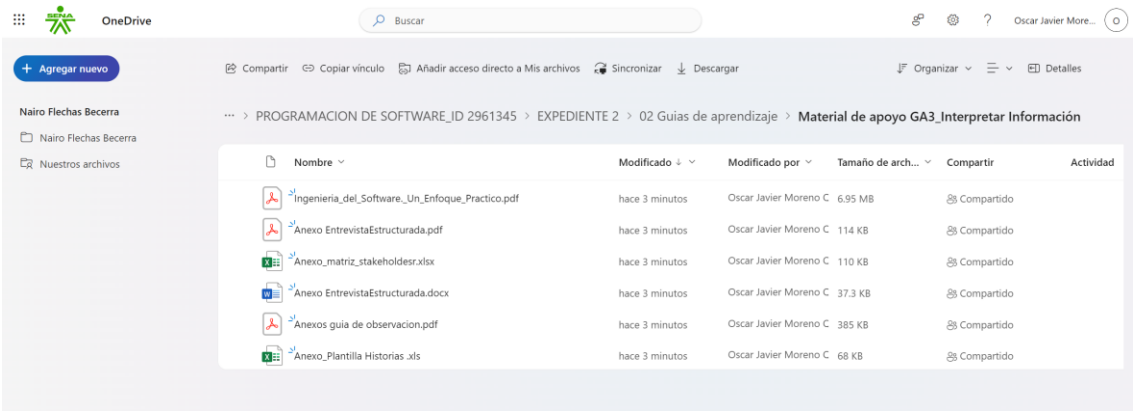
	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Oscar Javier Moreno Quiceno	Instructor	Programación de software	Agosto 2024

8. CONTROL DE CAMBIOS

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

MATERIAL DE APOYO PROGRAMACIÓN DE SOFTWARE

2961345



INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en Programación de Software	**ID**	2961345
Proyecto formativo:	Aplicaciones multiplataforma para áreas administrativas del sector educativo, industrial y servicios de micro y pequeñas empresas de la zona de influencia del CIAS		
Fase proyecto:	Planeación		
Resultado de aprendizaje:	**RAP 2.** Interpretar el informe de requisitos de la solución conforme a la identificación de las necesidades del negocio.		
Actividad de aprendizaje:	Introducción a los estándares de especificación de requisitos, como el IEEE 830, y su relevancia actual en la industria del software.		
Evidencia de Conocimiento	Cuestionario		

Criterios de evaluación:
Utiliza los estándares de especificación de requisitos de acuerdo con las características del proyecto.

2. Cuestionario

1. **¿Qué es el IEEE 830?**
   (Explique brevemente su propósito.)
2. **¿Por qué es importante seguir el estándar IEEE 830 al documentar los requisitos de software?**
   (Describa su relevancia de manera sencilla.)
3. **¿Qué debe incluir la sección de "Descripción General" en un documento de requisitos de software según el IEEE 830?**
   (Explique brevemente lo que se espera en esta sección.)
4. **¿Qué son los requisitos funcionales según el IEEE 830?**
   (Describa qué son y por qué son importantes.)
5. **¿Qué son los requisitos no funcionales según el IEEE 830?**
   (Explique su función en el documento de requisitos.)
6. **¿Qué se incluye en la sección de "Interfaces externas" en un documento conforme al IEEE 830?**
   (Describa brevemente qué se espera en esta parte del documento.)
7. **¿Qué información se debe incluir en la sección de "Referencias" de un documento según el IEEE 830?**
   (Mencione qué tipo de información se encuentra en esta sección.)
8. **¿Qué se describe en la sección de "Atributos del sistema" según el IEEE 830?**
   (Explique de manera simple qué se incluye en esta sección.)
9. **¿Cómo ayuda el estándar IEEE 830 a mejorar la calidad del software?**
   (Describa una o dos formas en que el estándar contribuye.)
10. **¿Cuál es la diferencia entre requisitos funcionales y no funcionales según el IEEE 830?**
    (Explique la principal diferencia de manera sencilla.)

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. Firmas

Firma Aprendiz: _____

Firma Instructor: _____

INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en Programación de Software	**ID**	2961345
Proyecto formativo:	Aplicaciones multiplataforma para áreas administrativas del sector educativo, industrial y servicios de micro y pequeñas empresas de la zona de influencia del CIAS		
Fase proyecto:	Planeación		
Resultado de aprendizaje:	**RAP 2.** Interpretar el informe de requisitos de la solución conforme a la identificación de las necesidades del negocio.		
Actividad de aprendizaje:	Elaboración de un informe de especificación de requisitos utilizando técnicas modernas de priorización y análisis.		
Evidencia de Desempeño	Informe de requisitos documentado		

Criterios de evaluación:
Documenta la especificación de requisitos según las necesidades del cliente.

2. Lista de chequeo

No.	VARIABLES/INDICADORES DE LOGRO	CUMPLE		Observaciones
		SI	NO	
1.	**El aprendiz identifica y prioriza** correctamente los requisitos de acuerdo con las necesidades del cliente durante la actividad.			
2.	**El aprendiz aplica** adecuadamente técnicas de análisis para la identificación y categorización de requisitos funcionales y no funcionales.			
3.	**El aprendiz utiliza** de manera efectiva herramientas de modelado, como UML, para representar gráficamente los requisitos durante la sesión práctica.			
4.	**El aprendiz organiza y documenta** la información de los requisitos de manera clara y coherente, siguiendo los estándares establecidos.			
S.	**El aprendiz entrega** conclusiones y recomendaciones basadas en el análisis y priorización de requisitos realizadas durante la actividad.			

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. Firmas

Firma Aprendiz: _____

Firma Instructor: _____

INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en Programación de Software	**ID**	2961345
Proyecto formativo:	Aplicaciones multiplataforma para áreas administrativas del sector educativo, industrial y servicios de micro y pequeñas empresas de la zona de influencia del CIAS		
Fase proyecto:	Planeación		
Resultado de aprendizaje:	**RAP 2.** Interpretar el informe de requisitos de la solución conforme a la identificación de las necesidades del negocio.		
Actividad de aprendizaje:	Implementación de la matriz de trazabilidad de requisitos en un caso de estudio		
Evidencia de Producto	Matriz de trazabilidad		

Criterios de evaluación:

- ✓ Utiliza los estándares de especificación de requisitos de acuerdo con las características del proyecto.
- ✓ Documenta la especificación de requisitos según las necesidades del cliente.

2. Lista de chequeo

No.	VARIABLES/INDICADORES DE LOGRO	CUMPLE		Observaciones
		SI	NO	
1.	**En el producto se evidencia que se identifican** correctamente los requisitos desde su origen hasta su implementación en el sistema.			
2.	**En el producto se evidencia que se relacionan** adecuadamente los requisitos con los diferentes artefactos y etapas del proyecto.			
3.	**En el producto se evidencia que se documenta** de manera clara y coherente el seguimiento de los requisitos a lo largo del ciclo de vida del proyecto.			
4.	**En el producto se evidencia que se utilizan** herramientas y plantillas adecuadas para la gestión de la trazabilidad de los requisitos.			
S.	**En el producto se evidencia que se revisan** y validan los enlaces y relaciones de trazabilidad, asegurando la integridad de la información.			

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. FIRMAS

Firma Aprendiz: _____

Firma Instructor: _____

SISTEMAS TELEINFORMÁTICOS

FICHAS: 2961347-2961395

2961381

PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. Identificación de la Guía de Aprendizaje

- **Denominación del Programa de Formación:** Técnico en Programación
- **Código del Programa de Formación:** 233108 - 1
- **Nombre del Proyecto Formativo:** Fortalecimiento de los servicios Teleinformáticos de equipos y redes en las MiPymes de la región de Boyacá.
- **Fase del Proyecto:** Evaluación.
- **Actividad de Proyecto Formativo:** Validar los procedimientos realizados en los sistemas Teleinformáticos.
- **Competencia:** Mantener equipos de cómputo según procedimiento técnico
- **Resultados de Aprendizaje:** RA3.Verificar la operación del sistema computacional de acuerdo con los procedimientos técnicos
- **Duración de la Guía de Aprendizaje:** 120 horas

2. Presentación

Esta guía de aprendizaje está diseñada para guiar a los Aprendices en el proceso de verificación de la operación de sistemas computacionales, enfocándose en la aplicación de sistemas operativos y software utilitario. A lo largo del curso, los Aprendices desarrollarán competencias críticas en la identificación y corrección de fallas, la ejecución de acciones correctivas, y la verificación del funcionamiento de equipos según manuales de operación y normativas técnicas.



Ilustración 1 Representación aula de formación

"El éxito no es el final, el fracaso no es fatal: lo que cuenta es el valor para continuar."

3. Formulación de las Actividades de Aprendizaje

3.1 Actividades de Reflexión Inicial

Actividad 3.1.1: Reflexión sobre Videos

- **Descripción:** Los Aprendices realizarán una línea de tiempo con las fechas más importantes de la historia de "Como se creó Microsoft". Después, escribirán una reflexión de media página sobre cómo los sistemas operativos influyen en la capacidad de verificar y mantener el correcto funcionamiento de los sistemas computacionales.
- **Duración:** 16 horas
- **Ambiente requerido:** Aula de clase o entorno virtual con acceso a internet.
- **Estrategias o técnicas didácticas:** Análisis y reflexión individual, discusión en grupo.
- **Materiales de formación:** Acceso a los videos, guías de reflexión.
- **Evidencia de Conocimiento:** Línea de tiempo.
    - **Instrumento de Evaluación:** Lista de chequeo.

Desarrollo de actividad:

Primero, verás dos videos que te proporcionarán una visión general de los sistemas operativos y su impacto en la tecnología:

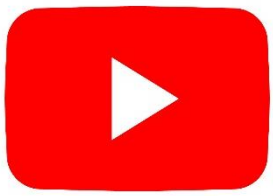


Ilustración 2 Ícono de YouTube que indica enlace a contenido en video. © YouTube.

https://www.youtube.com/watch?v=CIi_PMTUwkE

Los sistemas operativos ¿Qué son?

https://www.youtube.com/watch?v=666Pc5Yp5hM

¿Así funciona tu Sistema Operativo?

https://www.youtube.com/watch?v=3G6iMH07k3U

Cómo se creó Microsoft: La Historia de Bill Gates y Paul Allen

Recomendaciones SST

Evita reflejos en las pantallas.
Utilice Teclados ergonómicos
Mantén una distancia prudente a la pantalla del computador

3.1.1 Realizar una línea de tiempo de la Historia de Microsoft. Teniendo en cuenta el trabajo que ha venido realizando Microsoft a medida que la tecnología va avanzando.
3.1.2 ¿Usted considera ético que las personas utilicen activadores de Windows con el fin de no comprar la licencia?
3.1.3 Nombre otros S.O. que no sean del fabricante MICROSOFT.
3.1.4 ¿cómo los sistemas operativos influyen en la capacidad de verificar y mantener el correcto funcionamiento de los sistemas computacionales

3.2 Actividades de Contextualización e Identificación de Conocimientos Necesarios para el Aprendizaje

Actividad 3.2.1: Teoría de Sistemas Operativos

- **Descripción:** Clase teórica sobre sistemas operativos, su estructura, y cómo afectan la operación de los sistemas computacionales.
- **Duración:** 16 horas
- **Ambiente requerido:** Aula de clase con pizarra y recursos audiovisuales.
- **Estrategias o técnicas didácticas:** Clase magistral, discusión en grupo.
- **Materiales de formación:** Presentaciones, apuntes de clase, manuales técnicos.
- **Evidencia de Conocimiento:** Cuestionario.
    - **Instrumento de Evaluación:** Cuestionario.

Desarrollo de la actividad:

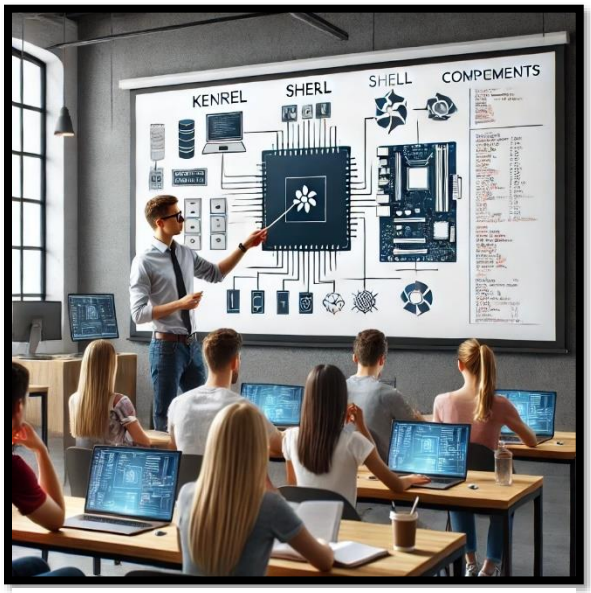


Ilustración 3 Clase Magistral

Esta actividad se centrará en una clase teórica exhaustiva sobre los sistemas operativos, enfocándose en su estructura interna y en cómo afectan la operación y el mantenimiento de los sistemas computacionales. La clase incluirá explicaciones sobre los diferentes tipos de sistemas operativos, su arquitectura, y su impacto en la gestión de recursos, la seguridad, y la eficiencia del sistema



Actividad 3.2.2: Instalación y Configuración de Sistemas Operativos

- **Descripción:** Los Aprendices instalarán y configurarán sistemas operativos en un entorno simulado, enfocándose en cómo estas acciones preparan el sistema para su verificación.
- **Duración:** 8 horas
- **Ambiente requerido:** Salón de informática con máquinas virtuales o equipos de prueba.
- **Estrategias o técnicas didácticas:** Taller práctico, trabajo en equipos.
- **Materiales de formación:** Manuales de instalación, software de sistemas operativos, herramientas de configuración.
- **Evidencia de Desempeño:** Video Instalación.
    - **Instrumento de Evaluación:** Lista de Chequeo.

¡¡Vamos a instalar Un sistema Operativo!!

Para el desarrollo de esta actividad instalaremos el sistema operativo Ubuntu; en el siguiente enlace encontrarás una Guía de como realizar el proceso:

YouTube

Ilustración 4 Ícono de YouTube que indica enlace a contenido en video. © YouTube.

https://www.youtube.com/watch?v=8MRibUo9VAA

Cómo instalar UBUNTU 22.04 PASO a PASO desde cero! TUTORIAL

Recomendaciones SST
Evita reflejos en las pantallas.
Utilice Teclados ergonómicos
Mantén una distancia prudente a la pantalla del computador

La actividad se realizará en clase bajo la supervisión del instructor. El proceso de edición del video continuará fuera del aula como parte del trabajo extracurricular. La actividad se llevará a cabo en parejas.

Actividad 3.2.3: Uso de Software Utilitario para Verificación

- **Descripción:** Los Aprendices aprenderán a utilizar software utilitario (como herramientas de diagnóstico y mantenimiento) para verificar la operación del sistema operativo.
- **Duración:** 16 horas
- **Ambiente requerido:** Salón de informática con acceso a computadores.
- **Estrategias o técnicas didácticas:** Taller práctico, demostración guiada.
- **Materiales de formación:** Software utilitario, guías de usuario.
- **Evidencia de Conocimiento:** Documento paso a paso para la instalación del software utilitario CCleaner, incluyendo imágenes de cada paso.
    - **Instrumento de Evaluación:** Lista de chequeo.

¡Bien Ahora, Vamos a verificar el funcionamiento del Sistema Operativo!

Primero que todo identificaremos que sistema Operativo hemos instalado; Si hemos instalado el Sistema operativo **Windows**; utilizaremos el Programa **Ccleaner**

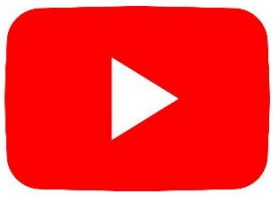


Ilustración 5 Ícono de YouTube que indica enlace a contenido en video. © YouTube.

https://www.youtube.com/watch?v=heD5hjP3Sa8

Mantenimiento Preventivo Software

Recomendaciones SST

Evita reflejos en las pantallas.
Utilice Teclados ergonómicos
Mantén una distancia prudente a la pantalla del computador

Ahora, si el Sistema Operativo Instalado es **Ubuntu** Procederemos a mirar el Material de apoyo "**Paso a paso Herramientas Utilitarias Ubuntu.pdf**" que tenemos para seguir el Paso a Paso para Instalar: **Stacer** o **BleachBit**



Ilustración 7 Interfaz de Stacer. (Labarta, 2021). https://labarta.es/wp-content/uploads/2021/02/stacer.png



Ilustración 6 Logo de BleachBit. (Wikimedia Commons, 2023). https://upload.wikimedia.org/wikipedia/commons/a/a9/Bleachbit_logo.svg

3.3 Actividades de Apropiación

Actividad 3.3.1: Implementación de Acciones Correctivas

- **Descripción:** Los Aprendices practicarán la identificación y corrección de fallas en sistemas operativos, siguiendo procedimientos técnicos establecidos.
- **Duración:** 16 horas
- **Ambiente requerido:** Salón de informática con acceso a computadores con sistemas operativos instalados.
- **Estrategias o técnicas didácticas:** Simulación de fallas, resolución de problemas en tiempo real.
- **Materiales de formación:** Manuales de diagnóstico, herramientas de mantenimiento.
- **Evidencia de Desempeño:** Informe práctico.
    - **Instrumento de Evaluación:** Rúbrica para la evaluación del informe técnico.

Desarrollo de la actividad.

Vamos a revisar las fallas encontradas en nuestro sistema Operativo, Si no tenemos, ¡**las simularemos!**

Primero realizaremos una investigación sobre las posibles fallas que pueden surgir en el sistema operativo durante y después de su instalación.



Ilustración 8 Ícono de Windows 10. (Microsoft, 2023).

Microsoft ofrece una guía completa sobre cómo solucionar problemas comunes en **Windows**, que incluye detalles sobre fallas de arranque, errores de pantalla azul, problemas con controladores, y más. Enlace.

Canonical proporciona documentación sobre cómo diagnosticar y corregir fallas en sistemas **Ubuntu**, que cubre desde problemas de arranque hasta errores en la instalación de paquetes. Enlace.



Ilustración 9Ícono de Ubuntu. (Canonical, s.f.). https://tutoriales.cect.org/comandos-utiles-para-ubuntu/

Bueno después de tener el conocimiento de la instalación y el uso de las herramientas, vamos a documentar todo el proceso y para ello lo entregaremos en un informe técnico, par

Actividad 3.3.2: Verificación del Sistema Operativo Post-Corrección

- **Descripción:** Después de corregir fallas, los Aprendices verificarán el funcionamiento del sistema operativo utilizando el manual de operación.
- **Duración:** 16 horas
- **Ambiente requerido:** Laboratorio de informática.
- **Estrategias o técnicas didácticas:** Taller práctico, trabajo en equipo.
- **Materiales de formación:** Manuales de operación, software de verificación.
- **Evidencia de Desempeño:** Informe sobre verificación post-corrección.
    - **Instrumento de Evaluación:** Lista de chequeo del proceso de verificación.

Actividad 3.3.3: Uso de Software Utilitario para Optimización

- **Descripción:** Los Aprendices usarán software utilitario para realizar un mantenimiento preventivo y verificar que el sistema esté funcionando correctamente.
- **Duración:** 8 horas
- **Ambiente requerido:** Laboratorio de informática.
- **Estrategias o técnicas didácticas:** Simulación de mantenimiento preventivo.
- **Materiales de formación:** Software de optimización, guías de mantenimiento.
- **Evidencia de Desempeño:** Informe sobre uso de software utilitario.
    - **Instrumento de Evaluación:** Rúbrica de evaluación del mantenimiento realizado.

3.4 Actividades de Transferencia del Conocimiento

Actividad 1: Proyecto Final de Verificación del Sistema Computacional

- **Descripción:** Los Aprendices implementarán un proyecto donde deberán verificar la operación completa de un sistema computacional, incluyendo el sistema operativo, siguiendo los procedimientos técnicos y utilizando software utilitario.
- **Duración:** 16 horas
- **Ambiente requerido:** Laboratorio de informática.
- **Estrategias o técnicas didácticas:** Proyecto en equipo, evaluación final.
- **Materiales de formación:** Sistemas computacionales, manuales técnicos, software utilitario.
- **Evidencia de Producto:** Proyecto final.
    - **Instrumento de Evaluación:** Rúbrica de evaluación del proyecto.

Actividad 2: Presentación del Proyecto y Retroalimentación

- **Descripción:** Los Aprendices presentarán su proyecto final, explicando las acciones correctivas implementadas, el uso del software utilitario, y la verificación final del funcionamiento.
- **Duración:** 8 horas
- **Ambiente requerido:** Aula de clase o sala de presentaciones.
- **Estrategias o técnicas didácticas:** Presentación oral, discusión en grupo.
- **Materiales de formación:** Proyector, diapositivas, materiales de apoyo.
- **Evidencia de Producto:** Presentación del proyecto final.
    - **Instrumento de Evaluación:** Lista de verificación para la presentación.

4. PLANTEAMIENTO DE EVIDENCIAS DE APRENDIZAJE PARA LA EVALUACIÓN EN EL PROCESO FORMATIVO.

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
EVALUACIÓN	validar los procedimientos realizados en los sistemas Teleinformáticos.	Línea de tiempo con las fechas más importantes de la historia de “Como se creó Microsoft”.	Evidencia de **Conocimiento** Línea de Tiempo	Verifica el funcionamiento del equipo, según el manual de operación.	Técnica: Observación Directa Instrumento de Evaluación: Lista de Chequeo
EVALUACIÓN	validar los procedimientos realizados en los sistemas Teleinformáticos.	Cuestionario sobre sistemas operativos, su estructura, y cómo afectan la operación de los sistemas	Evidencia de **Conocimiento** Cuestionario	Verifica el funcionamiento del equipo, según el manual de operación.	Técnica: Prueba escrita Instrumento de Evaluación: Cuestionario

		computacionales			
EVALUACIÓN	validar los procedimientos realizados en los sistemas Teleinformáticos.	Video Instalación Sistema Operativo	Evidencia de **Desempeño** Video	Verifica el funcionamiento del equipo, según el manual de operación.	Técnica: Observación Directa Instrumento de Evaluación: Lista de chequeo
EVALUACIÓN	validar los procedimientos realizados en los sistemas Teleinformáticos.	Documento paso a paso para la instalación del software utilitario CCleaner, incluyendo imágenes de cada paso	Evidencia de **Conocimiento** Documento técnico	Ejecuta software utilitario según requerimientos del cliente y normatividad	Técnica: Observación Directa Instrumento de Evaluación: Lista de chequeo
EVALUACIÓN	validar los procedimientos realizados en los sistemas Teleinformáticos.	Proyecto final, explicando las acciones correctivas implementadas, el uso del software utilitario, y la verificación final del funcionamiento.	Evidencia de **Producto** Documento Proyecto	Implementa acciones correctivas según las fallas y procedimientos técnicos establecidos Ejecuta software utilitario según requerimientos del cliente y normatividad Verifica el funcionamiento del equipo, según el	Técnica: Observación Directa Instrumento de Evaluación: Rúbrica

				manual de operación.	

5. GLOSARIO DE TÉRMINOS

Sistema Operativo: Software fundamental que gestiona los recursos de hardware y permite la interacción entre el usuario y la computadora.

Kernel: Núcleo del sistema operativo que controla las operaciones básicas del hardware y gestiona la memoria y los procesos.

Interfaz de Usuario (UI): Parte del sistema operativo que permite a los usuarios interactuar con el sistema, ya sea a través de gráficos (GUI) o de texto (CLI).

Licenciamiento de Software: Acuerdo legal que especifica cómo y en qué condiciones se puede utilizar un software.

Software Utilitario: Programas diseñados para realizar tareas de mantenimiento, optimización y diagnóstico en un sistema operativo.

Administrador de Tareas: Herramienta del sistema operativo que permite ver y gestionar los procesos en ejecución y el rendimiento del sistema.

Sistema de Archivos: Estructura que utiliza el sistema operativo para organizar y almacenar archivos en un dispositivo de almacenamiento.

Partición: División de un disco duro en secciones independientes, cada una con su propio sistema de archivos.

Backup: Copia de seguridad de datos que se guarda para evitar la pérdida de información en caso de fallo del sistema.

Herramienta de Diagnóstico: Software utilizado para analizar el funcionamiento del hardware y software de un sistema y detectar posibles fallos.

Driver: Software que permite que el sistema operativo controle y se comunique con el hardware.

Gestión de Procesos: Conjunto de funciones del sistema operativo que administra la ejecución de procesos y tareas.

Actualización del Sistema: Proceso de instalar mejoras y parches en el sistema operativo para corregir errores y mejorar su seguridad y rendimiento.

Restauración del Sistema: Función que permite revertir el sistema operativo a un estado anterior para corregir problemas recientes.

Fragmentación: Situación en la que los archivos en un disco están divididos en partes dispersas, lo que reduce la eficiencia del acceso a los datos.

Defragmentación: Proceso de reorganización de los datos en un disco duro para que los archivos estén almacenados de manera contigua, mejorando así el rendimiento del sistema.

Controlador de Dominio: Servidor que gestiona la seguridad y los permisos de acceso en una red de computadoras, generalmente utilizado en entornos Windows.

Virtualización: Técnica que permite ejecutar múltiples sistemas operativos en una sola máquina física, utilizando software especializado como máquinas virtuales.

Interfaz de Línea de Comandos (CLI): Método de interactuar con el sistema operativo mediante comandos de texto, en lugar de una interfaz gráfica.

Gestión de Memoria: Función del sistema operativo que controla cómo se asigna y utiliza la memoria en el sistema.

6. REFERENTES BILBIOGRÁFICOS

Raya Cabrera, José Luis., and Laura Raya González. *Instalación y Configuración de Sistemas Operativos*. 2014. Print.

Bellido Quintero, E., & Pérez Huguet, R. (2023). *Instalación y actualización de sistemas operativos. IFCT0309*. Certificado de Profesionalidad; IFCT0309-Montaje y Reparación de Sistemas Microinformáticos. Elibro. **https://sena-primo.hosted.exlibrisgroup.com/permalink/f/q6j6k0/sena_elibroELB232695**

Elementos esenciales de TI. (2018, enero 22). Academia de Redes. https://www.netacad.com/courses/os-it/it-essentials

Tanenbaum, A. S., & Bos, H. (2015). *Modern Operating Systems* (4th ed.). Pearson Education.

Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts* (10th ed.). Wiley.

Stallings, W. (2018). *Operating Systems: Internals and Design Principles* (9th ed.). Pearson.

Love, R. (2010). *Linux Kernel Development* (3rd ed.). Addison-Wesley Professional.

Nemeth, E., Snyder, G., Hein, T. R., & Whaley, B. (2017). *UNIX and Linux System Administration Handbook* (5th ed.). Prentice Hall.

Gookin, D. (2021). *Windows 10 for Dummies* (3rd ed.). John Wiley & Sons.

Pogue, D. (2020). *Windows 10: The Missing Manual* (2nd ed.). O'Reilly Media.

Bello Quintero, E., & Pérez Huguet, R. (2023). *Instalación y actualización de sistemas operativos. IFCT0309*. Certificado de Profesionalidad; IFCT0309-Montaje y Reparación de Sistemas Microinformáticos. Elibro.

Microsoft. (2023). *Windows Sysinternals*. Recuperado de https://docs.microsoft.com/en-us/sysinternals/

Valdés, M., & Aguilera, J. (2021). *Herramientas utilitarias para la gestión y mantenimiento de sistemas operativos*. Editorial Universitaria.

7. CONTROL DEL DOCUMENTO

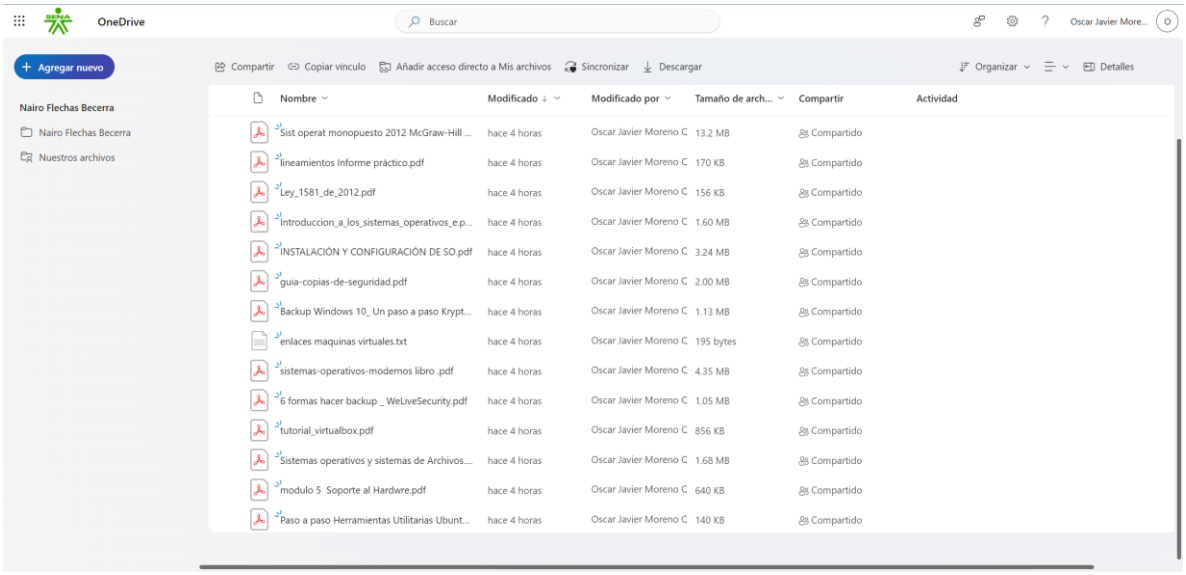
	Nombre	Cargo	Dependencia	Fecha
Autor (es)	**OSCAR JAVIER MORENO QUICENO**	**INSTRUCTOR**	**SISTEMAS TELEINFORMATICOS**	**AGOSTO 2024**

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio

MATERIAL DE APOYO SISTEMAS TELEINFORMÁTICOS

FICHAS 2961347 – 2961381 – 2961395



INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en sistemas Teleinformáticos.	**ID**	2961347 2961381 2961395
Proyecto formativo:	Fortalecimiento de los servicios Teleinformáticos de equipos y redes en las MiPymes de la región.		
Fase proyecto:	Evaluación.		
Resultado de aprendizaje:	**RAP 03** Verificar la operación del sistema computacional de acuerdo con los procedimientos técnicos.		
Actividad de aprendizaje:	Línea de tiempo con las fechas más importantes de la historia de "Como se creó Microsoft".		
Evidencia de Conocimiento	Línea de Tiempo		
Criterios de evaluación: Verifica el funcionamiento del equipo, según el manual de operación.			

2. Lista de chequeo

No.	VARIABLES/INDICADORES DE LOGRO	CUMPLE		Observaciones
		SI	NO	
1.	El aprendiz **identifica** los eventos más relevantes en la historia de Microsoft.			
2.	El aprendiz **describe** con claridad las fechas y los detalles de cada evento.			
3.	El aprendiz **utiliza** adecuadamente el cuaderno para representar la línea de tiempo.			
4.	El aprendiz **analiza** la influencia de Microsoft en la verificación de sistemas computacionales y lo incluye en la línea de tiempo.			
S.	El aprendiz **presenta** la línea de tiempo con un diseño atractivo y buena organización.			

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. Firmas

Firma Aprendiz: _____

Firma Instructor: _____

INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en sistemas Teleinformáticos.	**ID**	2961347 2961381 2961395
Proyecto formativo:	Fortalecimiento de los servicios Teleinformáticos de equipos y redes en las MiPymes de la región.		
Fase proyecto:	Evaluación.		
Resultado de aprendizaje:	**RAP 03** Verificar la operación del sistema computacional de acuerdo con los procedimientos técnicos.		
Actividad de aprendizaje:	Cuestionario sobre sistemas operativos, su estructura, y cómo afectan la operación de los sistemas computacionales		
Evidencia de Conocimiento	Cuestionario		
Criterios de evaluación: Verifica el funcionamiento del equipo, según el manual de operación.			

2. Cuestionario

1. **¿Qué es un sistema operativo?**
   (Explique de manera sencilla su función principal.)
2. **Mencione tres componentes básicos de un sistema operativo.**
   (Describa brevemente la función de cada uno.)
3. **¿Cuál es el rol del núcleo (Kernel) en un sistema operativo?**
   (Explique su importancia de forma simple.)
4. **¿Qué es un proceso en un sistema operativo?**
   (Defina qué es y por qué es importante.)
5. **¿Qué tipo de sistemas operativos se utilizan comúnmente en computadoras?**
   (Mencione y describa dos tipos.)
6. **¿Para qué sirve un sistema de archivos en un sistema operativo?**
   (Explique su función principal.)
7. **¿Qué es la planificación de procesos en un sistema operativo?**
   (Describa brevemente cómo funciona.)
8. **¿Cuál es la diferencia entre un sistema operativo de red y uno de tiempo real?**
   (Explique la principal diferencia.)
9. **¿Por qué es importante el licenciamiento en los sistemas operativos?**
   (Describa su relevancia de manera sencilla.)
10. **¿Cómo ayuda un sistema operativo a mantener el correcto funcionamiento de un equipo?**
    (Mencione una o dos maneras en que lo hace.)

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. Firmas

Firma Aprendiz: _____

Firma Instructor: _____

INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en sistemas Teleinformáticos.	**ID**	2961347 2961381 2961395
Proyecto formativo:	Fortalecimiento de los servicios Teleinformáticos de equipos y redes en las MiPymes de la región.		
Fase proyecto:	Evaluación.		
Resultado de aprendizaje:	**RAP 03** Verificar la operación del sistema computacional de acuerdo con los procedimientos técnicos.		
Actividad de aprendizaje:	Video Instalación Sistema Operativo		
Evidencia de Desempeño	Video		
Criterios de evaluación: Verifica el funcionamiento del equipo, según el manual de operación.			

2. Lista de chequeo

No.	VARIABLES/INDICADORES DE LOGRO	CUMPLE		Observaciones
		SI	NO	
1.	El aprendiz **desarrolla** el proceso de instalación del sistema operativo correctamente, siguiendo las instrucciones técnicas.			
2.	El aprendiz **configura** las opciones del sistema operativo para asegurar su funcionamiento óptimo.			
3.	El aprendiz **verifica** que todos los componentes del sistema operativo estén funcionando de acuerdo con el manual de operación.			
4.	El aprendiz **ajusta** las configuraciones del sistema operativo para mejorar su rendimiento y estabilidad			
S.	El aprendiz **documenta** cada paso del proceso de instalación y configuración, incluyendo capturas de pantalla o registros de actividad.			

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. Firmas

Firma Aprendiz: _____

Firma Instructor: _____

INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en sistemas Teleinformáticos.	**ID**	2961347 2961381 2961395
Proyecto formativo:	Fortalecimiento de los servicios Teleinformáticos de equipos y redes en las MiPymes de la región.		
Fase proyecto:	Evaluación.		
Resultado de aprendizaje:	**RAP 03** Verificar la operación del sistema computacional de acuerdo con los procedimientos técnicos.		
Actividad de aprendizaje:	Documento paso a paso para la instalación del software utilitario CCleaner, incluyendo imágenes de cada paso		
Evidencia de Conocimiento	Documento técnico		
Criterios de evaluación: Ejecuta software utilitario según requerimientos del cliente y normatividad.			

2. Lista de chequeo

No.	**VARIABLES/INDICADORES DE LOGRO**	**CUMPLE**		**Observaciones**
		SI	**NO**	
1.	El aprendiz **identifica** correctamente cada paso del proceso de instalación			
2.	El aprendiz **describe** con claridad cada paso, utilizando un lenguaje técnico apropiado.			
3.	El aprendiz **incluye** imágenes relevantes que ilustran cada paso del proceso.			
4.	El aprendiz **organiza** el documento de manera lógica y secuencial.			
S.	El aprendiz **explica** posibles errores o advertencias durante la instalación y cómo solucionarlos.			

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. Firmas

Firma Aprendiz: _____

Firma Instructor: _____

INSTRUMENTO DE EVALUACIÓN

1. DATOS GENERALES

Programa de formación:	Técnico en sistemas Teleinformáticos.	**ID**	2961347 2961381 2961395
Proyecto formativo:	Fortalecimiento de los servicios Teleinformáticos de equipos y redes en las MiPymes de la región.		
Fase proyecto:	Evaluación.		
Resultado de aprendizaje:	**RAP 03** Verificar la operación del sistema computacional de acuerdo con los procedimientos técnicos.		
Actividad de aprendizaje:	Proyecto final, explicando las acciones correctivas implementadas, el uso del software utilitario, y la verificación final del funcionamiento.		
Evidencia de Producto	Documento en PDF Proyecto		

Criterios de evaluación:

- ✓ Implementa acciones correctivas según las fallas y procedimientos técnicos establecidos
- ✓ Ejecuta software utilitario según requerimientos del cliente y normatividad
- ✓ Verifica el funcionamiento del equipo, según el manual de operación.

2. Rubrica de evaluación del Proyecto.

Criterio de Evaluación	Excelente (4 puntos)	Bueno (3 puntos)	Satisfactorio (2 puntos)	Necesita Mejorar (1 punto)	Puntaje Obtenido
1. Verificación Completa del Sistema Computacional	En el producto se evidencia que se verifica de manera exhaustiva y completa la operación de todos los componentes de hardware y software, sin omisiones.	En el producto se evidencia que se verifica la mayoría de los componentes, pero se omiten algunos detalles menores.	En el producto se evidencia que se verifica solo parcialmente el sistema, dejando algunos componentes importantes sin revisar.	En el producto se evidencia una verificación insuficiente o incompleta, con muchos componentes omitidos.	**/4**
2. Uso del Software Utilitario	En el producto se evidencia que se utiliza el software utilitario de manera óptima, demostrando un dominio completo de las herramientas y optimizando el	En el producto se evidencia un uso adecuado del software utilitario, con algunos errores menores en la ejecución.	En el producto se evidencia un uso básico del software utilitario, con varios errores o falta de optimización.	En el producto se evidencia un uso incorrecto o insuficiente del software utilitario, afectando negativamente el rendimiento del sistema.	**/4**

	rendimiento del sistema.				
3. Documentación y Evidencia Visual	En el producto se evidencia que se documenta cada paso de manera detallada y clara, con evidencia visual relevante y bien organizada.	En el producto se evidencia una documentación adecuada, pero con algunos detalles o evidencia visual que podría mejorarse.	En el producto se evidencia una documentación básica, con falta de detalles y evidencia visual incompleta o mal organizada.	En el producto se evidencia una documentación insuficiente, con falta de claridad y evidencia visual inadecuada o ausente.	**/4**
4. Identificación y Corrección de Fallas	En el producto se evidencia que se identifican y corrigen todas las fallas encontradas, utilizando procedimientos técnicos correctos y soluciones efectivas.	En el producto se evidencia que se identifican y corrigen la mayoría de las fallas, con algunas soluciones menos efectivas.	En el producto se evidencia que se identifican y corrigen algunas fallas, pero se omiten otras o se aplican soluciones inadecuadas.	En el producto se evidencia una identificación y corrección de fallas insuficiente, con muchas fallas sin abordar o mal resueltas.	**/4**
5. Informe Final	En el producto final se evidencia que se incluyen conclusiones claras, bien fundamentadas, y recomendaciones útiles basadas en un análisis exhaustivo.	En el producto final se evidencia que se incluyen conclusiones claras, pero con algunas recomendaciones o análisis que podrían ser más profundos.	En el producto final se evidencia que se incluyen conclusiones básicas, con recomendaciones o análisis superficiales.	En el producto final se evidencia que se incluyen conclusiones confusas o faltan recomendaciones, con un análisis insuficiente.	**/4**

3. EVALUACIÓN

Observaciones: _____

Juicio de valor: _____

4. FIRMAS

Aprendiz: _____

Instructor: _____