



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- **Denominación del Programa de Formación:** TGO Análisis y Desarrollo de Sistemas de Información
- **Código del Programa de Formación:** 228106 V102
- **Nombre del Proyecto:** Construcción de un sistema de información que cumpla con los requerimientos del cliente en procesos que se lleven a cabo en el sector productivo del departamento de Caldas
- **Fase del Proyecto:** ANÁLISIS
- **Actividad de Proyecto:** Seleccionar la alternativa de solución que cumpla con los requerimientos establecidos por el cliente
- **Competencia:** Analizar los requisitos del cliente para construir el sistema de información.
- **Resultados de Aprendizaje Alcanzar:** Interpretar el informe de requerimientos, para determinar las necesidades tecnológicas en el manejo de la información, de acuerdo con las normas y protocolos establecidos en la empresa
- **Duración de la Guía:** 10 horas

2. PRESENTACIÓN

¿Qué es y para qué sirve programar?. Simplificando, diremos que el ordenador es una máquina con extraordinarias capacidades para realizar cálculos matemáticos, mostrar gráficos e imágenes, procesar y almacenar datos, etc. Esta gran potencialidad está a la espera de que una persona (nosotros) le saque partido, y para ello lo único que hace falta es proporcionarle instrucciones a dicha máquina.

Diremos por tanto que programar es definir instrucciones para ser ejecutadas por un ordenador. El resultado de esta tarea es un paquete de instrucciones al que llamamos programa. El objetivo de programar suele ser resolver un problema.

Durante este proceso de formación nos introduciremos en la programación de computadores, pero como elemento principal primero debemos entrenar la lógica de programación y para ello vamos a utilizar **algoritmos, diagramas de flujo y pseudocódigos** con el fin tener muy claro el proceso de resolución de problemas por medio de la computadora antes de utilizar cualquier lenguaje de programación. También abordaremos los diferentes tipos de estructuras que nos permitirán desarrollar programas que solucionen problemas dentro del ámbito del desarrollo de software.

De manera paralela al aprendizaje de la representación de algoritmos y actividades en torno al desarrollo de la lógica de programación, nos apoyaremos en aplicaciones como PSeInt las cuales nos permitirán acercarnos de una forma más amigable al maravilloso mundo de la programación.



3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

- **Descripción de la(s) Actividad(es)**
 - **Actividades de aprendizaje**
 - Aplicar fundamentos de programación en el desarrollo de aplicaciones con programación estructurada.
 - Aplicar estructuras secuenciales en la solución de ejercicios algorítmicos con programación estructurada.
 - Emplear estructuras condicionales en la solución de ejercicios algorítmicos con programación estructurada.
 - Utilizar estructuras cíclicas en la solución de ejercicios algorítmicos con programación estructurada.
 - **Actividad de Reflexión inicial**

Actividad de reflexión inicial

Socialice según indicaciones del instructor, las siguientes preguntas con lo que sabe hasta ahora sobre el tema.

- ¿Qué será la lógica de programación?
- ¿Que cree usted, ¿La lógica se aprende o se desarrolla?
- ¿Qué será la programación?
- ¿Qué será codificar un programa?
- ¿Para qué se programa en el ámbito de la informática?
- ¿En que ha contribuido el desarrollo de software a la humanidad?

de las acciones realizadas según indicaciones del instructor. En esta actividad se identificarán conocimientos previos de los aprendices.

LA LÓGICA DE PROGRAMACIÓN

La programación es el arte de hacer que una computadora realice lo que nosotros necesitemos. Para esto es necesario conozcamos su lenguaje, y como es que este funciona, pero no solo esto, debemos crear mensajes claros y sencillos, es decir, tenemos que estructurar las órdenes que damos para que no puedan interpretarse de otra manera.

Para evitar este tipo de cosas tenemos la lógica, para poder darle orden a nuestras ideas y que estas funcionen de manera correcta.

Pero ¿Cómo usaremos la lógica para programar? La respuesta es sencilla, antes de programar, abrir nuestro IDE y “empezar a tirar código” sobre la marcha, analizaremos lo que se requiere y desarrollaremos un esquema mental o diagrama de lo que será nuestro programa, este nos llevará a darnos una idea del mensaje



que le queremos dar a nuestra computadora para que realice una tarea. (Este principio de diagramar antes de programar es equivalente a pensar antes de hablar).

VARIABLES, CONSTANTES Y OPERADORES

Variables

En programación, una variable está formada por un espacio en el sistema de almacenaje (memoria principal de un ordenador) y un nombre simbólico (un identificador) que está asociado a dicho espacio. Ese espacio contiene una cantidad o información conocida o desconocida, es decir un valor

A grandes rasgos los tipos de datos de las variables pueden ser:

- **Entero**

Un dato de tipo entero es un número que no tiene punto decimal, por lo tanto, en sus operaciones jamás va a generar decimales. Por ejemplo 25, -96 y 0. El hecho de que los datos de tipo entero no generen decimales significa que operan con un juego de reglas llamado Aritmética Entera. Una variable que se declare de tipo entero podrá almacenar solamente datos de tipo entero.

- **Real**

Un dato de tipo real es un número que tiene punto decimal, por lo tanto, en sus operaciones puede generar decimales. Por ejemplo 12.3, -78.56 o 45.0. El hecho de que los datos de tipo real generen decimales significa que operan con un juego de reglas llamado Aritmética Real.

- **Carácter**

Representa un carácter que ocupa 1 byte de memoria. Los caracteres simples se representan con comillas simples 'a'. En este tipo de dato se encuentran todos los caracteres conocidos, una letra, un número, un símbolo especial. Los caracteres se almacenan como números usando la codificación ASCII, lo que significa que es posible hacer operaciones aritméticas con los caracteres.

- **Cadena**

En programación, una cadena de caracteres, palabras, ristra de caracteres o frase (string, en inglés) es una secuencia ordenada (de longitud arbitraria, aunque finita) de elementos que pertenecen a un cierto lenguaje formal o a una oración. En general, una cadena de caracteres es una sucesión de caracteres (letras, números u otros signos o símbolos). Si no se ponen restricciones al alfabeto, una cadena podrá estar formada por cualquier combinación finita de los caracteres disponibles (las letras de la 'a' a la 'z' y de la 'A' a la 'Z', los números del '0' al '9', el espacio en blanco ' ', símbolos diversos '!', '@', '%', etcétera).

Operadores

Son elementos que relacionan de forma diferente, los valores de una o más variables y/o constantes. Es decir, los operadores nos permiten manipular valores. Existen operadores de diferentes tipos, los cuales trataremos a continuación.



Operadores aritméticos

Se utilizan para realizar operaciones matemáticas.

<i>Operadores aritméticos:</i>		
<i>Pseudocódigo:</i>	<i>C:</i>	
+	+	Suma
-	-	Resta
*	*	Multiplicación
**		Potencia
/	/	División
div	/	División
mod	%	Módulo (resto de la división entera)
+	+	Signo más
-	-	Signo menos

© carlospes.com

Operadores relacionales

Un operador relacional se utiliza para comparar los valores de dos expresiones. Éstas deben ser del mismo tipo (aritméticas, lógicas o de carácter).

<i>Operadores relacionales:</i>		
<i>Pseudocódigo:</i>	<i>C:</i>	
<	<	Menor que
<=	<=	Menor o igual que
>	>	Mayor que
>=	>=	Mayor o igual que
=	==	Igual que
<>	!=	Distinto que

© carlospes.com



Operadores lógicos

Un operador lógico actúa, exclusivamente, sobre valores de expresiones lógicas.

© carlospes.com		
Operadores lógicos:		
<hr/>		
<i>Pseudocódigo:</i>	<i>C:</i>	
<hr/>		
y	&&	Conjunción
o		Disyunción
<hr/>		
no	!	Negación

Creación de algoritmos

Para la creación inicial de un algoritmo debemos tener en cuenta factores como el análisis inicial del problema al cual nos estamos enfrentando, datos de entrada, datos de salida y los procesos necesarios para con base en los datos de entrada generar los datos de salida.

Definición del Problema

Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa. Es importante que se conozca lo que se desea que realice la computadora; mientras esto no se conozca del todo no tiene mucho caso continuar con la siguiente etapa.

Análisis del Problema

Una vez que se ha comprendido lo que se desea de la computadora, es necesario definir:

- Los datos de entrada.
- Cuál es la información que se desea producir (salida)
- Los métodos y fórmulas que se necesitan para procesar los datos.

Una recomendación muy práctica es el que nos pongamos en el lugar de la computadora y analicemos que es lo que necesitamos que nos ordenen y en que secuencia para producir los resultados esperados.



TÉCNICAS PARA REPRESENTAR ALGORITMOS

Existen varias técnicas para representar algoritmos, sin embargo, en esta guía solo abordaremos las más utilizadas en el campo de la programación: **Diagramas de flujo y pseudocódigos**.

Diagramas de flujo

Los diagramas de flujo son una manera de representar visualmente el flujo de datos a través de sistemas de tratamiento de información. Los diagramas de flujo describen que operaciones y en que secuencia se requieren para solucionar un problema dado.

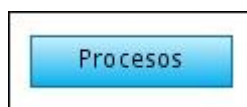
Un diagrama de flujo u organigrama es una representación diagramática que ilustra la secuencia de las operaciones que se realizarán para conseguir la solución de un problema. Los diagramas de flujo se dibujan generalmente antes de comenzar a programar el código frente a la computadora. Los diagramas de flujo facilitan la comunicación entre los programadores y la gente del negocio. Estos diagramas de flujo desempeñan un papel vital en la programación de un problema y facilitan la comprensión de problemas complicados y sobre todo muy largos. Una vez que se dibuja el diagrama de flujo, llega a ser fácil escribir el programa en cualquier idioma de alto nivel. Vemos a menudo cómo los diagramas de flujo nos dan ventaja al momento de explicar el programa a otros. Por lo tanto, está correcto decir que un diagrama de flujo es una necesidad para la documentación mejor de un programa complejo.

Reglas para dibujar un diagrama de flujo.

Los Diagramas de flujo se dibujan generalmente usando algunos símbolos estándares; sin embargo, algunos símbolos especiales pueden también ser desarrollados cuando sean requeridos. Algunos símbolos estándares, que se requieren con frecuencia para diagramar programas de computadora se muestran a continuación:



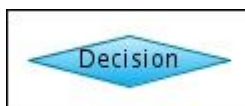
Inicio o fin del programa



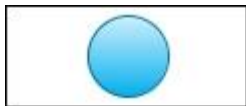
Pasos, procesos o líneas de instrucción de programa de computo



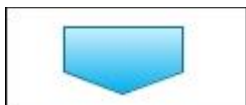
Operaciones de entrada y salida



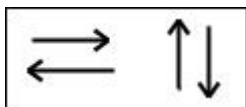
Toma de decisiones y Ramificación



Conector para unir el flujo a otra parte del diagrama



Conector de pagina



Líneas de flujo



Anotación

Reglas para la creación de Diagramas

- Los Diagramas de flujo deben escribirse de arriba hacia abajo, y/o de izquierda a derecha.
- Los símbolos se unen con líneas, las cuales tienen en la punta una flecha que indica la dirección que fluye la información de procesos, se deben de utilizar solamente líneas de flujo horizontal o verticales (nunca diagonales).
- Se debe evitar el cruce de líneas, para lo cual se quisiera separar el flujo del diagrama a un sitio distinto, se pudiera realizar utilizando los conectores. Se debe tener en cuenta que solo se van a utilizar conectores cuando sea estrictamente necesario.
- No deben quedar líneas de flujo sin conectar
- Todo texto escrito dentro de un símbolo debe ser legible, preciso, evitando el uso de muchas palabras.
- Todos los símbolos pueden tener más de una línea de entrada, a excepción del símbolo final.
- Solo los símbolos de decisión pueden y deben tener más de una línea de flujo de salida.

Ejemplos de diagramas de flujo

Hallar la hipotenusa de un triángulo rectángulo dados sus dos lados

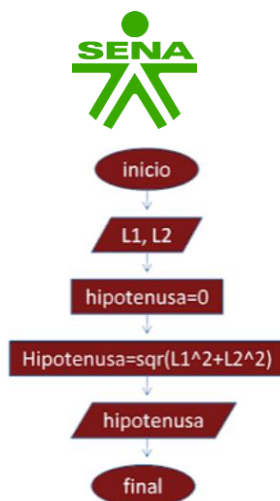
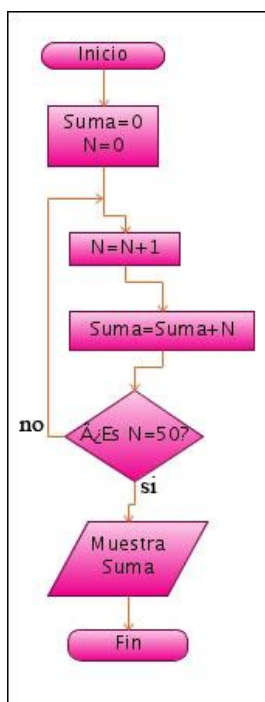


Diagrama de flujo que encuentra la suma de los primeros 50 números naturales



ALGORITMOS COMPUTACIONALES – SEUDOCODIGOS

En ciencias de la computación, y análisis numérico, el pseudocódigo (o falso lenguaje) es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.

Utiliza las convenciones estructurales de un lenguaje de programación real, pero está diseñado para la lectura humana en lugar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación. Normalmente, el pseudocódigo omite detalles que no son esenciales para la comprensión humana del algoritmo, tales como declaraciones de variables, código específico del sistema y algunas subrutinas. El lenguaje de programación se complementa, donde sea conveniente, con descripciones detalladas en lenguaje natural, o con notación matemática compacta.



Se utiliza pseudocódigo pues este es más fácil de entender para las personas que el código del lenguaje de programación convencional, ya que es una descripción eficiente y con un entorno independiente de los principios fundamentales de un algoritmo. Se utiliza comúnmente en los libros de texto y publicaciones científicas que se documentan varios algoritmos, y también en la planificación del desarrollo de programas informáticos, para esbozar la estructura del programa antes de realizar la efectiva codificación.

No existe una sintaxis estándar para el pseudocódigo. Aunque sea parecido, el pseudocódigo no debe confundirse con los programas esqueleto que incluyen código ficticio, que pueden ser compilados sin errores.

Ejemplo

Proceso millasMetros

//DATOS:

//CONSTANTES

Definir MILL_METR **Como Numérico**;

MILL_METR = 1609;

//VARIABLES

Definir Dmillas **Numérico**;

Definir Dmetros **Numérico**;

//ALGORITMO:

Escribir "Distancia en millas: ";

Leer Dmillas;

Dmetros = Dmillas * MILL_METR;

Escribir "Distancia en metros:", Dmetros

FIN

Para el diseño de soluciones informáticas de cualquiera de los 2 tipos que hablamos anteriormente es necesario tener en cuenta los conceptos que constituyen dichas soluciones como son las variables, constantes, tipos de datos, instrucciones, expresiones y otros más que veremos a continuación

Actividad 01

Los siguientes ejercicios se deben realizar dentro y fuera del ambiente de aprendizaje utilizando lápiz y papel en diagrama de flujo y en pseudocódigo.



Crear el Diagrama de flujo de los siguientes ejercicios según las indicaciones del instructor:

1. Hacer un algoritmo que declare una variable para guardar el promedio del semestre, otra para guardar el nombre de un estudiante y otra para guardar el número de notas perdidas. Mostrar todos los datos
2. Hacer un algoritmo que lea dos números enteros A y B y muestre el resultado de realizar: $(A + B) * 2 + 10$
3. Hacer un algoritmo que lea el nombre de un artículo, el valor unitario, la cantidad a comprar y muestre el nombre y el total a pagar.
4. Hacer un algoritmo para sumar dos números, los cuales serán tecleados por el usuario. Mostrar el resultado.
5. Hacer un algoritmo que lea el alto y el ancho de un rectángulo y muestre su área y su perímetro.
6. Hacer un algoritmo que lea el nombre de una persona, el valor de la hora trabajada y el número de horas que trabajó. Se debe mostrar el nombre y el pago de la persona.
7. Pedir el radio de un círculo y calcular su área. $A = \pi * r^2$.
8. Pedir el radio de una circunferencia y calcular su longitud.
9. Pedir el lado de un cuadrado, mostrar su área y su perímetro.
10. Calcular el área de un rectángulo de lados X e Y.

- **Actividades de transferencia del conocimiento**

Los aprendices deberán demostrar los conocimientos conceptuales y procedimentales que adquirieron a partir de las actividades de apropiación mediante el siguiente instrumento de evaluación:

Los aprendices deberán realizar cada una de las actividades propuestas para adquirir dominio en los conocimientos relacionados con los diferentes temas. Se evaluarán los conocimientos adquiridos mediante instrumentos de evaluación de Desempeño y Producto relacionados en la presente guía

- **Ambiente Requerido**

Ambiente de SISTEMAS con conexión eléctrica e internet

- **Materiales**

- Computadores (30)
- Sillas (3)
- Televisor (1)
- Resma tamaño carta (1)
- Marcadores (3)
- Lápiz (1)
- Lapicero (1)

4. ACTIVIDADES DE EVALUACIÓN



Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento: Evidencias de Desempeño: Asistencia y participación activa en las diferentes actividades propuestas Evidencias de Producto: Respuestas y procedimiento de los talleres realizados	Crea la base de datos en el motor de base de datos seleccionado, siguiendo especificaciones técnicas del informe, según normas y protocolos de la empresa.	Observación: EXC_D_01 Valoración del Producto: EXC_P_01

5. GLOSARIO DE TÉRMINOS

Sistema de información: es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su uso posterior, generados para cubrir una necesidad o un objetivo

Sistema operativo – Es un conjunto de programas que sirven para manejar un ordenador.

Software - El conjunto de programas, procedimientos y documentación asociado a un sistema informático.

Javascript: es un lenguaje de programación del lado del cliente que se utiliza con frecuencia en diseño WEB para generar efectos más complejos que no se puedan alcanzar usando HTML.

HTML: Siglas de las palabras inglesas: Hypertext Markup Language. Es decir, lenguaje de marcado de hipertexto. Lenguaje informático para crear páginas web. Conjunto de etiquetas o instrucciones que permiten estructurar el contenido de una web e incluir los hipervínculos o enlaces a otras páginas. Este lenguaje lo inventó en 1991 el Doctor Berners-Lee del CERN en Suiza.

HTTPS: Siglas de las palabras inglesas: HyperText Transfer Protocol Secure o versión segura del protocolo HTTP. Es el protocolo empleado para la transferencia de ficheros HTML cifrados que puedan contener información confidencial.

HTTP: siglas de las palabras inglesas: Hypertext Transfer Protocol. A saber en español: Protocolo de Transmisión de Hipertexto. Protocolo estándar de transferencia de hipertexto. Es decir: el protocolo de comunicaciones en el que está basado la Word Wide Web.

Script: es un archivo de órdenes o archivo de procesamiento por lotes. Es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

MySQL: es un sistema de gestión de bases de datos de código abierto que, junto con PHP, permite darle a las páginas web cierto dinamismo, es decir, disponer de manera adecuada los datos solicitados por los



navegadores. Es un sistema multiplataforma y su uso está tan extendido en las bases de datos que podría considerarse un estándar.

SEO (Search Engine Optimisation) Optimización en buscadores: técnica utilizada para asegurar que una página Web es compatible con los motores de búsqueda y así tener la posibilidad de aparecer en las posiciones más altas en los resultados de búsqueda.

Diseño web adaptable (responsive web design): se llama así al diseño web de aquellas páginas que se adaptan al tamaño de la pantalla o ventana en que se despliegan, por medio del uso de, idealmente, un solo documento HTML y un solo documento CSS. Esto permite hacer una sola página web para smartphones, phablets, tablets y PC.

Diagrama o Modelo Entidad Relación (DER): denominado por sus siglas en inglés, E-R "Entity relationship", o del español DER "Diagrama de Entidad Relación") es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades

Bases de Datos (BD): es un banco de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera, pero que comparten entre sí algún tipo de vínculo o relación que busca ordenarlos y clasificarlos en conjunto.

6. REFERENTES BIBLIOGRÁFICOS

- Documentos técnicos relacionados en la plataforma
- <https://www.php.net>
- <http://formaentic.weebly.com/>

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Julian Salazar Pineda	Instructor	Centro de Procesos Industriales y Construcción	10 Febrero de 2024

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					