PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUIA DE0 APRENDIZAJE

- Denominación del Programa de Formación: Programación de Software
- Código del Programa de Formación: 233104
- Nombre del Proyecto Formativo (si aplica):
- Fase del Proyecto: Análisis
- Actividad de Proyecto Formativo: Establecer requisitos de la solución de software de acuerdo con estándares y procedimiento técnico
- Competencia: DESARROLLAR LA SOLUCIÓN DE SOFTWARE DE ACUERDO CON EL DISEÑO Y METODOLOGÍAS DE DESARROLLO
- Resultados de Aprendizaje: RAP 2. Interpretar el informe de requisitos de la solución conforme a la identificación de las necesidades del cliente.
- Duración de la Guía de Aprendizaje: 96 Horas

2. PRESENTACIÓN

Este logro esperado se orienta a que el aprendiz sea capaz de construir software siguiendo un diseño previamente definido asi:

Implica comprender y respetar la arquitectura, los modelos y los requisitos establecidos para la solución.

El aprendiz aplica metodologías de desarrollo que ordenan el trabajo y facilitan el control del proceso.

Promueve el uso de buenas prácticas de programación para garantizar calidad y mantenibilidad del software.

Permite desarrollar soluciones coherentes con las necesidades del cliente o del contexto organizacional.

Fomenta el trabajo disciplinado y estructurado dentro de un equipo de desarrollo.

Contribuye a la correcta implementación de funcionalidades según lo planeado en

el diseño.

Ayuda a reducir errores y retrabajos durante el ciclo de vida del software.

Fortalece la capacidad de adaptación a diferentes metodologías como ágiles o tradicionales.

Finalmente, prepara al aprendiz para un desempeño laboral eficiente y alineado con estándares profesionales.

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1. Actividades de reflexión inicial:

Descripción de la actividad:

Una institución educativa presenta dificultades en el procesamiento de las calificaciones de sus aprendices, ya que el cálculo de promedios y la determinación de aprobación o reprobación se realizan de forma manual. Esto genera errores en los resultados, demoras en la entrega de informes académicos y falta de confiabilidad en la información.

La institución solicita el desarrollo de una solución de software que permita registrar las calificaciones de los aprendices, calcular automáticamente el promedio final y emitir un resultado de aprobación o reprobación de acuerdo con los criterios establecidos. El aprendiz, desde su rol de desarrollador en formación, debe interpretar el informe de requisitos, identificar las necesidades del cliente y emplear lenguaje de programación para resolver el problema, aplicando lógica proposicional y estructuras de control, de acuerdo con metodologías de desarrollo de software.

Esta situación permite evidenciar el desempeño laboral del aprendiz en un contexto real y cercano a su entorno académico.

3.2. Pregunta problema

¿Cómo interpretar los requisitos de una institución educativa para desarrollar una solución algorítmica que calcule calificaciones finales y determine la aprobación o reprobación de los aprendices, aplicando lógica proposicional y estructuras de control?

3.3. **Actividad de aprendizaje** – Reflexión inicial

Técnica didáctica activa: Estudio de caso + trabajo colaborativo

Nombre de la actividad

"¿Aprueba o no aprueba?"

Descripción de la actividad

El instructor presenta a los aprendices un estudio de caso basado en una situación real: una institución educativa que necesita automatizar el cálculo de calificaciones. Se entrega un breve informe de requisitos que incluye criterios de evaluación, nota mínima de aprobación y número de evaluaciones.

En equipos de trabajo, los aprendices analizan el caso y reflexionan sobre cómo debería resolverse el problema desde el punto de vista lógico y algorítmico, sin escribir aún código.

Desarrollo de la actividad

1. Lectura y análisis del informe de requisitos.
2. Identificación de las necesidades del cliente.
3. Determinación de datos de entrada, procesos y salidas.
4. Discusión grupal sobre las condiciones lógicas (aprobado / reprobado).
5. Socialización de conclusiones entre los equipos.

Producto de la actividad

- Mapa o cuadro de entradas, procesos y salidas.
- Listado de condiciones lógicas necesarias para resolver el problema.
- Primer acercamiento conceptual al algoritmo.

Propósito formativo

Favorecer la reflexión inicial del aprendiz, activar conocimientos previos en lógica proposicional y algoritmia, y fortalecer la competencia de interpretar requisitos del cliente antes de desarrollar una solución de software.

Ambiente requerido: Ambiente

Estrategias o técnicas didácticas activas: Estudio de caso + trabajo colaborativo

Materiales de formación: internet

Material de apoyo: Documentos Anexo 1.

Duración de la actividad: 2 horas.

3.4. Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje:

Descripción de la actividad:

Contexto de la situación problema

Entorno productivo real:

La situación se sitúa en una institución educativa (colegio, instituto técnico o academia) donde el personal administrativo y los docentes realizan el registro de calificaciones de manera manual. El desempeño laboral del aprendiz se centra en el desarrollo de soluciones de software que apoyen procesos administrativos y académicos, específicamente en el cálculo de promedios y determinación de aprobación o reprobación de los aprendices.

¿Quién tiene el problema y cuándo/dónde surge?

El problema lo tiene el área administrativa y los docentes de la institución educativa. Surge durante el cierre de periodos académicos, cuando se deben procesar las notas de decenas o cientos de aprendices y generar informes confiables.

Causa del problema:

El problema se relaciona con la falta de automatización y el uso de métodos manuales para calcular promedios y determinar aprobación o reprobación. Esto puede provocar errores humanos, retrasos en la entrega de resultados y dificultades para el análisis académico. La solución requiere desarrollar habilidades en interpretación de requisitos, lógica proposicional y estructuras de control, aplicables en entornos reales de software educativo.

Importancia para el aprendiz:

Trabajar en este contexto permite que el aprendiz desarrolle competencias directamente relacionadas con la solución de problemas reales mediante software,

fortaleciendo conocimientos en algoritmos, estructuras de control y lógica, mientras mejora su desempeño laboral en un entorno productivo real.

Actividad de aprendizaje – Contextualización e identificación de conocimientos previos

Técnica didáctica activa: Tormenta de ideas guiada + mapa conceptual

Nombre de la actividad: “Descubriendo cómo se calcula”

Descripción de la actividad

Un grupo de docentes necesita calcular calificaciones finales de un grupo de aprendices y determinar aprobación o reprobación. Se entrega un ejemplo de notas de varios estudiantes.

Los aprendices, en equipos pequeños, deben:

1. Analizar el ejemplo y describir cómo se haría el cálculo manualmente.
2. Identificar qué datos se necesitan para calcular los promedios y las condiciones de aprobación/reprobación.
3. Construir un mapa conceptual que relacione: entradas, procesos, salidas y condiciones lógicas.
4. Compartir sus mapas y reflexionar sobre las dificultades de hacer este proceso manualmente.

Producto de la actividad

- Mapa conceptual que represente el flujo de información y cálculo.
- Listado de datos requeridos para resolver el problema.
- Reflexión escrita sobre los retos del cálculo manual y la importancia de automatizarlo.

Propósito formativo

- Activar los conocimientos previos sobre cálculos, promedios y condiciones lógicas.
- Identificar los elementos clave para el desarrollo de la solución algorítmica.
- Contextualizar al aprendiz en un entorno laboral real y motivarlo para el diseño de software que resuelva un problema concreto.

Ambiente requerido:

Estrategias o técnicas didácticas activas: Tormenta de ideas guiada + mapa conceptual

Materiales de formación: internet

Material de apoyo: Anexo 1

Duración de la actividad: 20 horas.

3.5. **Actividades de apropiación**:

Objetivos de aprendizaje

Objetivos Cognitivos (conocimiento):

- Comprende los conceptos de lógica proposicional y estructuras de control aplicadas al desarrollo de algoritmos.
- Identifica los elementos de entrada, procesos y salida necesarios para desarrollar una solución de software.
- Analiza los requisitos del cliente y los traduce en especificaciones funcionales de software.

Objetivos Procedimentales (habilidad):

- Diseña algoritmos en pseudocódigo o diagramas de flujo aplicando estructuras secuenciales, condicionales, cíclicas y arreglos.
- Implementa soluciones de software utilizando un lenguaje de programación o herramientas de desarrollo como PSeInt.
- Realiza prueba de escritorio para validar la lógica y funcionamiento de la solución desarrollada.

Objetivos Valorativos / Actitudinales:

- Demuestra responsabilidad y disciplina en la interpretación de requisitos y desarrollo de soluciones.
- Aplica pensamiento crítico y atención al detalle en la construcción de algoritmos.
- Trabaja colaborativamente respetando ideas de sus compañeros y fomentando el aprendizaje en equipo.

4. Instrucciones para aprender los conocimientos del proceso productivo

Conocimiento 1: Lógica proposicional.

Instrucción:

- Analiza los problemas de calificación de estudiantes y determina las condiciones lógicas que definen aprobado o reprobado.
- Identifica y clasifica las estructuras secuenciales, condicionales y cíclicas aplicables al cálculo de calificaciones.
- Realiza los retos propuestos

Técnica didáctica activa: Tormenta de ideas guiada + diagramas de flujo

Agrupación: Individual y Equipos colaborativos

Producto: Mapa conceptual de lógica de calificación y listado de reglas lógicas.

Material de apoyo: Anexo 1, Anexo 2

Conocimiento 2: **Estructuras de control**

Instrucción:

- Interpreta los requisitos del cliente y define los datos de entrada, procesos y resultados esperados.
- Elabora el diagrama de flujo, aplicando estructuras secuenciales, condicionales y cíclicas.
- Comparte y discute con el equipo para validar la lógica y proponer mejoras.
  Técnica: Trabajo en equipo y revisión entre pares.

Agrupación: Equipos colaborativos

Producto: Diagrama de flujo del algoritmo de calificaciones.

Material de apoyo: Anexo 2.

Conocimiento 3: Prueba de escritorio

Instrucción:

- Transfiere el algoritmo a un lenguaje de programación o herramienta de desarrollo como PSeInt.
- Introduce casos de prueba y verifica los resultados mediante prueba de escritorio.
- Corrige errores y ajusta la solución según los resultados obtenidos.

Técnica: Laboratorio de programación y prueba de escritorio.

Agrupación: Individual

Producto: Código funcional y registros de prueba de escritorio.

Material de apoyo: Anexo 3, Anexo 4.

Conocimiento 4: Documentación y presentación de la solución

Instrucción:

- Documenta los pasos del algoritmo, la lógica empleada y los resultados de las pruebas.
- Presenta la solución al equipo o instructor explicando la funcionalidad, entradas, procesos y salidas.

Técnica: Presentación oral con soporte de documentos o diagramas.

Agrupación: Equipos colaborativos

Producto: Documento de diseño y presentación de la solución.

Tipo de agrupación por actividad

Actividad	Agrupación
Reflexión inicial / mapa conceptual	Equipos colaborativos
Diseño de algoritmo (pseudocódigo/diagrama)	Equipos colaborativos
Implementación y prueba de escritorio	Individual

Documentación y presentación	Equipos colaborativos

Productos a elaborar

1. Mapa conceptual de la lógica de calificaciones.
2. Listado de reglas lógicas y condiciones de aprobación/reprobación.
3. Pseudocódigo o diagrama de flujo del algoritmo.
4. Código funcional en PSeInt o lenguaje de programación.
5. Prueba de escritorio documentada.
6. Documento de diseño y presentación de la solución.

10. Temporización sugerida

Sesión	Duración	Actividad	Producto
1	1 hora	Reflexión inicial, análisis de la situación problema	Mapa conceptual, listado de reglas lógicas
2	1 hora	Diseño del algoritmo en pseudocódigo o diagrama de flujo	Pseudocódigo/diagrama de flujo
3	2 horas	Implementación de la solución en PSeInt o lenguaje de programación	Código funcional
4	1 hora	Prueba de escritorio y ajuste del algoritmo	Registros de prueba de escritorio
5	1 hora	Documentación y presentación de la solución	Documento y presentación final

Descripción de la actividad:

Ambiente requerido:

Estrategias o técnicas didácticas activas:

Materiales de formación: MATERIAL PARA REALIZAR LAS PRACTICAS

Material de apoyo: Anexo 1,Anexo 2

Evidencias de aprendizaje:

Instrumentos de evaluación:

LISTA DE CHEQUEO DE PRODUCTO

4.1. LISTA DE CHEQUEO DE DESEMPEÑO

CUESTIONARIO

Duración de la actividad: 6 horas.

Actividades de Transferencia el Conocimiento:

Producto, desempeño y conocimiento esperado

Producto final esperado:

- Algoritmo funcional que registre las calificaciones de varios aprendices, calcule promedios y determine si aprueban o reprueban, implementado en un lenguaje de programación o herramienta como PSeInt, acompañado de documentación y prueba de escritorio.

Desempeño esperado del aprendiz:

- Interpreta correctamente los requisitos del cliente.
- Diseña un algoritmo que aplique lógica proposicional y estructuras de control adecuadas.
- Implementa el algoritmo de manera funcional y realiza pruebas para validar resultados.
- Documenta el proceso de desarrollo y explica claramente la solución a sus compañeros o instructor.

Conocimiento esperado:

- Comprende la lógica proposicional y estructuras secuenciales, condicionales y cíclicas aplicadas a la solución de problemas.
- Identifica datos de entrada, procesos y salidas para modelar un problema real.
- Aplica metodologías de desarrollo de software desde el análisis hasta la implementación y prueba del algoritmo.

Instrucciones en secuencia para concretar el producto final

Conocimiento de proceso: Desarrollo de algoritmos para la solución de problemas de calificación.

Instrucciones:

1. Analiza el informe de requisitos proporcionado por la institución educativa y determina las condiciones de aprobación y reprobación de los aprendices.
2. Identifica los datos de entrada necesarios (número de estudiantes, calificaciones parciales, nota mínima de aprobación).
3. Determina los procesos que se deben realizar (cálculo de promedio, comparación con nota mínima, generación de resultados).
4. Define las salidas esperadas (promedio de cada estudiante y estado “aprobado/reprobado”).
5. Diseña el algoritmo en pseudocódigo o diagrama de flujo aplicando estructuras secuenciales, condicionales, cíclicas y arreglos para organizar los datos de los estudiantes.
6. Verifica el algoritmo mediante revisión en equipo y discusión de la lógica aplicada, ajustando errores antes de la codificación.
7. Implementa el algoritmo en PSeInt o lenguaje de programación, respetando la estructura lógica definida.
8. Introduce casos de prueba y registra los resultados de cada prueba, comparando con los resultados esperados.
9. Corrige y ajusta la solución según los resultados de la prueba de escritorio hasta que los cálculos sean correctos.
10. Documenta la solución describiendo entradas, procesos, salidas, estructura del algoritmo y resultados de prueba.
11. Presenta la solución final al instructor y al equipo explicando la lógica, la implementación y cómo se resuelve la situación problema.

Descripción de la actividad:

Ambiente requerido:

Estrategias o técnicas didácticas activas:

Materiales de formación

Material de apoyo:

Evidencias de aprendizaje:

LISTA DE CHEQUEO DE PRODUCTO

LISTA DE CHEQUEO DE DESEMPEÑO

CUESTIONARIO

Instrumentos de evaluación:

Duración de la actividad: horas.

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

PLAN DE EVALUACIÓN

Norma: Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo

Resultado de Aprendizaje:

RAP 2. Interpretar el informe de requisitos de la solución conforme a la identificación de las necesidades del cliente.

4.3. Enfoque de la evaluación

La evaluación se concibe como un proceso continuo, integral y orientado al desempeño laboral, que permite valorar el avance del aprendiz durante el desarrollo del proyecto (evaluación formativa) y verificar el logro de los resultados de aprendizaje al finalizar el ABP (evaluación sumativa).

Se evalúan de manera integrada:

- El producto (solución de software desarrollada).
- El desempeño (cómo el aprendiz aplica el proceso de desarrollo).
- El conocimiento (comprensión de los conceptos y principios trabajados).

4.4. Evaluación formativa (durante el proceso)

Propósito

Acompañar el aprendizaje, identificar dificultades oportunamente y retroalimentar al aprendiz para mejorar su desempeño y la calidad de la solución.

Momentos de evaluación formativa

Momento	Estrategia	Evidencia	Instrumento
Análisis del problema	Observación y preguntas orientadoras	Identificación de requisitos, entradas y salidas	Lista de chequeo
Diseño del algoritmo	Revisión entre pares y retroalimentación del instructor	Pseudocódigo o diagrama de flujo	Rúbrica analítica
Implementación	Acompañamiento en laboratorio	Código parcial funcional	Guía de observación
Prueba de escritorio	Discusión de resultados	Casos de prueba documentados	Lista de verificación

Criterios de evaluación formativa

- Interpreta correctamente los requisitos del cliente.
- Aplica lógica proposicional de forma coherente.
- Usa estructuras secuenciales, condicionales y cíclicas adecuadamente.
- Muestra disposición para corregir errores y mejorar la solución.

La retroalimentación se realiza de manera oral y escrita, orientando al aprendiz a mejorar antes de la entrega final.

4.5. Evaluación sumativa (al finalizar el proyecto)

Propósito

Verificar el logro del resultado de aprendizaje y el nivel de desempeño alcanzado por el aprendiz frente a la norma de competencia laboral.

Evidencias de evaluación sumativa

Tipo de evidencia	Descripción
Producto	Algoritmo funcional implementado en PSeInt o lenguaje de programación

Desempeño	Desarrollo ordenado del proceso: análisis, diseño, implementación y prueba
Conocimiento	Explicación clara de la lógica, estructuras usadas y resultados obtenidos

4.6. Criterios de evaluación sumativa

Criterio	Evidencia observable
Interpretación de requisitos	La solución responde a las necesidades del cliente
Lógica proposicional	Condiciones de aprobación/reprobación correctamente implementadas
Estructuras de control	Uso adecuado de secuencias, condicionales, ciclos y arreglos
Funcionamiento del algoritmo	Cálculos correctos y resultados confiables
Prueba de escritorio	Casos de prueba claros y coherentes
Documentación	Claridad y orden en la explicación del proceso
Actitud laboral	Responsabilidad, orden y cumplimiento de entregas

Fase del proyecto formativo	Actividad del proyecto formativo	Actividad de Aprendizaje	Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación

5. GLOSARIO DE TÉRMINOS

Algoritmo

Conjunto ordenado y finito de instrucciones lógicas que permiten resolver un problema o realizar una tarea específica paso a paso.

Aprendizaje Basado en Proyectos (ABP)

Metodología activa en la que el aprendiz desarrolla conocimientos, habilidades y actitudes mediante la resolución de un problema real, contextualizado y significativo.

Arreglo

Estructura de datos que permite almacenar y organizar múltiples valores del mismo tipo bajo un solo nombre, facilitando su acceso y procesamiento mediante índices.

Competencia laboral

Capacidad demostrada por una persona para desempeñar una función productiva de manera eficaz, integrando conocimientos, habilidades y actitudes en un contexto real de trabajo.

Criterio de evaluación

Parámetro o referente que permite valorar el nivel de logro de un resultado de aprendizaje a partir del desempeño, el producto o el conocimiento del aprendiz.

Desempeño

Manifestación observable de las acciones que realiza el aprendiz durante el desarrollo de una actividad, evidenciando la aplicación de conocimientos y habilidades.

Estructuras de control

Elementos de la programación que permiten definir el flujo de ejecución de un algoritmo, incluyendo estructuras secuenciales, condicionales y cíclicas.

Evaluación formativa

Proceso continuo de valoración del aprendizaje que tiene como propósito retroalimentar al aprendiz y mejorar su desempeño durante el desarrollo de la actividad.

Evaluación sumativa

Evaluación que se realiza al finalizar un proceso formativo para verificar el logro de los resultados de aprendizaje y emitir un juicio sobre el nivel de competencia alcanzado.

Lógica proposicional

Rama de la lógica matemática que analiza proposiciones y sus valores de verdad, utilizada en programación para tomar decisiones mediante condiciones.

Metodologías de desarrollo de software

Conjunto de enfoques, principios y prácticas que orientan de manera estructurada las etapas del desarrollo de software, desde el análisis hasta la implementación y prueba.

Prueba de escritorio

Técnica de verificación que consiste en simular manualmente la ejecución de un algoritmo para validar su lógica y los resultados esperados antes de su implementación.

Resultado de aprendizaje

Enunciado que describe lo que el aprendiz debe saber, saber hacer y demostrar al finalizar un proceso formativo, en coherencia con una competencia laboral.

Solución de software

Producto tecnológico desarrollado para resolver un problema específico mediante el uso de algoritmos, estructuras de datos y lenguaje de programación.

Técnica didáctica activa

Estrategia pedagógica que involucra al aprendiz de manera participativa en su proceso de aprendizaje, promoviendo el análisis, la reflexión y la construcción del conocimiento.

6. REFERENTES BILBIOGRÁFICOS

Ausubel, D. P. (2002). *Adquisición y retención del conocimiento: Una perspectiva cognitiva*. Paidós.

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university* (4th ed.). Open University Press.

Bruner, J. S. (1997). *La educación, puerta de la cultura*. Visor.

Díaz Barriga, F., & Hernández Rojas, G. (2010). *Estrategias docentes para un aprendizaje significativo: Una interpretación constructivista* (3.ª ed.). McGraw-Hill.

Jonassen, D. H. (2011). *Learning to solve problems: A handbook for designing problem-solving learning environments*. Routledge.

Marquès Graells, P. (2012). *Metodologías activas para la formación por competencias*. Universidad Autónoma de Barcelona.

Ministerio de Educación Nacional. (2016). *Orientaciones pedagógicas para la formación por competencias*. MEN.

Pressman, R. S., & Maxim, B. R. (2020). *Ingeniería de software: Un enfoque práctico* (9.ª ed.). McGraw-Hill.

Sommerville, I. (2016). *Ingeniería del software* (10.ª ed.). Pearson Educación.

SENA. (2013). *Modelo pedagógico de la formación profesional integral*. Servicio Nacional de Aprendizaje.

SENA. (2014). *Lineamientos para la evaluación del aprendizaje en la formación profesional integral*. Servicio Nacional de Aprendizaje.
SENA. (2017). *Diseño curricular basado en competencias laborales*. Servicio Nacional de Aprendizaje.
SENA. (2020). *Guía para la formulación de resultados de aprendizaje y criterios de evaluación*. Servicio Nacional de Aprendizaje.

Referentes específicos en algoritmia y lógica
Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.
Joyanes Aguilar, L. (2018). *Fundamentos de programación: Algoritmos, estructuras de datos y objetos*. McGraw-Hill.
Rosen, K. H. (2019). *Discrete mathematics and its applications* (8th ed.). McGraw-Hill Education.

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)				

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					