**PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**FORMATO GUÍA DE APRENDIZAJE**

**IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE**

* **Denominación del Programa de Formación:** Tecnología en Análisis y Desarrollo de Software
* **Código del Programa de Formación:** 228118
* **Nombre del Proyecto:** Construcción de software integrador de tecnologías orientadas a servicios.
* **Fase del Proyecto:** Planeación
* **Actividad de Proyecto:** Determinar la Estructura Lógica y Tecnológica del Sistema
* **Competencia:** Modelado de los artefactos del software
* **Resultados de aprendizaje a alcanzar:** 
  + Estructurar el modelo de datos del software de acuerdo con las especificaciones del análisis.
* **Duración de la Guía de aprendizaje:** 50 horas

**2. PRESENTACIÓN**

En el vasto mundo del desarrollo de software, dos aspectos fundamentales emergen como pilares esenciales: la Arquitectura de la Información y la eficiencia de los algoritmos secuenciales. Mientras que la Arquitectura de la Información se encarga de organizar y estructurar la información en sistemas interactivos, los algoritmos secuenciales son la base sobre la cual se construyen las funcionalidades de un software. En esta presentación, exploraremos la intersección entre estos dos aspectos, destacando la importancia de la Arquitectura de la Información en el diseño de sistemas y proporcionando estrategias para mejorar la eficiencia de los algoritmos secuenciales.

**3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE**

En el ámbito de la enseñanza de algoritmos, las actividades de aprendizaje juegan un papel fundamental. Estas actividades son acciones diseñadas e intencionadas, que se llevan a cabo paso a paso con el objetivo de alcanzar metas específicas tanto en la teoría como en la práctica. En esta guía, las actividades están cuidadosamente diseñadas como hitos parciales, cuya suma contribuye al logro de los objetivos de aprendizaje establecidos.

Es crucial que cada actividad se aborde de manera comprensiva y crítica. Cada una de ellas proporciona la información y los conocimientos necesarios para avanzar en la construcción de evidencias de aprendizaje requeridas para la evaluación y aprobación de los resultados de aprendizaje relacionados con la competencia en Especificación de Requisitos del Software.**¡Bienvenido/a y disfrute del aprendizaje!**

* 1. **Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.**

Título de la Actividad: Exploración Preliminar en Programación

Descripción:

Esta actividad tiene como objetivo introducir a los participantes en el mundo de la programación, contextualizándolos en los conceptos básicos necesarios para el aprendizaje posterior. Se centrará en identificar y comprender los conocimientos esenciales que se requerirán para comprender mejor las actividades posteriores.

Desarrollo de la Actividad:

Sesión Introductoria (2 horas):

* Se llevará a cabo una breve presentación sobre los conceptos básicos de la programación, incluyendo definiciones de algoritmos, operadores y tipos de datos.
* Se discutirán ejemplos simples para ilustrar cómo se aplican estos conceptos en la práctica.

Ejercicios Prácticos (4 horas):

* Los participantes realizarán una serie de ejercicios prácticos diseñados para aplicar los conceptos discutidos en la sesión introductoria.
* Se proporcionará asistencia individualizada para ayudar a los participantes a superar cualquier dificultad que puedan encontrar durante la realización de los ejercicios.

Resultados Esperados:

* Los participantes tendrán una comprensión básica de los conceptos fundamentales de la programación.
* Habrán identificado los conocimientos necesarios para participar en las actividades de aprendizaje posteriores.
* Estarán preparados para abordar actividades más avanzadas relacionadas con la programación.
  1. **Actividades de apropiación del conocimiento (Conceptualización y Teorización).**

Título de la Actividad: Entendiendo los Fundamentos de la Programación y su Aplicación Práctica

Descripción: Esta actividad está diseñada para proporcionar a los participantes una comprensión profunda de los conceptos fundamentales de la programación, así como para demostrar su aplicación práctica a través de ejercicios y proyectos.

Objetivos:

Comprender los principios básicos de la programación.

Explorar los conceptos de algoritmos, eficiencia y operadores en programación.

Aplicar los conocimientos teóricos en la resolución de problemas prácticos.

Fomentar la capacidad de abstracción y pensamiento lógico en la resolución de problemas de programación.

Desarrollo de la Actividad:

Ponencia sobre Mejoras en la Eficiencia de Algoritmos Secuenciales (6 horas):

* Los participantes asistirán a una ponencia donde se explorarán estrategias para mejorar la eficiencia de los algoritmos secuenciales.
* Se discutirán conceptos clave como complejidad algorítmica, optimización de código y selección de estructuras de datos.
* Se presentarán ejemplos prácticos y casos de estudio para ilustrar las estrategias discutidas.

Taller de Desarrollo de una Herramienta para el Cálculo de Factoriales (8 horas):

* Los participantes trabajarán en grupos para desarrollar una herramienta funcional para el cálculo de factoriales utilizando conceptos de programación aprendidos en la ponencia.
* Se proporcionará orientación y asistencia por parte de los instructores para abordar desafíos y resolver problemas durante el desarrollo de la herramienta.
* Al finalizar, cada grupo presentará su herramienta y explicará el enfoque utilizado para su desarrollo.

Sesión de Conocimiento sobre Conceptos Fundamentales y Aplicaciones de Operadores en Programación (6 horas):

* Los participantes asistirán a una sesión de conocimiento donde se explorarán los fundamentos y aplicaciones de los operadores en programación.
* Se discutirán diferentes tipos de operadores (aritméticos, lógicos, de asignación, etc.) y cómo se utilizan en diferentes contextos de programación.
* Se proporcionarán ejemplos prácticos y se animará a los participantes a realizar ejercicios para reforzar su comprensión de los conceptos presentados.

Resultados Esperados:

* Los participantes tendrán un entendimiento profundo de los fundamentos de la programación, incluyendo algoritmos, eficiencia y operadores.
* Habrán adquirido habilidades prácticas para aplicar estos conceptos en la resolución de problemas de programación.
* Se habrá fomentado el trabajo en equipo y la colaboración a través del desarrollo del taller.
* Los participantes estarán preparados para abordar desafíos de programación de manera más efectiva y eficiente.
  1. **Actividades de transferencia del conocimiento.**

**Nombre de la Actividad 1:**

Ponencia sobre Mejoras en la Eficiencia de Algoritmos Secuenciales

Técnica Didáctica: Exposición

Estrategia Didáctica: Aprendizaje colaborativo

Materiales para esta actividad: Portátil, pantalla y conectividad.

Resultado esperado de esta actividad:

Los participantes habrán adquirido un entendimiento profundo de las estrategias para mejorar la eficiencia de los algoritmos secuenciales.

Tiempo sugerido para la actividad: 6 horas

**Nombre de la Actividad 2:**

Taller de Desarrollo de una Herramienta para el Cálculo de Factoriales

Técnica Didáctica: Desarrollo Práctico

Estrategia Didáctica: Aprendizaje basado en proyectos

Materiales para esta actividad: Ordenadores, software de programación, pizarra o pantalla para proyecciones.

Resultado esperado de esta actividad:

Los participantes habrán desarrollado una herramienta funcional para el cálculo de factoriales utilizando conceptos de programación.

Tiempo sugerido para la actividad: 8 horas

**Nombre de la Actividad 3:**

Sesión de Conocimiento sobre Conceptos Fundamentales y Aplicaciones de Operadores en Programación

Técnica Didáctica: Clase Magistral

Estrategia Didáctica: Enseñanza expositiva y participativa

Materiales para esta actividad: Pizarra o pantalla para proyecciones, materiales de escritura, material de apoyo (como ejemplos de código).

Resultado esperado de esta actividad:

Al finalizar la sesión, los participantes comprenderán los fundamentos y aplicaciones de los operadores en programación.

Tiempo sugerido para la actividad: 6 horas

**4. ACTIVIDADES DE EVALUACIÓN**

Tome como referencia la técnica e instrumentos de evaluación citados en la guía de Desarrollo Curricular

|  |  |  |
| --- | --- | --- |
| **Evidencias de Aprendizaje** | **Criterios de Evaluación** | **Técnicas e Instrumentos de Evaluación** |
| **Evidencias de Conocimiento:**  Conceptos fundamentales y aplicaciones de operadores en programación  **Evidencias de Desempeño:**  Ponencia sobre mejoras en la eficiencia de algoritmos secuenciales.  **Evidencias de Producto:**  Taller de desarrollo de una herramienta para el cálculo de factoriales | Argumenta y estructura en un orden lógico la importancia de elaborar un buen diseño gráfico.  Responde correctamente a preguntas sobre usabilidad, iconografía, diseño web, mapa de navegación, interfaz.  Aplica los conceptos de interfaz gráfica en el proyecto formativo o estudio de caso. | Preguntas - Cuestionario.  Ensayo – Lista de Chequeo  Preguntas – Rúbrica. |

**5. GLOSARIO DE TÉRMINOS**

* Algoritmo: Conjunto finito de pasos definidos y ordenados que conducen a la solución de un problema.
* Eficiencia: Medida de la capacidad de un algoritmo para utilizar los recursos disponibles de manera óptima.
* Complejidad: Evaluación del número de operaciones o recursos necesarios para ejecutar un algoritmo en función del tamaño de la entrada.
* Recursividad: Técnica que implica que una función se llame a sí misma para resolver un problema más pequeño.
* Iteración: Proceso de repetir un conjunto de instrucciones en un algoritmo hasta que se cumpla una condición de salida.
* Operador: Símbolo o función que realiza una operación específica en uno o más operandos.
* Lógico: Operador utilizado para realizar operaciones lógicas como AND, OR y NOT.
* Aritmético: Operador utilizado para realizar operaciones matemáticas como suma, resta, multiplicación y división.

**6. REFERENTES BILBIOGRÁFICOS**

Nigel, Lim., L., Lim., Haribabu, Komatineni. (2022). Pseudocode and Demonstration of a Multi-Use Artificial Intelligence Algorithm to Perform Challenging and Highly Optimised Pipeline/Cable Routing Cases. doi: 10.4043/31360-ms

(2022). Logical Pseudocode: Connecting Algorithms with Proofs. doi: 10.48550/arxiv.2201.12572

(2023). Pseudocode for Describing Algorithms. Structural Dynamics, doi: 10.1017/9781108909617.014

M.M., de, Weerdt. (2018). What is Pseudocode. doi: 10.4233/UUID:8AF9F012-24FF-4CB0-95A3-F740ED52D047

**7. CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | Jesús Ariel González Bonilla | Instructor | Formación Profesional Integral | 04-04 |
|  |  |

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

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