PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA 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 lasnecesidades tecnológicas en el manejo de la información, de acuerdocon las normas y protocolos establecidos en la empresa
- **Duración de la Guía:** 40 horas

2. PRESENTACIÓN

En la actualidad las computadoras se han convertido en una de las herramientas más importantes en todas las actividades humanas, por las grandes cantidades de información que procesan a unas velocidades que el ser humano no podría igualar. De aquí la importancia de aprender a manejar dicha herramienta, pero lo más importante es desarrollar la habilidad de **resolver problemas** con la utilización de las computadoras. Espero que este trabajo sirva a ese propósito. Para lograrlo, debemos trabajar en la construcción de algoritmos, para desarrollar la lógica y poder trabajar con las máquinas según nuestra necesidad.

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

Los aprendices deberán descargar el documento **Actividad operadores.pdf**, y de forma individual desarrollar los ejercicios. Los ejercicios se deberán realizar con papel y lápiz y la evidencia será subir las fotos de las acciones realizadas según indicaciones del instructor. En esta actividad se identificarán conocimientos previos de los aprendices.

ALGORITMOS

Un algoritmo es una secuencia de pasos lógicos y ordenados con las cuales le damos solución a un problema determinado.

En la vida diaria cada uno de nosotros diseña y realiza algoritmos para solucionar los problemas cotidianos, es así que al levantarnos de la cama ya tenemos en la mente una serie de pasos que debemos seguir para llegar al sitio de estudio o al sitio de trabajo. Una vez en el sitio de estudio, tenemos en nuestra mente una serie de tareas que debemos realizar en unos horarios ya definidos.

Si quisiéramos realizar una comida especial, en nuestra mente construimos un algoritmo o serie de pasos que debemos seguir en un orden específico para que todo nos salga como queremos. Si quisiéramos transcribir estos pasos en una hoja de papel, para que otra persona realizará las mismas tareas y obtenga el mismo resultado que nosotros, debemos seguir una serie de normas para que esta otra persona nos entienda.

Por ejemplo, debe estar escrito en el idioma que ella comprende, se deben enumerar los pasos etc. Las normas que se deben seguir al momento de transcribir el algoritmo dependen de quién será el encargado de ejecutarlo, por ejemplo si quisiéramos escribir la receta para que la ejecute una persona adulta las normas serán diferentes a las que debemos seguir si quisiéramos escribir la receta para que le ejecute una niña.

LAS CARACTERÍSTICAS DE LOS ALGORITMOS:

- Un algoritmo debe ser **preciso,** tiene que resolver el problema sin errores e indicar el orden de realización de cada paso
- Un algoritmo debe estar **definido**. Si se sigue el algoritmo varias veces, se debe obtener el mismo resultado cada vez.

- Un algoritmo debe ser **finito**. Si se sigue el algoritmo, se debe terminar en algún momento, o sea debe tener un número finito de pasos (tiene un inicio y un final).
- Debe ser **legible,** cualquier persona que vea el algoritmo debe ser capaz de comprenderlo

Actividad 01

los aprendices deberán realizar las siguientes actividades (Tiempo estimado: 15 min):

- Reflexionar sobre los conocimientos que tengan relacionados con la lógica de programación y mediante la lectura de los siguientes enunciados, los aprendices permitirán que el instructor obtenga la información necesaria para determinar el punto de partida con la temática relacionada con la Lógica de Programación.
- Con base a los ejemplos y la lectura, realizar los siguientes algoritmos:
  - Algoritmo de las actividades para ir al trabajo
  - Realización de una consignación en una entidad bancaria.

La realización de esta actividad le permitirá al instructor crear el plan de trabajo a seguir con los aprendices, teniendo en cuenta los conocimientos previos y la base conceptual sobre la cual, dará continuidad a la actual fase del ciclo de vida del software que corresponde a la etapa del proyecto de formación que se adelanta actualmente.

Algoritmos para ser ejecutados por personas

Para que un algoritmo sea ejecutado por una persona, debe estar escrito de tal manera que esta persona lo entienda claramente, algunas de las normas que debe seguir la construcción del algoritmo son las siguientes:

- Debe estar escrito en el idioma que comprende la persona que realizará el algoritmo.
- Debe enumerar cada uno de los pasos a realizar en un orden lógico.
- Debe utilizar palabras que comprenda claramente la persona que realizará el algoritmo.

Algunos de los algoritmos diseñados para que sean ejecutados por personas son: las recetas de cocina, los manuales de funcionamiento, itinerarios, guía de matrícula etc.

Ejemplos de algoritmos para ser ejecutados por personas:

- Un cliente ejecuta un pedido a una fábrica. La fábrica examina en su banco de datos si el cliente está activo (no es moroso con sus deudas) entonces se acepta el pedido, en caso contrario se rechaza.

1. Inicio
2. Leer el pedido
3. Examinar ficha del cliente
4. Si el cliente está activo aceptar el pedido, en caso contrario rechazar el pedido.
5. Terminar

- En la taquilla de una sala de cine, se pide la identificación de los usuarios, se verifica si es mayor de edad (mayores 17 años) si es así se cobra el valor de la entrada y se deja pasar, en caso contrario no se deja entrar.

1. Inicio
2. Pedir cédula
3. Si la edad es mayor a 17 ir a 4 en caso contrario ir a 6
4. Pedir el valor de la entrada y dejar pasar al cliente
5. Siga en 7
6. No dejar pasar al cliente
7. Terminar

Algoritmos para ser ejecutadas por las computadoras

Los pasos para la solución de un problema utilizando como herramienta la computadora son:

- Diseño del algoritmo que describa la secuencia ordenada de pasos, que conducen a la solución de un problema dado (análisis del problema y desarrollo del algoritmo).
- Expresar el algoritmo como un programa en un lenguaje de programación adecuado (fase de codificación). La actividad de expresar un algoritmo en forma de programa se denomina programación.
- Ejecución y validación de programa por la computadora.

El primer paso es el más importante, en él se determina el problema y describimos una posible solución, utilizando nuestra malicia, conocimientos y habilidad para dar una solución al problema.

Si queremos hacer algoritmos para que las computadoras los ejecuten, debemos seguir ciertas normas y ese es uno de los objetivos de este trabajo, enseñar las normas que se deben seguir para realizar algoritmos que luego serán ejecutados por las computadoras. El otro objetivo es darle la posibilidad de desarrollar la habilidad de solucionar problemas, pero como todas las habilidades esta sólo se adquiere después de hacer muchos intentos.

Ahora los aprendices realizarán la lectura del siguiente texto, con el fin de documentarse y obtener un conocimiento básico relacionado con las partes de un algoritmo. Esta actividad debe realizarse en el ambiente de aprendizaje y bajo la guía del instructor, quien se encargará de resolver las inquietudes que surjan con dicha lectura. Tiempo máximo: 10 minutos

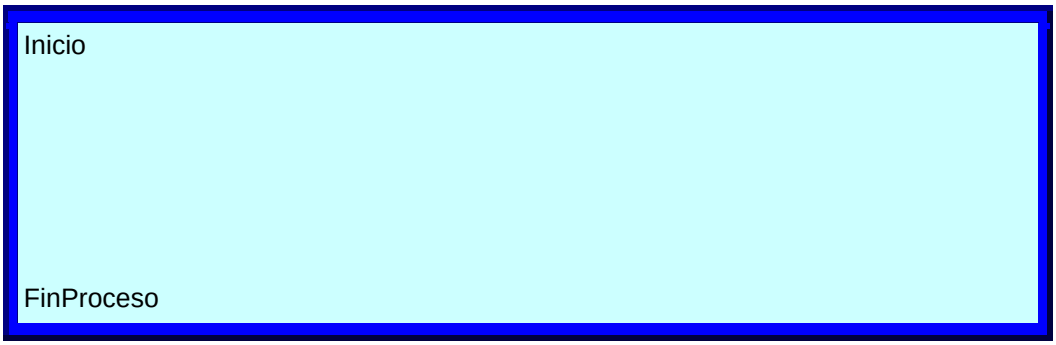
PARTES DE UN ALGORITMO

INICIO Y FIN:

Una de las características de los algoritmos es que deben ser finitos. Se debe indicar claramente donde inicia y donde termina.

Para indicar, donde comienza nuestro algoritmo vamos a utilizar la palabra *Inicio* y para indicar donde se termina nuestro algoritmo vamos a utilizar las palabras *FinProceso*.

Ejemplo: El siguiente algoritmo, que no hace nada sólo indica donde inicia y donde termina.



DECLARACIÓN DE VARIABLES:

Los datos son una parte muy importante en un algoritmo, pues son ellos el punto de partida y son ellos quienes sufren las transformaciones que darán los resultados deseados. Por esta razón el algoritmo debe guardar los datos en un sitio donde los pueda leer y modificar cada vez que lo requiera. Los sitios donde el algoritmo guarda los datos los llamaremos ESPACIOS DE MEMORIA y el tamaño de estos dependen del tipo de dato que se quiera guardar en ellos.

Además, si se tienen varios datos se debe tener la posibilidad de diferenciarlos de una manera que no se presenten confusiones asignándoles un identificador válido y único a estos espacios de memoria.

Un espacio de memoria se denomina VARIABLE cuando su contenido puede variar en el tiempo y de CONSTANTE cuando no se permite que su contenido varíe.

Lo primero que se hace en al algoritmo es declarar las variables. Donde se separan los espacios de memoria del tamaño indicado según el tipo de dato que guardarán y asignándoles un nombre o identificador válido, con el cual nos referiremos a la información que se guarda en dicho espacio de memoria.

Para declarar las variables se hará de la siguiente forma:

```
Definir variable Como TipoDeDato;
```

- Primero se coloca la palabra reservada **Definir**, luego se coloca el nombre de la variable seguido de la palabra reservada **Como**, y por último el tipo de dato según la información que se guardará en los espacios de memoria a crear.

ENTRADA DE DATOS:

Cuando un algoritmo requiera que el usuario ingrese datos, se utilizará la instrucción LEER seguido del nombre de la variable donde se guardarán los datos ingresados por el usuario.

```
leer identificador1
```

También se pueden leer varias variables al tiempo, de esta forma se podrá tener la siguiente instrucción:

Leer X, Y, Z

donde se le pide al usuario que entre tres datos el primero de los cuales se guarda en el espacio de memoria que tiene identificador X, el segundo se guardará en el espacio de memoria que tiene identificador Y, y el tercero se guardará en el espacio de memoria que tiene identificador Z.

LA OPERACIÓN DE ASIGNACIÓN

Es el modo de copiar un valor específico en una variable o espacio de memoria. La operación de asignación se representa con el símbolo: **<-**

La forma general de una operación de asignación es:

```
variable <- valor
```

Hay que tener en cuenta que en una variable determinada sólo se podrán guardar datos que correspondan al tipo con el que fue declarada la variable. Por esta razón veamos cómo realizar la asignación en cada uno de los tipos de datos.

- **ASIGNACION EN VARIABLES NUMÉRICAS (de tipo entero o tipo real):**

En una variable numérica(declarada como entero o real) sólo se podrán guardar datos numéricos y existe dos formas de hacerlo:

1. Asignación de un número (constante numérica) a una variable numérica. Asumamos que la variable A ha sido declarada de tipo entero, si queremos copiar el valor de 5 en ella lo haríamos de la siguiente manera.

A <- 5

Se copia el valor de 5 en la variable A.

2. Asignación del resultado de una expresión aritmética a una variable numérica. asumamos que las variable A ha sido declarada de tipo entero y queremos copiar en ella el resultado de la expresión aritmética 5*2 + 1 se haría de la siguiente manera.

A <- 5* 2 +1

Se evalúa primero la expresión aritmética y el resultado se guarda en la variable A. En este caso se copia en A el valor de 11.

Es posible que la expresión aritmética este formada con variables numéricas, para ilustrarlo asumamos que A, B y C han sido declaradas de tipo entero y que en B ya hemos copiado un 10 (B = 10) y en C hemos copiado un dos (C = 2). Si queremos copiar en A lo que tiene B mas lo que tiene C, se haría de la siguiente manera.

A <- B + C

Se evalúa la expresión aritmética, teniendo en cuenta los contenidos de las variables B y C, en este caso el resultado es 12 que se asigna a la variable A.

ASIGNACIÓN EN VARIABLES DE TIPO CADENA

En una variable de tipo cadena se pueden guardar cadenas directamente o el resultado de evaluar una expresión de cadena.

Asumamos que la variable NOMBRE ha sido declarada de tipo cadena, si queremos copiar en ella el nombre "María" lo haríamos de la siguiente manera.

NOMBRE <- "María"

Copiaría en la variable NOMBRE el valor de "María".

ASIGNACION EN VARIABLES DE TIPO LÓGICO

En una variable lógica sólo se podrán guardar datos lógicos ("verdadero" o "falso") o el resultado de evaluar una expresión lógica. Asumamos que la variable BANDERA ha sido declarada de tipo lógico, si queremos copiar en ella el resultado de la expresión lógica 2 > 10, lo haríamos de la siguiente manera.

BANDERA <- 2 > 10

Copiaría en la variable BANDERA el valor de "falso", pues 2 no es mayor que 10.

También podemos asignar la variable de la siguiente manera:

BANDERA <- verdadero

INSTRUCCIONES DE DECISIÓN

Las instrucciones de decisión son necesarias cuando en un algoritmo una o muchas tareas se deben hacer o no, dependiendo de una situación en particular. Esta situación nos ayudará a decidir si hacemos o no las tareas indicadas. Las instrucciones de decisión que estudiaremos son las básicas, la INSTRUCCIÓN DE DECISIÓN **SI** y la INSTRUCCIÓN DE DECISIÓN **SI NO**.

Instrucción de decisión SI

La estructura es la siguiente:

```
Inicio
Instrucción 1
si expresión_lógica entonces
.....Instrucción 1
.....Instrucción 2
.....Instrucción 3
FinSi

Instrucción n
…
finProceso
```

Se evalúa la expresión lógica si es verdadera se realizan las instrucciones internas, de lo contrario se salta a la siguiente instrucción fuera de la estructura en este caso a la instrucción.

Instrucciones de decisión SI NO

La estructura es la siguiente:

```
Inicio
Instrucción 1
si expresión_lógica entonces
.....Instrucción 11
.....Instrucción 12
.....Instrucción 13
SiNo
.....Instrucción 21
.....Instrucción 22
.....Instrucción 23
FinSi
Instrucción n
…
fin del programa
```

Se evalúa la expresión lógica si es verdadera se realizan las instrucciones internas al si en este caso las instrucciones (11, 12, 13,...) y luego se sigue con la instrucción n. Si la expresión lógica es falsa se realizan las instrucciones internas al si no en este caso las instrucciones (21, 22, 23) y luego se sigue con la instrucción n. Es de notar que este tipo de instrucción es excluyente, o sea que si entra por el sí, no entra por el si no. Y si entra por el si no, no entra por el sí.

INSTRUCCIONES DE REPETICIÓN

Las instrucciones de repetición son necesarias cuando en un algoritmo hay que realizar una o muchas tareas varias veces, las instrucciones de repetición básicas son: el MIENTRAS y el PARA, cada una de las cuales tiene su propia representación y su propia manera de controlar el número de veces que se repetirá el ciclo (instrucciones internas). Estas características hacen que una instrucción de repetición sea más adecuada que la otra en una situación particular.

ESTRUCTURA MIENTRAS (while)

La estructura repetitiva mientras es aquella en que las instrucciones internas (bucle) se ejecutan mientras se cumple una determinada condición. La estructura es la siguiente:

```
Inicio
Instrucción 1
mientras expresión_lógica hacer
.....Instrucción 11
.....Instrucción 12
.....Instrucción 13
FinMientras
Instrucción n
…
fin del programa
```

Cuando se ejecuta la instrucción mientras. La primera cosa que sucede es que se evalúa la condición (una expresión lógica). Si la expresión es falsa, ninguna acción del bucle (parte interna) se ejecuta y el programa continúa en la siguiente instrucción al bucle. Si la expresión es verdadera, entonces se ejecuta el cuerpo del bucle. Después de lo cual se evalúa de nuevo la expresión booleana. Este proceso se repite una y otra vez mientras la expresión lógica (condición) sea verdadera. Dentro del cuerpo del bucle debe existir una instrucción que modifique la expresión de tal manera que en algún momento haga que su valor sea falso. Es decir que garantice la terminación del ciclo.

Para controlar el número de repeticiones del ciclo se puede hacer de dos maneras:

Utilizando una variable contador:

Estructura mientras (while) controlada con una variable contador:

Una situación específica de esta estructura se muestra a continuación:

```
i = valorInicial
mientras (i <= valorFinal) hacer
    mostrar ( i )
    i = i + incremento
FinMientras
```

En este tipo de mientras se pueden identificar las siguientes partes:

- **Variable controladora:** En este ejemplo la variable controladora es i, pues i es la variable que forma parte de la expresión lógica que decide si se entra o no el ciclo.
- **Valor inicial:** corresponde al valor inicial que toma la variable controladora, siempre se asigna antes de entrar al ciclo, en este caso el valor inicial es uno (1).
- **Valor final:** corresponde al valor final que puede tomar la variable controladora para que ingrese al ciclo. Siempre forma parte de la expresión lógica, en este caso el valor final es 10.

- **Incremento:** es el valor que se le incrementa a la variable controladora. Siempre se hace dentro del ciclo, en este caso el incremento es de uno (1).
- Número de repeticiones: se calcula a si:

Número de repeticiones = (valor final - valor inicial + 1) / incremento

Número de repeticiones = (10 - 1+1) / 1 = 10

Según la información que se tenga para construir el mientras se pueden estar en uno de los siguientes casos:

1. Cuando se conoce el valor inicial, el valor final y el incremento.
2. Cuando se conoce el valor inicial, el incremento y el usuario entra el valor final.

Utilizando una variable centinela:

Estructura mientras (while) controlados por una variable centinela:

Se utilizan cuando no sabemos cuántas veces se debe repetir el ciclo. En este caso se utiliza una variable que nos servirá de centinela, para indicarnos cuando terminar el ciclo.

Un ejemplo clásico del centinela es cuando el enunciado nos indica que el usuario ingresará una lista de datos mayores que cero (0) y para notificar de que no quiere ingresar más datos ingresa un dato negativo o cero.
Se utiliza el siguiente esquema.

```
leer  N
mientras N > 0 hacer


   leer N
fin mientras
```

Observar que es necesario leer antes de ingresar al mientras y antes de cerrar el mientras. La variable que está sirviendo de centinela es N el dato que ingresa el usuario, pues si el dato no es mayor que cero se termina el ciclo.

Estructura de repetición para (FOR)

Permite que un grupo de instrucciones se repita cero o más veces, dependiendo del valor que resulte al evaluar una expresión de tipo lógico.

La estructura es la siguiente:

```
Para expresión_inicio, expresión_lógica, expresion_incremento
.....Instruccion1
.....Instruccion2
.....Instruccion3
.....Instruccion4
…
FinPara
Instrucción n
…
```

La **expresión_inicio** establece la condición inicial para la variable de control evaluada en la expresión lógica.

La **expresion_incremento** modifica la variable de control.
La **expresión_lógica** es una expresión formado con la variable de control, y que sirve para controlar el número de iteraciones del ciclo el cual termina cuando su valor sea falso

Ejemplo:

```
para i=1 hasta 10 con paso 1 hacer
    Escribir 'Número: ', i;
FinPara
```

Actividad 02

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 los siguientes algoritmos para ser ejecutados por personas:

1. Un retiro de dinero por parte de un cliente en una entidad bancaria.
2. Crear un algoritmo que describa los pasos necesarios para ir a la universidad a clase de 6 a.m. tenga en cuenta que si llega 10 minutos tarde no puede entrar al salón de clase y si al llegar no trae el carnet no puede entrar a la universidad.
3. Ir al cine.
4. Reparar un pinchazo de una bicicleta.
5. Hacer una llamada a un compañero.

Crear los siguientes algoritmos para ser ejecutados por computadoras:

1. Hacer un algoritmo que declare una variable para guardar el número de horas de estudio, y otra para guardar el nombre. Escribir ambos datos.
2. Hacer un algoritmo que lea dos números enteros A y B y muestre el doble de su suma.
3. 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.
4. Hacer un algoritmo que lea dos números enteros A y B y muestre el resultado de realizar: (A + B) *2 +10
5. Hacer un algoritmo que declare una variable para guardar el nombre de una persona, otra para guardar la comida preferida y otra para guardar la cantidad de dinero que posee.
6. 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.
7. Hacer un algoritmo para sumar dos números, los cuales serán tecleados por el usuario. Mostrar el resultado.
8. Hacer un algoritmo que lea el nombre de una persona y número de horas que estudia en la semana.
9. Hacer un algoritmo que lea el nombre de un estudiante, la cantidad de materias perdidas y la cantidad de materias ganadas.
10. Hacer un algoritmo que lea el alto y el ancho de un rectángulo y muestre su área y su perímetro.
11. Hacer un algoritmo que lea dos números enteros A y B y muestre su diferencia.
12. 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.
13. Pedir el radio de un círculo y calcular su área. A=PI*r^2.
14. Pedir el radio de una circunferencia y calcular su longitud.
15. Pedir el lado de un cuadrado, mostrar su área y su perímetro.
16. Calcular el área de un rectángulo de lados X e Y.
17. Pedir dos números y decir si son iguales o no.
18. Pedir un número e indicar si es positivo o negativo.
19. Pedir dos números y decir si uno es múltiplo del otro.
20. Pedir dos números y decir cuál es el mayor.
21. Pedir dos números y decir cuál es el mayor o si son iguales.
22. Pedir dos números y mostrarlos ordenados de mayor a menor.
23. Pedir tres números y mostrarlos ordenados de mayor a menor.
24. Pedir un número entre 0 y 9.999 y decir cuántas cifras tiene.
25. Pedir una nota de 0 a 5 y mostrarla de la forma: Insuficiente (0 – 2,9), Suficiente (3 – 4,5) y Bien (4,6 – 5)
26. Pedir una nota numérica entera entre 0 y 10, y mostrar dicha nota de la forma: cero, uno, dos, tres...
27. Pedir un número y decir si es par o impar.
28. Un trabajador recibe su pago, según la cantidad de horas trabajadas y su valor. Si la cantidad de horas trabajadas es mayor que 40, éstas se consideran horas extra, y tienen un incremento de $10000 (diez mil) sobre el valor de la hora. Calcular y mostrar el salario (pago) del trabajador. Nota: leer horas trabajadas y valor de la hora.
29. Dado un monto, calcular el descuento considerando que por encima de 100 el descuento es del 10% y por debajo de 100, el descuento es del 2%.
30. Leer dos números y calcular su división, teniendo en cuenta que el denominador no debe ser 0 (cero)

- **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

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: **Evidencias de Desempeño:**	Crea la base de datos en el	

Asistencia y participación activa en las diferentes actividades propuestas **Evidencias de Producto:** Respuestas y procedimiento de los talleres realizados	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 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 estandar.

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 BILBIOGRÁFICOS

- 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 Humberto Salazar Pineda	Instructor	Centro de Procesos Industriales y Construcción	14 de Octubre de 2020

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					