**PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**FORMATO GUÍA DE APRENDIZAJE**

## 1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: Análisis y Desarrollo de Software.
- Código del Programa de Formación: 228118
- Nombre del Proyecto Formativo (si aplica): Integración de tecnologías en el desarrollo de software.
- Fase del Proyecto (si aplica): Planeación.
- Actividad de Proyecto Formativo (si aplica): Estructurar procesos lógicos para la construcción de algoritmos.
- Competencia: Aplicación de conocimientos de las ciencias naturales de acuerdo con situaciones del contexto productivo y social.
- Resultados de Aprendizaje:
    - Solucionar problemas asociados con el sector productivo con base en los principios y leyes de la física.
    - Proponer acciones de mejora en los procesos productivos de acuerdo con los principios.
    - Verificar las transformaciones físicas de la materia utilizando herramientas
    - Identificar los principios y leyes de la física en la solución de problemas de acuerdo al contexto productivo.
- Duración de la Guía de Aprendizaje (horas): 40.

## 2. PRESENTACIÓN

La algoritmia es un área fundamental que estudia los algoritmos, secuencias de instrucciones definidas y finitas que permiten resolver problemas o llevar a cabo tareas específicas. Los algoritmos son una herramienta esencial en campos como la informática, las matemáticas, la ingeniería, y en realidad, en muchos aspectos de la vida cotidiana, desde la búsqueda de información en internet hasta la

realización de transacciones bancarias o la toma de decisiones logísticas. Podemos imaginar los algoritmos como una receta de cocina: siguen un conjunto de pasos bien definidos para lograr un objetivo, y la ciencia de la algoritmia se ocupa de diseñar, analizar y optimizar estos procesos.

Un algoritmo debe cumplir varias características: ser finito, es decir, terminar después de un número determinado de pasos; ser claro, con instrucciones precisas y sin ambigüedades; ser eficiente, utilizando la menor cantidad de recursos posibles; y tener entradas y salidas claramente definidas, proporcionando resultados efectivos. Estas características aseguran que los algoritmos sean herramientas poderosas para resolver problemas de manera eficaz.

La importancia de la algoritmia radica en que permite abordar problemas complejos de una manera estructurada. En un mundo donde se manejan grandes volúmenes de información y se requiere rapidez en la toma de decisiones, contar con algoritmos optimizados es clave. Por ejemplo, los motores de búsqueda como Google utilizan algoritmos para encontrar información relevante en cuestión de segundos. En logística, las empresas usan algoritmos para planificar rutas de entrega de manera óptima. En la seguridad informática, los algoritmos criptográficos protegen la información sensible, y en la inteligencia artificial, permiten que las máquinas aprendan y mejoren su desempeño con el tiempo. Por ello, la algoritmia no solo es una disciplina académica, sino una herramienta clave en la tecnología y el desarrollo actual.

Existen distintos tipos de algoritmos que se adaptan a diferentes problemas. Por ejemplo, los algoritmos de búsqueda permiten encontrar un elemento en una estructura de datos, mientras que los algoritmos de ordenación organizan conjuntos de datos de manera eficiente. Los algoritmos recursivos resuelven problemas dividiéndolos en subproblemas más pequeños, y los algoritmos de grafos son útiles para resolver problemas de redes, como encontrar el camino más corto en un mapa. También existen algoritmos de aproximación, que proporcionan soluciones cercanas al óptimo cuando una solución exacta es inviable debido a limitaciones de tiempo o recursos.

En el estudio de los algoritmos es fundamental el concepto de complejidad algorítmica, que mide los recursos que un algoritmo necesita para ejecutarse, principalmente tiempo y espacio. Se utiliza la notación de O grande para describir la complejidad en términos de cómo crece el tiempo de ejecución a medida que aumenta el tamaño del problema. Entender la complejidad de un algoritmo permite comparar distintas soluciones y seleccionar la más adecuada según el contexto. Esto es esencial cuando se trabaja con grandes volúmenes de datos, donde una diferencia en eficiencia puede representar una gran mejora en el desempeño.

El proceso de diseño de algoritmos comienza con un análisis detallado del problema a resolver. Una vez entendido el problema, se diseña una solución paso a paso, considerando las restricciones y objetivos. Después, se valida el algoritmo, asegurándose de que funcione correctamente en distintos escenarios. La implementación del algoritmo en un lenguaje de programación transforma la idea teórica en una herramienta funcional, y finalmente, se analiza la eficiencia del algoritmo, buscando formas de optimizar su rendimiento. Estas etapas permiten que los algoritmos no solo resuelvan problemas, sino que lo hagan de la manera más eficiente posible.

Para motivar el aprendizaje y fomentar el trabajo autónomo, es importante que los estudiantes vean la relación entre los algoritmos y su aplicación en la vida cotidiana. Por ejemplo, un algoritmo de



búsqueda binaria permite encontrar elementos en listas ordenadas de forma eficiente, lo que es la base de muchas funciones en bases de datos y motores de búsqueda. Algoritmos como Quicksort permiten ordenar grandes cantidades de datos rápidamente, y el algoritmo de Dijkstra, que encuentra el camino más corto en un grafo, es esencial en aplicaciones como los sistemas de navegación por GPS.

Relacionar el conocimiento previo con los nuevos aprendizajes es crucial para la apropiación de los conceptos. Los estudiantes pueden identificar cómo han utilizado la lógica y la toma de decisiones estructurada en otras áreas, como en la matemática básica o incluso en juegos de estrategia, y transferir esos conceptos al estudio de la algoritmia. Además, el aprendizaje colaborativo juega un rol importante en la resolución de problemas algorítmicos. Discutir soluciones en equipo, compartir ideas y enfrentarse a problemas complejos en grupo no solo refuerza el entendimiento individual, sino que también permite descubrir nuevas formas de abordar los desafíos.

El estudio de la algoritmia, además de ser una herramienta teórica, ofrece oportunidades para el desarrollo de proyectos prácticos donde los estudiantes pueden aplicar los algoritmos en situaciones del mundo real. Esto fomenta la transferencia de conocimientos y permite a los estudiantes conectar lo aprendido en el aula con problemas cotidianos, como la optimización de recursos en una empresa o el análisis de datos para la toma de decisiones. La práctica continua y la participación en desafíos de programación son excelentes formas de profundizar en el conocimiento de algoritmos, mejorando las habilidades de resolución de problemas y la capacidad de aplicar la teoría en contextos diversos.

La algoritmia es una herramienta poderosa para resolver problemas de manera eficiente, y su estudio permite desarrollar habilidades de pensamiento lógico, estructurado y crítico. Aprender algoritmia no solo es relevante para quienes desean dedicarse a la programación, sino para cualquier persona que busque mejorar su capacidad de análisis y resolución de problemas en su vida académica, profesional y personal. La clave está en relacionar los conceptos con la realidad, practicar de manera constante y colaborar en la resolución de problemas complejos para lograr una verdadera apropiación del conocimiento.

## 3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

### 3.1 Actividades de reflexión inicial:

**Descripción de la actividad:**

Se inicia la formación explicando que los algoritmos no son solo un concepto abstracto o relacionado exclusivamente con la programación, sino que también están presentes en muchas actividades diarias. Después, se pregunta a los aprendices: ¿Qué es lo primero que les viene a la mente cuando escuchan la palabra “algoritmo”? Esta pregunta sirve como punto de partida para que cada uno comparta sus ideas iniciales, relacionándolas con sus experiencias o ideas previas sobre el tema.

Se plantea a los aprendices una situación cotidiana: “Piensa en alguna actividad rutinaria que realices todos los días, como vestirte, preparar un desayuno, o ir al trabajo o la escuela. Ahora, trata de describir los pasos que sigues para completar esa actividad.”

A partir de esta consigna, los estudiantes tienen unos minutos para pensar en su actividad y descomponerla en una serie de pasos. Se les pide que piensen en la claridad y el orden de esos pasos, identificando las posibles decisiones o bifurcaciones en el proceso. Por ejemplo, si alguien describe cómo prepara el desayuno, podría reflexionar sobre decisiones como: “Si hay leche en la nevera, preparo cereal; si no, hago una tostada.”

Los aprendices, en grupos pequeños, comparten la actividad que eligieron y los pasos que identificaron. Luego, cada grupo selecciona uno de los procesos descritos y lo expone al resto de la clase. Se guía la conversación, subrayando cómo cada uno de estos procesos puede interpretarse como un algoritmo, ya que consisten en una serie de instrucciones ordenadas para lograr un objetivo.

Después de escuchar las descripciones de los aprendices, se introduce formalmente el concepto de algoritmo, explicando cómo esas actividades cotidianas que acaban de describir son, en esencia, algoritmos básicos que seguimos para realizar tareas de manera eficiente. La reflexión inicial permite que los aprendices vean la algoritmia como algo más cercano y práctico, facilitando la transición hacia el estudio técnico del tema.

Se invita a los aprendices a reflexionar sobre la pregunta: “¿En qué otras áreas de tu vida crees que podrías aplicar el pensamiento algorítmico?” Esta reflexión final ayuda a consolidar la idea de que el pensamiento lógico y algorítmico es una habilidad útil en diferentes contextos y no solo en el ámbito de la programación o la informática.

**Ambiente requerido:** Ambiente de formación.

**Estrategias o técnicas didácticas activas:**

Aprendizaje basado en experiencias, trabajo colaborativo y en grupo, refuerzo positivo y autoevaluación y reflexión.

**Materiales de formación:** Tablero, marcadores, borrador y equipo de cómputo.

**Material de apoyo:** Documento en PDF sobre algoritmia.

**Duración de la actividad:** 4 horas.

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

**Descripción de la actividad:**

Se comienza explicando brevemente que un algoritmo no es solo un conjunto de pasos en la informática, sino una secuencia de instrucciones claras que permiten resolver problemas en cualquier contexto. Para establecer una conexión con la vida diaria y el conocimiento previo de los estudiantes, se plantea la pregunta: "¿Cómo crees que las tareas que realizas a diario pueden ser vistas como un proceso algorítmico?"

Se forma un mapa conceptual colectivo en el que los aprendices, proponen ejemplos de actividades cotidianas que involucran procesos algorítmicos. Estas actividades pueden ir desde organizar una lista de tareas hasta planificar un viaje o cocinar una receta. El mapa se va construyendo en el tablero o digitalmente, visualizando cómo cada una de estas actividades involucra una serie de pasos o decisiones estructuradas.

Se introduce la idea de que algunos de los pasos de estos procesos incluyen decisiones (por ejemplo, si una ruta está congestionada, elegir una alternativa) o repeticiones (como revolver una mezcla en la cocina hasta lograr una consistencia). A partir de aquí, se pregunta: "¿Conocen algún proceso que se resuelva a través de decisiones o repeticiones?" El objetivo es que los aprendices relacionen estos conceptos con actividades conocidas o experiencias previas, ya sea en situaciones cotidianas o en otras asignaturas, como matemáticas o ciencias.

Los aprendices, individualmente o en parejas, hacen una lista de ejemplos que conozcan donde se tomen decisiones (como bifurcaciones en un algoritmo) o se repitan pasos hasta alcanzar un resultado. Esta etapa permite descubrir qué tanto conocen sobre la lógica algorítmica sin haberla estudiado formalmente, ayudando a identificar áreas de fortaleza y aspectos que necesitarán más atención.

A continuación, se explica que las estructuras algorítmicas básicas como las decisiones (condicionales) y las repeticiones (bucles) son clave en la resolución de problemas computacionales. Se ejemplifica cómo, en el contexto de la programación, estas mismas ideas se transforman en instrucciones que las computadoras pueden seguir.

Los aprendices comparten sus ejemplos y experiencias, y se va guiando la discusión hacia la formalización del pensamiento algorítmico, resaltando cómo los conceptos que ya conocen pueden aplicarse en el ámbito de la programación. Esta parte del ejercicio ayuda a generar conexiones entre lo que los aprendices ya entienden de manera intuitiva y el nuevo conocimiento que aprenderán.

La actividad concluye con una síntesis donde se refuerza la idea de que muchos de los procesos cotidianos que los aprendices identificaron son en realidad algoritmos en acción. De esta manera, se contextualiza el tema y se identifican los conocimientos previos que servirán de base para aprender a estructurar algoritmos más formales y complejos.

**Ambiente requerido:** Ambiente de formación.

**Estrategias o técnicas didácticas activas:**

Aprendizaje basado en experiencias, trabajo colaborativo y en grupo, refuerzo positivo y autoevaluación y reflexión.

**Materiales de formación:** Tablero, marcadores, borrador y equipo de cómputo.

**Material de apoyo:** Documento en PDF sobre algoritmia.

**Duración de la actividad:** 4 horas.

### 3.3 Actividades de apropiación:

**Descripción de la actividad:**

Se comienza revisando brevemente los conceptos clave que los aprendices han aprendido sobre algoritmos: la secuencia de pasos, las decisiones (condicionales) y las repeticiones (bucles). Explica que, en esta actividad, el objetivo es apropiarse de estos conceptos diseñando algoritmos para resolver problemas reales, utilizando lo que han aprendido en clases anteriores.

Se les presenta a los aprendices un problema realista que pueden resolver mediante la creación de un algoritmo. El problema debe estar relacionado con una situación cotidiana o con un contexto que les resulte familiar, para facilitar la conexión entre los conceptos teóricos y la vida práctica.

Los aprendices, organizados en grupos pequeños, deberán trabajar en la creación de un algoritmo que resuelva el problema presentado. Deberán pensar en cada uno de los pasos necesarios, tomando en cuenta las condiciones y las posibles decisiones o repeticiones que se requieren para llegar a la solución.

Se guía el proceso, respondiendo preguntas y ayudando a los aprendices a estructurar sus algoritmos de manera clara y lógica, pero dejando que sean ellos quienes propongan soluciones.

Después de haber diseñado el algoritmo, los grupos deben representarlo de manera visual, utilizando diagramas de flujo. Esto les ayudará a visualizar el proceso y a entender mejor la estructura de su algoritmo.

Cada grupo presenta su algoritmo al resto de la clase, explicando los pasos que han diseñado y cómo solucionan el problema. Los otros grupos pueden hacer preguntas o sugerencias para mejorar el diseño del algoritmo.

También se da retroalimentación constructiva, destacando qué aspectos del algoritmo son claros y bien definidos, y señalando posibles áreas de mejora en la eficiencia o en la estructura del proceso. El propósito es que los aprendices puedan refinar sus soluciones a partir de esta retroalimentación.

**Ambiente requerido:** Ambiente de formación.

**Estrategias o técnicas didácticas activas:**

Aprendizaje basado en experiencias, trabajo colaborativo y en grupo, refuerzo positivo y autoevaluación y reflexión.

**Materiales de formación:** Tablero, marcadores, borrador y equipo de cómputo.

**Material de apoyo:** Documento en PDF sobre algoritmia.

**Evidencias de aprendizaje:** Taller fundamentos algoritmia.

**Instrumentos de evaluación:** Prueba de conocimientos.

**Duración de la actividad:** 4 horas.

## 3.4 Actividades de Transferencia el Conocimiento:

**Descripción de la actividad:**

Se comienza recordando que los algoritmos no solo se aplican en problemas técnicos o de programación, sino que también son útiles para resolver problemas en diferentes áreas como la organización de recursos, la toma de decisiones, la optimización de procesos, entre otros. Explica que, en esta actividad, el reto es transferir los conocimientos algorítmicos adquiridos en clase a un campo diferente al de la programación, usando los mismos principios lógicos.

Se proponen varios escenarios o áreas temáticas nuevas que requieren la aplicación de un algoritmo para resolver un problema. Estos escenarios pueden abarcar desde la gestión empresarial hasta la planificación de eventos o la optimización de tareas en un entorno real.

En grupos pequeños, los aprendices seleccionan uno de los escenarios propuestos y diseñan un algoritmo que resuelva el problema planteado, pero trasladando el conocimiento aprendido en las clases de algoritmia a este nuevo contexto.

Los aprendices deben trabajar de manera autónoma, aplicando sus conocimientos previos para formular la solución.

Una vez que cada grupo haya diseñado su algoritmo, deben discutirlo dentro del grupo, revisando si cumple con las condiciones del problema y si es eficiente. Se pide a los aprendices que revisen posibles mejoras y que optimicen su solución si encuentran formas de hacerlo. Esta etapa refuerza el aprendizaje colaborativo y fomenta el pensamiento crítico.

Cada grupo presenta su algoritmo al resto de la clase, explicando el problema que resolvieron y el proceso que siguieron para diseñar su solución. Otros grupos pueden hacer preguntas o sugerir mejoras.

Se proporciona retroalimentación, enfocándose en cómo los aprendices lograron transferir el conocimiento algorítmico a un nuevo contexto, evaluando la claridad de los pasos, la eficiencia del algoritmo y su adecuación al problema planteado.

La actividad concluye con una reflexión grupal, donde se plantea preguntas como: "¿Cómo este ejercicio les ayudó a entender que los principios algorítmicos pueden aplicarse más allá de la programación?" y "¿En qué otras áreas creen que podrían aplicar el conocimiento sobre algoritmos?".

**Ambiente requerido:** Ambiente de formación.

**Estrategias o técnicas didácticas activas:**

Aprendizaje basado en experiencias, trabajo colaborativo y en grupo, refuerzo positivo y autoevaluación y reflexión.

**Materiales de formación:** Tablero, marcadores, borrador y equipo de cómputo.

**Material de apoyo:** Documento en PDF sobre algoritmia.

**Evidencias de aprendizaje:** Taller estructuras secuenciales, taller estructuras selectivas y taller estructuras iterativas.

**Instrumentos de evaluación:** Prueba de conocimientos.

**Duración de la actividad:** 4 horas.

## 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 | Estructurar procesos lógicos para la construcción de algoritmos | Algoritmia | Taller estructuras secuenciales.<br>Taller estructuras selectivas.<br>Taller estructuras iterativas. | Da respuesta sobre conectores lógicos, entidades primitivas, algoritmos, fases de los algoritmos, estructuras y su aplicación | Prueba de conocimientos. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | en la solución de un problema.<br><br>Representa procesos del sistema a partir de la construcción de algoritmos, como parte de la solución a situaciones planteadas.<br><br>Realizar algoritmos utilizando variables, constantes, variables, bucles, contadores, acumuladores, selección, pseudocódigo. | |

## 5. GLOSARIO DE TÉRMINOS

**Lectura:** Consiste en recibir desde un dispositivo de entrada (p.ej. el teclado) un valor.

**Escritura:** Consiste en mandar por un dispositivo de salida (p.ej. monitor o impresora) un resultado o mensaje.

**Algoritmo:** Secuencia de pasos lógicos para dar solución a un problema, con un inicio y fin definidos.

**Variable:** Espacio reservado de memoria cuyo valor puede cambiar durante la ejecución del programa.

**Constante:** Espacio reservado de memoria que permanece constante durante la ejecución del programa.

**Diagrama de flujo de datos:** Representación gráfica de un algoritmo.

**Asignar:** Establecer un valor a una variable.

**Condicional:** Estructura algorítmica de decisión que permite seguir un camino verdadero o falso en base a la o las condiciones.

**Ciclo Anidado:** Estructura algorítmica que consiste en ubicar un ciclo dentro de otro para realizar alguna tarea específica.

**Index:** Variable de tipo entera encargada de controlar (contar) el número de iteraciones o veces que se repite un ciclo.

**Ciclo:** Estructura algorítmica que permite repetir una o varias líneas de código según se cumpla una condición.

**Valor de verdad:** Unidades para validación de los procesos de la lógica, puede ser Verdadero o falso ó en su defecto Si o No ó 1 o 0.

## 6. REFERENTES BILBIOGRÁFICOS

Prieto, S. (sf). Principios de algoritmia.

Instituto tecnológico de Sonora (2016). Pseudocódigo y PSEINT.

## 7. CONTROL DEL DOCUMENTO

| | Nombre | Cargo | Dependencia | Fecha |
|---|---|---|---|---|
| **Autor (es)** | Raúl Gómez | Instructor | Red de Informática y Sistemas | 02/10/2024 |

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

| | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
|---|---|---|---|---|---|
| **Autor (es)** | | | | | |