

## ***GUÍA DE APRENDIZAJE***

### ***1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE***

- Denominación del Programa de Formación: ANALISIS Y DESARROLLO DE SOFTWARE
- Código del Programa de Formación: 228118
- Nombre del Proyecto Formativo (si aplica): 2919375
- Fase del Proyecto (si aplica): Analisis
- Actividad de Proyecto Formativo (si aplica):
- Competencia: ANÁLISIS DE LA ESPECIFICACIÓN DE REQUISITOS DEL SOFTWARE.
- Resultados de Aprendizaje: DESARROLLAR PROCESOS LÓGICOS A TRAVÉS DE LA IMPLEMENTACIÓN DE ALGORITMOS.
- Duración de la Guía de Aprendizaje (horas): 67 horas - 90 horas

### ***2. PRESENTACIÓN***

Esta guía tiene como propósito acompañarte en el desarrollo de habilidades para comprender desde cero la lógica de programación, permitiendo así la construcción de algoritmos que solucionen problemas reales. Empezarás desde lo más básico de la lógica proposicional hasta llegar al manejo de estructuras complejas como archivos.

### ***3. ¿POR QUÉ ES IMPORTANTE APRENDER ESTOS TEMAS?***

El pensamiento lógico es la base de la programación. Entender cómo se resuelven problemas a través de algoritmos te permitirá dar soluciones eficientes, estructuradas y escalables en cualquier lenguaje de programación. Estas competencias son esenciales para tu formación como desarrollador.

### ***4. ¿QUÉ VAS A APRENDER CON ESTA GUÍA?***

- Fundamentos de la lógica proposicional.
- Construcción de algoritmos con estructuras secuenciales, condicionales y cíclicas.
- Uso de funciones, procedimientos, arreglos y archivos en la solución de problemas.
- Desarrollo del pensamiento computacional para la resolución de problemas.

### ***5. LA RUTA HACIA EL APRENDIZAJE INTEGRAL***

1. Reflexión y motivación inicial.
2. Identificación de saberes previos.
3. Apropiación de conceptos y práctica guiada.
4. Transferencia de conocimientos a casos reales.
5. Evaluación formativa y retroalimentación.

## ***6. MISIÓN COMO APRENDIZ***

Serás capaz de analizar problemas, abstraer soluciones y representarlas de forma clara mediante algoritmos lógicos, sentando bases para convertirte en un programador estructurado y eficaz.

## ***7. REFLEXIÓN Y FRASE MOTIVACIONAL***

"La programación no es más que decirle a la máquina lo que uno ya sabe hacer. Aprende a pensar, y sabrás programar".

## ***8. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE***

### ***Actividad de Reflexión Inicial***

**Descripción:** Foro participativo donde los aprendices reflexionan sobre el papel de la lógica en la vida cotidiana.

**Ambiente requerido:** Aula virtual o presencial.

**Estrategias:** Pregunta detonante, lluvia de ideas.

**Materiales:** Computador, pizarra, marcadores.

**Duración:** 2 horas.

**Cierre:** Conclusiones compartidas en plenaria.

### ***Actividades de Contextualización e Identificación de Conocimientos Previos***

**Tema:** Solución de problemas con lógica proposicional.

**Descripción:** Actividades diagnósticas de razonamiento lógico, tablas de verdad y conectores.

**Ambiente:** Aula de clase con acceso a computadores.

**Estrategias:** Talleres, ejercicios prácticos guiados.

**Materiales:** Guías, videos.

**Duración:** 4 horas.

**Cierre:** Evaluación diagnóstica de conceptos.

## *Actividades de Apropiación*

**Tema:** Estructuras secuenciales y de control.

**Descripción:** Taller de construcción de algoritmos simples utilizando pseudocódigo.

**Ambiente:** Aula de sistemas.

**Estrategias:** Resolución de retos en parejas.

**Materiales:** Diagramas de flujo, cuaderno, software de pseudocódigo.

**Duración:** 10 horas.

**Cierre:** Socialización de soluciones y retroalimentación.

**Tema:** Estructuras cíclicas.

**Descripción:** Resolución de problemas que requieren ciclos (mientras, repetir, para).

**Ambiente:** Aula de sistemas.

**Estrategias:** Aprendizaje basado en problemas.

**Materiales:** Entorno de programación visual (como PSeInt).

**Duración:** 6 horas.

**Cierre:** Presentación de retos solucionados.

**Tema:** Funciones y procedimientos.

**Descripción:** Diseño modular de algoritmos.

**Ambiente:** Aula de sistemas.

**Estrategias:** Trabajo colaborativo.

**Materiales:** Editor de código, guía de ejercicios.

**Duración:** 6 horas.

**Cierre:** Revisión cruzada entre pares.

**Tema:** Arreglos y archivos.

**Descripción:** Creación de algoritmos que gestionen listas de datos y persistencia en archivos.

**Ambiente:** Aula de sistemas.

**Estrategias:** Práctica guiada.

**Materiales:** IDE, ejemplos de datos.

**Duración:** 10 horas.

**Cierre:** Prueba funcional de soluciones.

## *Actividades de Transferencia del Conocimiento*

**Descripción:** Proyecto final: Solución de un problema real mediante la creación de un algoritmo modular completo.

**Ambiente:** Aula de sistemas y acceso a repositorio compartido.

**Estrategias:** Aprendizaje basado en proyectos.

**Materiales:** Guía de proyecto, instructivos técnicos.

**Duración:** 12 horas.

**Cierre:** Presentación del proyecto y retroalimentación grupal.